CONSTRAINED MODEL PREDICTIVE CONTROL FOR COMPLIANT POSITION TRACKING OF PNEUMATIC SYSTEMS

A Dissertation
Presented to
The Academic Faculty

by

Hannes G. Daepp

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
The George Woodruff School of Mechanical Engineering

Georgia Institute of Technology
August 2016

CONSTRAINED MODEL PREDICTIVE CONTROL FOR COMPLIANT POSITION TRACKING OF PNEUMATIC SYSTEMS

Approved by:

Wayne J. Book, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Aldo A. Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Nader Sadegh
School of Mechanical Engineering
Georgia Institute of Technology

Mark Costello
School of Aerospace Engineering
Georgia Institute of Technology

Eric J. Barth
School of Mechanical Engineering
Vanderbilt University

Date Approved: May 2, 2016

To my parents,

Pam and Ueli

ACKNOWLEDGEMENTS

I would like to thank my adviser, Dr. Wayne J. Book, for his invaluable mentorship. His guidance and expertise have been a tremendous asset to my graduate career, and I've enjoyed our conversations, professional and otherwise. I am also grateful for the helpful insights and feedback from my committee, Drs. Sadegh, Ferri, Barth, and Costello. Furthermore, I'd like to thank J.D. Huggins, who showed me how to properly design hardware testbeds early on in my Ph.D., and who assisted later, providing guidance based on his worthy experience. I'd also like to acknowledge Dr. Alper Erturk and the SSDSL for use of their seismic shaker, and Dr. Stephen Leadenham for helping to initially configure it.

I would further like to express my gratitude to my fellow lab members in the IMDL: Heather Humphreys, Brian Post, Marc Killpack, Ryder Winck, Mark Elton, Sam Seifert, Rahul Chipalkatty, Aaron Enes, and Longke Wang. I was fortunate to have a great group of lab mates, and am thankful for their plentiful advice, encouragement, and guidance. I also benefited from many rewarding conversations with my friends and colleagues at Georgia Tech – Stephen Leadenham, Wayne L. Maxwell Jr, Arto Kivila HRH, Ellen Skow, Elliott Gruber, Martin Cacan, Anne Mallow, Chris Sugino, Jason Jones, Shima Shahab, John Dykes, Melih Turkseven, Billy Gallagher, Antonio Moualeu, David Samet, Nick Earnhart, and Alek Kerzhner – in our offices, labs, or on any of many, many trips to get coffee.

This thesis was made possible by my funding sources – the Center for Compact and Efficient Fluid Power, which initiated my involvement in pneumatics and fluid power, the National Defense, Science, and Engineering Graduate Fellowship (NDSEG), which supported me from 2011-2014, and the Woodruff School and ME 2110, for whom I was a Teaching Assistant for the last five semesters of my graduate career. I'd especially like to thank CJ Adams, Gina Magnotti, and Dr. Tom Kurfess, as well as numerous engaging, talented, and motivated undergraduates, for helping to make my TA experience a positive one.

Finally, I'm indebted to my parents, who showed me first-hand the the value of a strong and diligent work ethic, knowledge, and focus, to my grandparents, who patiently waited for me to complete this degree and only sparingly mentioned their advancing age, and to my sister, for editing the only section she could understand – this one. I've also been fortunate to be part of a supportive community of friends at Georgia Tech and in Atlanta, and in addition to my friends and colleagues above, would like to thank Drs. Emily Johnston, Theresa Wilkes, Amanda Clemm, and Joe Gruber, as well as Kavin Manickaraj, Graeme Wicks, and Lauren Eskew for showing me shrimp and crawfish boils, proper river shoots, scavenger hunts, how to dominate at triva, and joining in many other memorable adventures that helped provide relief from graduate school and made Atlanta a great home for the last few years.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Pneumatic actuation is frequently applied to situations that warrant inherent compliance, such as prostheses, orthoses or walking robots, i.e., natural motions and applications in which interaction with humans/the environment are anticipated. However, compliance, as well as friction, lead to position control challenges that are commonly countered using aggressive controllers like sliding mode (SMC) or high-gain PID control, resulting in stiff system dynamics. Even hybrid force-position controller dynamics are ultimately subject to a clear trade-off of compliance and accuracy. In this thesis, this challenge is addressed via a constrained Model Predictive Controller that treats compliance as a bound rather than a target to achieve *compliant tracking*. A comprehensive literature review explores the state-of-the-art and defines performance targets, and a set of 1 degree of freedom (DoF) tests is established to compare controllers and convert qualitative controller goals into quantitative design specifications. Four benchmark controllers that span the stiffness-accuracy spectrum – SMC, Linear Quadratic Regulation/Tracking, PID, and Impedance Control – are implemented in simulation and on hardware, and are used to produce baseline results and verify performance targets. The predictive controller is implemented with admittance and impedance constraints and compared to benchmarks on the 1-DoF system. Additionally, new friction compensation methods are introduced that leverage the predictive structure to improve friction compensation for slow systems, and are compared to additive compensation methods. Results show that constrained MPC enforces impedance bounds on a tracked system, and achieve results with accuracy comparable to the best benchmark performance at a given compliance bound. Additionally, because compliance is enforced as a bound rather than a target, the highest tracking accuracy achieved with MPCs ultimately happens at the minimum necessitated impedance, without a-priori knowledge of that impedance bound. Results are shown to extend to a multi-DoF system using a planar robotic arm with simultaneously actuated joints and subject to unexpected disturbances.

CHAPTER I

INTRODUCTION

The objective of this dissertation is to develop a controller for safe and accurate position control of pneumatic systems in the presence of humans and environmental interaction. For systems moving freely in the presence of humans, intrinsic safety is critical: unexpected collisions should involve low impact forces and be resolved by a stable, non-oscillatory return to desired motion. This behavior is ensured by using a compliant actuator and keeping output impedance low; impulse and force are kept to a minimum in the case of unexpected contact with users. Accuracy implies the minimization of position error with respect to a reference for free space tracking. Broadly, the goal is to develop an optimal position controller for naturally compliant actuators without unnecessarily sacrificing system stiffness.

Using a constrained model predictive controller (MPC), the system effectively decouples closed-loop impedance and error minimization by treating impedance as a constraint on an optimal tracking controller. The primary MPC approach uses a low-level force controller to simplify system dynamics, then applies impedance or admittance constraints, coupled with friction compensation as needed. The controllers were verified at distinct points in hardware and more thoroughly in a corresponding simulation, first on a single single degree-of-freedom (DoF) actuator, and then on a multi-DoF robotic platform.

1.1 Motivation

This research stems from the need for accurate, compliant control in applications near humans. It originated from work on the Compact Rescue Robot (CRR) [19, 25], shown in Figure 1, a six degree of freedom pneumatically actuated robot that uses two three-link arms to move about and interact with its surroundings. The robot is supported by a wheeled cart, and moves around by dragging the cart with its arms, which may function as front legs.

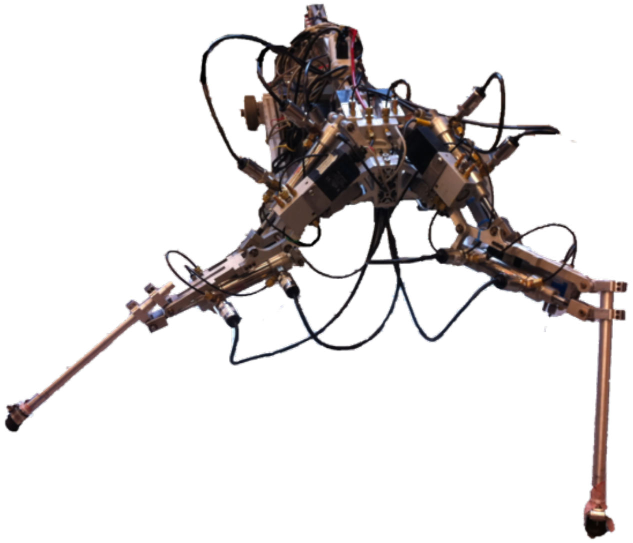

Figure 1: The Compact Rescue Robot (CRR).

The goal of the CRR is to demonstrate the benefits that fluid power could offer rescue robotics, particularly when combined with compact, lightweight power sources [38,68,69]. The robot is intended to be capable of rescue tasks, move freely, and interact with victims and the environment. Pneumatic power is a natural solution: it provides high power density for untethered operation, high force density when large force output is required, it is clean and safe, and it offers compliance for interaction and locomotion [10, 11, 51, 84]. In the specific case of moving near and around humans, the robot must be accurate, reliable, and inherently safe.

Pneumatic systems are prevalent among platforms intended to interact with humans and the environment. Several robots were developed prior to the CRR's inception [54,55,64], as well as systems for prostheses [44], and rehabilitation [8]. These applications are well-suited for pneumatic actuation, especially in terms of motion speed and bandwidth. Pneumatic systems are typically verified over trajectories changing at 2 - 5 Hz, which is on par with natural motions like walking, as well as human motions such as those employed in classic Human-Robot Interaction (HRI) situations.

However, even when the actuator is appropriately selected for the application, compliant control generally necessitates a trade-off. In the absence of additional sensing, most controllers sacrifice compliance for accuracy, as shown in Figure 2. The figure shows a preview of the benchmark controller parameter variation test results, detailed in Chapter 4, and illustrates a trade-off between compliance and accuracy among different controllers. Each dot represents a different set of controller and input parameters, and the coloring corresponds to particular types of benchmark controllers, as indicated in the figure. Compliance is measured by observed stiffness resulting from response to disturbance inputs (a component of the output impedance), and accuracy is indicated by tracking error from a separate reference-following test; therefore, the dots shows the comparative performance of a controller on each test. Ideally, a compliant controller would be in the lower left-hand corner. An aggressive controller like sliding mode control, while robust, will result in a correspondingly high-impedance closed loop system. Conversely, an impedance controller may be used to lower output-impedance, but in the absence of added force sensing, will result in a corresponding drop in accuracy. Other controllers can be used, but as evidenced in the figure, they are generally subject to the same trade-off between compliance and accuracy.

One solution is to add sensors and actuators [76]; however, in a robot such as the CRR, these added capabilities take up valuable space and operating power, especially as the degrees of freedom are increased. Additionally, it is useful to study the effect that control alone may have on system capability, since added sensing can then be used more efficiently. This thesis seeks to show how a pure controls solution can be used to increase compliance for a given accuracy (or vice versa), and there are two clear opportunities to do so. First, the capabilities of compliant control could be improved by generating controllers that are better than the benchmarks, i.e., that are able to reach the lower-left hand corner in Figure 2. Controllers like those seen in Figure 2 generally have a bandwidth greater than the necessary 2 - 5 Hz, and track well in this region, but are too stiff. One hypothesis, then, is that these controllers leave gaps in the low-frequency applications that could be filled by a more intelligently designed controller. However, a clearer opportunity is indicated by the dashed magenta box in the bottom-right side of the figure: there is a certain stiffness

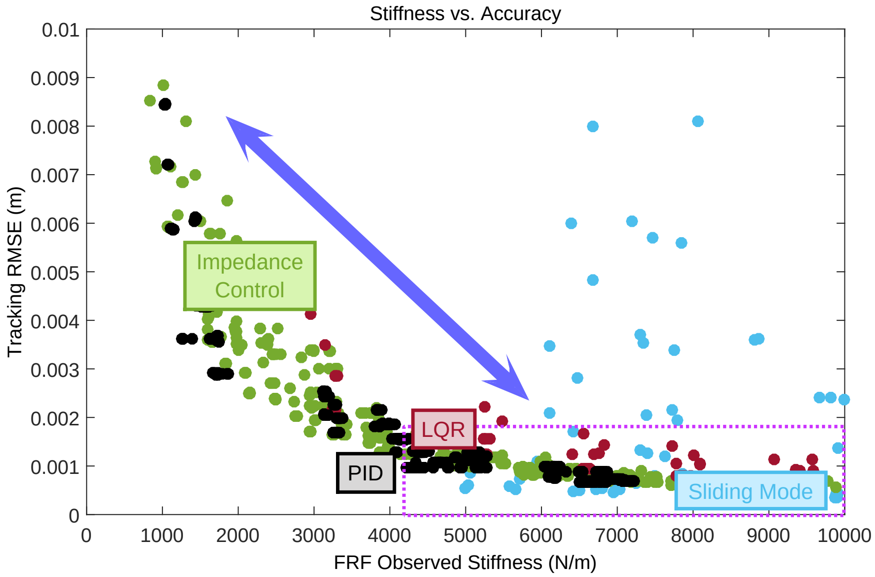

Figure 2: General trade-off of accuracy and compliance among controllers.

at which point the increase in accuracy is negligible. For compliant tracking, a controller should reach but not exceed this maximum required stiffness, ideally without requiring a priori knowledge of that target stiffness value. Similarly, the controller should obtain the best tracking for a given stiffness.

In this thesis, a constrained model predictive controller is used to achieve these goals an improve compliant control for pneumatic systems. Model Predictive Control provides an obvious framework: by coupling a low-level linearizing force controller with an MPC that treats impedance as a constraining upper bound (rather than a target), comparable accuracy can be achieved in the desired frequency regions without an equivalent rise in system stiffness, unless necessary for better tracking. The controller is applied to a standard pneumatic actuator consisting of a cylinder, one 5 port/3-way valve, pressure and position sensing, and the corresponding extensions to multiple linkages. The resultant controller is designed specifically for pneumatic systems, but may also be used for the broader category of actuators with intentional inherent compliance, notably Series Elastic Actuators (SEAs) and Variable Stiffness Actuators (VSAs).

1.2 Research Goals & Contributions

The primary asset of MPC for this application is its ability to include and enforce constraints on system behavior. By setting compliance constraints, MPC can be used to provide good tracking while preserving low-output impedance properties. However, the controller design is limited by implementation and theoretical requirements. Therefore, the research goals are intended to explore controller design and capability, and have associated contributions:

1. **Design of predictive controller to enforce desired tracking on a compliant system without significantly increasing system stiffness/impedance**
    - Definition of linearized state models
    - Derivation of constraints for compliance enforcement
    - Derivation of constraints that define limiting dynamics over prediction horizon
2. **Analysis of friction compensation needs and benefits**
    - Development of a predictive estimator that leverages MPC properties
    - Comparison of predictive and classical friction compensation
3. **Practical & Theoretical Considerations**
    - Stability & performance analysis
    - Hardware implementations with multiple DoF

These are realized in the form of 6 key components:

1. Literature review of needs, standards, and a set of proposed tests to establish clear design specifications for controllers
2. High fidelity simulation to compare controllers
3. Performance of benchmark controllers on 1-DoF system in simulation & hardware
4. Compliant control of a 1-DoF system with MPC in simulation & hardware

5. Compliant control of a 2-DoF system with MPC

6. Overview of explored extensions to further improve performance through control

1.3 Thesis Organization

The thesis structure first familiarizes the reader with problem detail and system dynamics, then introduces the controller, and then discusses theoretical analysis, implementation details, necessary extensions and results. The subsequent chapters primarily correspond to the the 6 key results introduced in section 1.2.

Chapter 2 provides a more detailed literature review on the state-of-the-art in compliant control, as well as needs and the particular applicability of pneumatic actuation. Chapter 3 discusses the dynamics of pneumatic systems and introduces testbed platforms and an equivalent simulation for model comparison. Next, Chapter 4 proposes a set of tests used to translate qualitative aims to design specifications for each controller, provides several benchmark controllers to span a performance baseline, and compares their capabilities. In Chapter 5, MPC design is detailed for a single degree of freedom system, and friction compensation, performance, and stability guarantees, are discussed, and the application to a multi-degree of freedom planar robot is addressed in Chapter 6. Finally, Chapter 7 will provide conclusions and discuss topics for future work.

CHAPTER II

BACKGROUND, MOTIVATION, & TARGETS

This chapter provides an overview of position control methods in pneumatic systems, the role of position control in the broader field of compliant systems and its relation to pneumatics, and the gaps that exist in the literature. Performance of comparable control strategies is discussed and used to derive specific target behaviors, and the use of MPC as a solution is heuristically justified.

2.1 Pneumatic actuator control

Pneumatic actuation has been widely employed in industry because of its many benefits, including power and force density, clean, safe actuation, and low cost [11,84], especially compared to electric motors [10]. Inherent compliance and light weight has made pneumatics a good choice for platforms geared at interaction with humans and the surrounding environment, such as walking robots [26,55] or prostheses [44]. Pneumatics is further appealing because of its potential to actively control stiffness or impedance properties [44,85].

However, many of these high performance applications have faced challenges in actuator control. The inherent compliance can cause vibrations and even instability. Control is particularly limited by friction. This must be compensated for or reduced through the use of low-friction cylinders [35], which are often more costly and fragile than regular cylinders. Friction compensation typically involves some form of feedback linearizing additive terms that are added to the control effort, though the success of this method is limited by the respectively slow pressure dynamics of the actuator, particularly when compensating for stiction, which occurs instantaneously. Further, pneumatic systems are highly nonlinear, including discontinuities resulting from flow direction, non-smooth transitions resulting from sonic flow conditions, and a number of nonlinear terms, especially in the case of choked flow.

As a result, numerous control strategies have been attempted to achieve good position and force tracking, and to bypass issues of friction and nonlinear behavior. As with many fields, there is no singular best control; instead, some controllers are more prevalent for specific applications.

2.1.1 Position & Force Tracking

Control for position and force tracking with pneumatic systems has been extensively explored over the past several decades. Simple control approaches have been investigated, such as Proportional-Integral-Derivative (PID) or Position-Velocity-Acceleration (PVA) control [44,51,93], but even well-tuned controllers of this form rarely achieve the tracking or robustness possible with more advanced controllers. The most successful basic controllers have used an inner loop on pressure and high proportional and derivative gains, thereby effectively transforming the system into a stiffer one with higher output impedance. More advanced controllers have been designed for position tracking, such as fuzzy state feedback [73], impedance control [100], neural networks [16], and adaptive control [10], though the best results have generally been achieved using sliding mode controllers [11], which enforce a level of robustness that seems particularly beneficial to the pneumatic servo control applications. These results have mostly been limited to single degree of freedom applications. In systems where human or environmental interaction is desired and force tracking is of interest, researchers have tested passivity-based control [31] and impedance control [67,90,100], which also enhances passivity [75], to ensure a safer and more stable interface.

Optimal control has been minimally explored as an option for control of pneumatic actuators, primarily for energy efficiency, as was done by Ke et al [49]. A few recent efforts have looked at model predictive control (MPC) – a subset of optimal control – for pneumatic systems, primarily because of the potential to enforce constraints while implementing an optimal control policy [6,18,57,85]. These papers typically use a stabilizing control loop on pressure or force, but differ in overall system design. Todorov et al. [85] applied MPC to determine two control inputs to a two-valve/single cylinder actuator that formed one joint of a 38-DoF pneumatic robot. They used an iterative Linear Quadratic Gaussian observer to estimate the MPC reference input states based on the desired force reference. Aschemann and Schindele [6] designed a special nonlinear model predictive controller (NMPC) for a pneumatic muscle actuator that was mechanically simplified by a valve that internally regulated mass flow. They also included a disturbance model in their reference that compensated for un-modeled parameters, such as friction. Matousek [57] applied an adaptive predictive controller that used recursive least squares to update the linearized model at each time step, resulting in generally successful control but periodically observed sudden high magnitude deviations from the reference seemingly after steady state had been reached. Finally, Chickh et al. [18] demonstrated that a feedback linearized pneumatic actuator can be reformulated into a particular form (CARIMA) for use in General Predictive Control (GPC), a particular implementation of MPC. MPC has also been successfully applied to related but structurally very different industrial pneumatic systems [99].

2.1.2 Friction Compensation

Friction, and particularly static friction, is prevalent in pneumatic control, and is often a main hindrance reaching desired, low error tracking behavior. Friction in pneumatic systems occurs because of the interaction between the seal and lubricant with the cylinder, as well as at distinct operating points, such as at the end stops. In pneumatic systems, friction is further troubling because it exists as a nonlinear term sandwiched between the pressure and position dynamics, commonly known as a Sandwich System [82, 83, 96]. A sandwiched nonlinearity is problematic because it can lead to oscillations or even instability [83]. When the nonlinearity is invertible, a control loop with an inversion term can be used to eliminate the nonlinearity [83]. However, past efforts have usually worked with static nonlinearities, such as a dead zone, which are simpler and likely more easily characterized than friction. Another method, for use with input saturation, is a scheduled low-gain approach [96], essentially a specialized gain scheduling effort.

Low-friction cylinders are useful for some applications, e.g. in-MRI surgery [35]. These minimize material friction by design, such as with graphite and glass as piston and cylinder materials, but are often more fragile and expensive than other pneumatic actuators. Further, the problem of friction compensation extends beyond pneumatics to a wide variety of similar mechanical systems, so a controls solution is more broadly applicable.

An alternative approach is to use a software-based compensation method. Armstrong-Helouvry et al provide a good overview of research on friction modeling, analysis, and compensation in their survey paper [5]. One method is to mitigate friction rather than actively compensate for it, such as via integral control, joint-torque control, stiff PD control, impulse control, or dither. Alternatively, a feed-forward term can offset the effects of friction [5, 42, 47, 72, 89]. Variations differ in choice of friction model and methods for including parameter uncertainty. Some treat friction as a periodic disturbance that may be identified in the frequency domain, then apply repetitive control to effectively adapt the compensation method over multiple iterations [72, 89]. Others use approaches such as feedback linearization to attempt to cancel the friction term, then recursively adapt the friction model.

One of the challenges with feed-forward compensation is that the system must be fast enough to provide the compensation term almost instantaneously. The best success is generally achieved using high bandwidth systems, such as electric motors [47, 89]. In the case of a pneumatic actuator, the pressure dynamics are slow compared to the switching nature of the stiction nonlinearity. Optimal control could address this concern, but its application has been limited largely due to the challenge of discontinuous co-states [30], which makes an analytical solution practically impossible. Recent efforts [79] have found that by designing problem-specific numerical methods, a computationally practical solution can be found, yet this requires a solver specially designed for that problem.

2.2 Pneumatics & Compliant Position Control

The properties of pneumatic actuators are particularly appealing to applications that involve interaction with humans and the environment, including search and rescue robotics [25, 64], prostheses [44], and rehabilitation [8]. Such applications often require compliant position control: the actuator or arm must be able to be accurately positioned using ideally smooth motions, but if obstructed by a user or object, interaction forces must be minimized to avoid damage and ensure user safety [67,88]. The rescue application is a particularly apt application – in previous work, the authors developed 6-DoF pneumatic rescue robot [25,26] to demonstrate how force and power density could be coupled with lightweight power sources to provide rescue machines that are capable yet also reassuring to a victim, and that possess a natural compliance that makes them safe to interact with.

2.2.1 Compliant Position Control: Demand, Definition, and Applications

Numerous researchers have pointed to the need for a compliant actuator and control scheme. In one of their introductory Series Elastic Actuator (SEA) papers, Pratt et al. stated that "for natural tasks where small motion bandwidth is not paramount concern, actuator to load interfaces should be significantly less stiff than in most present designs" [65]. They later elaborated on the concept of a "natural task", giving examples such as walking or manipulation, and offering benefits of compliant actuation, such as shock tolerance, lower reflected inertia, less damage during inadvertent contact, and energy storage. Other general application areas include physical therapy and gait/motion rehabilitation [60,67,88,91], prostheses [91], walking robots [4,66,91], as well as several efforts to create "human-friendly robots" [9, 27, 39, 41, 101]. These typically refer to almost-industrial robots: robots for industrial settings intended for use around humans, which are usually either more lightweight, smaller, or slower than state-of-the-art industrial systems. In rehabilitation and physical therapy applications, the user must often by trained in "natural motions" by a robot, but is prone to spasms to which the systems must be compliant. Therefore, these applications specifically require "compliant tracking" [88] or motion that is compliant to a user's motion [67]. To achieve compliant motion, it is common for robots for rehabilitation, physical therapy, and walking to apply impedance control or use series elastic actuators (which includes some effective impedance-lowering control), while the modern physical Human-Robot Interaction (pHRI) applications, such as extensive work at the DLR [39], tend to measure safety using more empirical strategies, such as the Head Impact Criterion and other *severity indices*, which are then used to define constraints on acceptable robot behavior. In general, the goal of a robot that possesses *intrinsic safety*, i.e. inherently safe dynamics, is highly desirable; the Robotics Handbook authors go so far as to call it the “holy grail” of safe human-robot interaction [9].

2.2.2 Safety & Impedance Standards

Most target compliant control applications either aim for compliant tracking, or for motion with safety guarantees for a user in the workspace. Both of these aims can be quantified in a number of ways.

One approach is to specify actuator (joint space) compliance [4], though for impedance control cases, it is much more common to cite the desired compliance at the end effector (task space compliance) [4, 67, 88]. This is an important distinction, as an actuator might have a stiffness in the 1000 - 10000 N/m range [4], but be significantly more compliant in the task space, with stiffnesses of 30 - 250 N/m [67], as corresponds to measured human impedance [88]. For industrial manipulators, linear stiffnesses are typically quite high– for example Ferretti et al. develop an impedance controller for elastic joints in industrial manipulators, and set an end effector target of $k = 2000N/m, \zeta = 13, \omega = 10rad/s$, as well as some lower angular impedances [33]. Several researchers, especially those focusing on variable stiffness actuators (VSAs), emphasize the need for dynamic range. For example, Tsagarakis et al. [87] developed a soft actuator for small-scale robots using springs with 62 kN/m stiffness, though based on their robots' geometry, these instead held a rotary stiffness of 150-160 Nm/rad. Using control, they modulated the stiffnesses from 30 to 600 Nm/rad (corresponding to 10 to 240 kN/m linear stiffness), or 0.5 - 4 x the mechanical stiffness, while tracking a 0.8 Hz signal. Several of these authors also examined the force step response to get an idea of the dynamics of the system [4, 101].

It is rather unclear from the literature how well the target impedances correspond to achieved impedances. While theory has been well established for some actuators, e.g. SEAs, and impedances can be analyzed qualitatively [80], most authors set a target and observe the response, but don't necessarily assess the correspondence between desired and target impedance. Instead, authors comment on qualitative or individual aspects, such as the peak force due to contact [100], RMSE error, or steady state contact force.

Researchers dealing more broadly with the concept of human-friendly robots or safe robots have instead investigated other types of safety measures. There are several *safety indices*: empirically deduced, quantitative metrics for danger, usually based on impact with some body part. One common example is the Head Impact Criterion (HIC) [9,27,39,41,60], which has been used by several researchers to define a speed constraint for a robot (typically a large, industrial robot). Haddadin et al. performed biomechanical tests onto further investigate the actual dangers of robotic collisions [41] and relate HIC to velocity criteria based on robot specifications. There are also ISO 10218 standards [1], though they have been suggested to be limiting by prominent researchers in the field [9, 39].

In short, there is no singular unifying standard that defines idealized compliant control. However, there are guiding metrics, established performance goals, and a bevy of results from past researchers that may be sourced to define numeric targets for a new compliant controller.

2.2.3 Performance of Comparable Systems

While the need for compliant control has been clearly established, it is necessary to know the typical operating ranges to understand why pneumatic actuators are a reasonable choice for this application. The following sections seek target performance ranges as seen in current compliant control applications. Furthermore, these targets are compared to the aims of pneumatic tracking systems and applications. The results from this survey will be used in section 2.2.4 to define target performance ranges.

2.2.3.1 Speeds and Frequencies

Many researchers in the compliant motion domain [27, 41, 46] reference the solution to the *safe brachistochrone problem* [36]: given a variable impedance system with state (velocity) constraints, find the stiffnesses that enable a path to be followed in minimal time (brachistochrones are minimum time problems; this one is specific to safety). The general solution consists of stiff slow motion and flexible fast motion. Bicchi et al. further investigate a number of optimal control solutions to variations on the safe brachistochrone problem [36], though of course they all deal with a minimum time solution. These investigations have resulted in some general guidelines: Haddadin et al. [41] found linear relationships between (mass,velocity,curvature) and injury, and general velocity bounds for robots have been suggested at 4.5 m/s [41], 3.3 m/s [97], or below.

While these speeds are limiting for robots with industrial aims, robots for walking, rehabilitation, physical therapy, etc., and other tasks in the "human-friendly" arena are generally subjected to slow tests. Tsuji & Tanaka [88] test tracking up to about 10 rad/s for a rehabilitation robot. Another physical therapy robot by Richardson et al., which used pneumatic actuation [67], tested excitations up to 2 Hz, and in fact filtered the input reference at 6 Hz. Some more general cases also include slower references: Pratt & Krupp state that while small-force SEA bandwidth is up to 50 Hz, the large-force bandwidth is in the 7-10 Hz range. In their introduction of Distributed Macro-Mini (DM^2) actuation, Zinn et al. [101] display position tracking at 1 and 2 Hz, and show error results for 5 Hz sinusoid tracking over a 15 cm range. In fact, compared to human motions, these rates are appropriate; one heuristic on bounding rates can be obtained from elite Olympic curlers, who sweep the floor (a rapid, oscillatory motion) at about 4.5 Hz [61], which matches the suggested upper bound on fast human motion.

As with human-like ("natural") motions, pneumatic tracking motions are typically slow. Pneumatic tracking is usually validated on reference curves up to 2 Hz [11, 52, 63, 100], rarely going higher, though some human-operated or walking position trajectories, while dominated by sub-2 Hz content, may contain some higher frequency content [31,67,90], and good tracking results have been measured at 4 Hz [76]. Most pneumatic tracking is tested along the actuator (no geometric stroke amplification) and following smooth sinusoidal curves, but other results have also looked at tracking S-curves over a 2 second span [94], human-guided or human-like motion [31], which follows slow, smooth curves, step responses [93], or curve tracking for walking motions [90], which are not necessarily smooth, but generally slow: 0.5 - 2 Hz over a 10 cm range, with a few immediate velocity changes.

2.2.3.2 Stiffness & Impedance Performance Expectations

Section 2.2.2 provided some example impedance and stiffness targets of HRI and human-like motion applications. This category is broadly defined and exact targets are largely dependent on application. Systems that use pneumatic actuators, however, are much more well-defined, since they have similar bounds on feasible performance. Accordingly, actuators of similar dimension are also more likely to be applied to similar scenarios: walking and motion of small- and medium-scale robots, rehabilitation, orthoses, etc.

Several authors that have applied impedance control to pneumatic systems. Najafi and Hejrati [63] examined impedance control for a fast on/off switching control of solenoid valves and tested tracking on sinusoids ranging from 0.1 to 1 Hz, a unit step, and a ramp tracking with sudden wall contact. They did so at target impedances with stiffnesses from 4 to 10 x 10^3 N/m, and damping ranging from 25 to 100 Ns/m. Similarly, Zhu and Barth [100] looked at impedance control for contact tasks, recording motion tracking (in free space) at 1.5 Hz, as well as a ramp response that is then obstructed by a wall. They used three sets of target impedances (m [kg], b [Ns/m], k [N/m]): (0.5, 200, 800), (1, 400, 1600), and (2, 400, 3200). While it wasn't clear how well the target impedance compared to observed dynamic behavior, they did note that the stiffer impedance controlled systems tended to track the ramp better prior to contact, and higher impedances resulted in higher steady-state contact forces. Shen & Goldfarb used two valves to simultaneously control force and stiffness of a pneumatic actuator in [76], and where able to achieve stiffnesses ranging from 2000-3000 N/m to about 14000 N/m, while tracking a position signal over a -20 to 20 mm range, using an 80 psig supply pressure. These results, which possess an extra degree-of-freedom (DoF) over the single valve model investigated here, should be considered upper and lower bounds for pneumatic stiffness, though they cautioned that a greater range might be achievable with a higher supply pressure. Their position errors were in the 5 - 10% range. Richardson et al. [67] observed a 5 mm tracking error for an impedance controlled pneumatic system, as well as step responses with 0.1 - 0.2 second rise times.

2.2.3.3 Pneumatic Tracking Control

The previous sections have noted position error and tracking RMSE. It is useful to define the standard for tracking in the absence of any additional aims (e.g., compliance).

Bone & Shing [11] offer the most widely cited (and arguably best) direct comparison of controllers for the aim of general, minimum-error position control. They compared a number of controllers to their own, which yielded errors under 1 mm over a 70 mm stroke, at speeds from 0.25 - 1 Hz, an improvement over comparisons that had RMSE errors of 2 - 3 mm. Researchers who look at step response measurements generally look at the steady-state error [93], which has been as small as 0.05 mm, though time constants can also be extracted and are generally as low as 0.2 s for closed loop position tracking. The effect of stroke length and speed was addressed somewhat by Lee et al. [52], tracking over a 70 mm range, they measured errors of 2-7 mm (3-10%) on tests varying from 0.2 - 0.5 Hz, and for a test at 0.2 Hz with 30-70 mm stroke, they measured 2-3 mm error. This last point may help to show that for tracking measured at the stroke, the actual error should be used for comparison, not necessarily the percentage error. This is not necessarily true if the stroke is amplified by some linkage geometry and position error is measured at an end effector.

2.2.4 Numeric Performance Targets

Using the previous research as a guide, performance target ranges were established:

1. **Tracking Signals**: At minimum the controller should track a sinusoidal reference over a majority of the stroke at up to 2 Hz with good accuracy (2-5 mm RMSE). Ideally, it should track these signals excellently (under 1 mm RMSE), and also track a sinusoidal reference up to 5 Hz well (up to 10% error, as a base).

2. **Disturbance Response**: The system should be backdriveable by a human pushing on the joint without exerting significant force. Significant force is based on heuristics: maximum forces applied by the elbow range from 50 100 N and 70 100 N by the shoulder [81], and injury from static collisions often occurs in excess of 50 N [48], so a threshold for significant force is set at 50 N. This is also 33% of the maximum

force for chest impact based on ISO standards for safe industrial robots [1, 40]. Force guidelines are fairly generous for the single DoF test case, as they're generally derived for safe robot-human interaction and are designed for larger scale industrial robots [40]; therefore, impedance targets must also be satisfied. Further, the system's response to disturbance may be assessed by seeing how it behaves when the disturbance is removed, using standards such as overshoot, settling time, steady state and RMS error.

3. **Target Impedance & Dynamic Stiffness**: The bounds from [76] of 2 - 14 N/mm can be used here, or can be calculated from the mechanical stiffness equation derived in [76]. Stiffnesses should be low – under 4 N/mm at the actuator, with damping constants under 100 Ns/m. Ideally, a high dynamic range of the closed loop stiffness should be achievable[1].

2.2.5 Controllers for Compliant Pneumatic Control

This thesis aims to define a controller for pneumatic systems capable of compliant position tracking; that is: position tracking that is as accurate as possible in the absence of interaction, but is compliant to accidental interaction. The aim is to do so via a controls solution, i.e. without requiring additional contact sensors, and on a standard single valve-single cylinder implementation.

One approach is to simply try and minimize error, and hope that the impedance or stiffness characteristics of the closed loop system fall into the target range. With this strategy, it is most logical to apply either an optimal controller (e.g., LQR/LQT), or a Sliding Mode controller. As noted in section 2.2.3.3, Sliding Mode Control (SMC) has consistently provided the most robust approach for achieving good tracking [11]. SMC uses a switched control law to drive a system to a sliding surface defined by the performance targets (e.g., zero state error), then keeps the state on the surface using a feedback-linearizing control input [78]. In fact, in a system with multiple inputs, the sliding surface could be

[1] A factor of 6 is good for pneumatics, factors of 10-100 are good for SEA comparison, and a factor of about 20 is about normal for VSAs

extended to include target impedance profiles, as was done by Shen and Goldfarb using a two-cylinder implementation [76]. However, for the SISO case, the use of the switching law, and the associated robustness, result in a high-impedance closed loop system and very large interaction forces, making it undesirable for compliant control.

A second approach would be to use an infinite-horizon optimal controller, such as the Linear Quadratic Regulator (LQR – or it's tracking extension) coupled with a low-level linearizing force controller. As long as impedance goals are sufficiently high and modeling error can be bypassed, this should minimize tracking error. The main challenge with LQR is finding the right balance of tuning parameters that both minimizes tracking error without requiring too aggressive an input.

Alternatively, a user could try impedance control, which sets uses an impedance filter to ensure a particular force-position profile for the system. Impedance control has been previously implemented on pneumatic systems intended for contact and interaction [67,90,100]. In impedance control, the user specifies a transfer function for the output impedance, $F(s)/x(s)$, typically as a mass-spring-damper model. The downside is that without additional force sensing, a low impedance results in a correspondingly slower system and higher tracking error. Further, impedance control is heavily reliant on the system model; depending on the linearity, the inherent model dynamics may be difficult to counter, thereby limiting the range of achievable impedances. Additional force sensing can also be used to filter inherent dynamics.

2.3 MPC for compliant position control

There is some mismatch among intent of researchers investigating compliant systems and safe robotics. While many researchers cite the need for a variable stiffness actuator, the stiffness is often varied according to some impedance profile, obtained based on somewhat arbitrary guidelines. By contrast, for industrially oriented applications, the minimum time optimal control problem is a natural motivating source for target impedances, but it doesn't apply especially well to physical therapy, walking, or manipulation that follows natural motions (natural motions can be characterized by optimality principles, but these typically are not minimum time [86]). Instead, there is a need for modulating stiffness to optimize tracking while still maintaining compliant dynamics, online, of a given path, and not in minimum time. In pneumatic systems, the compliance is inherently related to tracking control – in fact, in all closed loop systems, the output impedance is affected by the controller. Rather than using fixed gains, the control should be selected to minimize tracking error while maintaining compliance that satisfies some safety constraints. Optimal control has been used for off-line impedance control gain tuning [60], but reference tracking in human-motion scenarios often happens online and is subject to user safety constraints.

It is valuable to recognize this problem as a constrained optimal control problem (error minimization with impedance constraints). In fact, the time optimal problem seen in industry offers a clear parallel in the related field of safe, fast motion control: *The Safe Brachistochrone Problem.* This is the challenge of velocity-constrained, time-optimal control: given a system with [velocity] state constraints, determine a control strategy such that the closed-loop impedance/velocity combination satisfies some safety criterion, e.g. the HIC, and completes the point-to-point motion in minimum time. The general solution consists of stiff, slow motion and flexible, fast motion, and has been cited by numerous researchers in the compliant motion domain [27, 41, 46]. Bicchi et al. further investigated a number of optimal control solutions to variations on the safe brachistochrone problem [36], though of course they all deal with a minimum time solution.

By recognizing the parallels between safe time-optimal control and compliant tracking (safe position-error-optimal control), it is clear that a constrained optimal controller would be a logical solution choice. Model Predictive Control is therefore a clear solution choice; MPC enables real-time constrained position control.

2.3.1 MPC Background

The basic implementation to MPC is that a system model is used to solve a finite-time optimal control problem over a time horizon termed the *prediction horizon*, N_p. The control input is assumed to have a length $N_c <= N_p$ – the *control horizon.* This control problem is solved at each time-step, updating state and model information as time progresses, but only ever using the first control input. A thorough overview of MPC implementation, stability, robustness, and other properties is provided in the seminal 2000 survey paper by Mayne, Rawlings, Rao, and Scokaert [59]. Lee's 2011 survey paper on MPC provides more recent updates [53] to the application domain, which has been aided through implementation-oriented tutorials, textbooks, etc. [12, 14, 95].

For the unconstrained, linear case, an analytical solution exists; for most practical realizations, however, a numerical solver is used to solve the constrained optimization problem. Constrained control is important; results from [28] showed that for control of a system with input saturation, an [unconstrained] Linear Quadratic Regulator (LQR) could lead to instability due to lack of constraint awareness, whereas a constrained MPC resulted in better and stable performance.

Historically, Model Predictive Control, and the related control variety, Receding Horizon Control (RHC) originated in chemical process control applications – slow systems with fixed steady-state setpoints, where fast optimization is not critical. However, recent advances in optimization [59, 98], as well as explicit MPC[2] [3] have broadened the application domain. As observed in section 2.1.1, MPC for pneumatic applications has been limited to a few specific cases, and is largely chosen to include operating constraints in position and force control aims.

2.3.2 Predictive Friction Compensation with MPC

Predictive control offers a feasible way to design an optimal friction compensation strategy. MPC may be coupled with feedforward or mitigation techniques like those discussed above [20], but a more appealing planning-based method can be used in the form of a disturbance observer [62]. Disturbance models augment the state matrix with estimates of the unknown parameters, described as a series of known discrete values over the prediction horizon that are then included in the solution to the finite time optimal control problem. In this thesis, it is suggested that by using cascaded predictive controllers, state estimates can be found that are used as inputs to an observer that estimates friction over the prediction

[2] where as solution is computed off-line and a lookup table is used to implement the solution to the constrained optimization

horizon. This can then be used as a disturbance model in a second predictive controller.

This approach bears resemblance to two past friction compensation methods. Runzi et al [72] modeled friction as a periodic disturbance and used a predictive controller coupled with repetitive compensation. A disturbance model was updated at each friction occurrence by subtracting the error, converging to a good estimate over several iterations. The planned nature of MPC is also similar to impulse control, which works by supplying properly timed impulse inputs to overcome the stiction effect [5]. Researchers have also designed general adaptively updated disturbance models intended to compensate for friction, among other properties, e.g. with stochastic estimation techniques [62], but did not explicitly model or address friction.

2.3.3 Configuration and Research Aims

It is evident that MPC can offer a novel take on design for compliant position control. The primary assets of MPC are that it enables online solution of a constrained optimal control problem, and because its predictive structure is conducive to improving friction compensation for slow systems.

In this thesis, the MPC framework will be leveraged to solve the minimum error tracking problem while subjecting the system to force and impedance constraints. However, present tools for realtime control with MPC largely require either custom, specialized solvers, or linearized systems. Similarly, constraints must be designed that are straightforward, scientifically justified, and effective. MPC also comes with trade-offs; it is computationally expensive and requires a fairly involved implementation. Therefore, it is necessary to examine how much MPC can affect control, and how it compares to controllers with similar or related aims, both at the single- and multi-DoF implementation level. The following research goals will be discussed in subsequent chapters:

- **What kind of system models are sufficient for control?** Pneumatic system representation will be addressed and simplified to a form that is usable in controllers. The model must be valid over a prediction horizon and sufficiently describe the desired performance characteristics of the system.

- **What MPC configuration is most effective in the desired target behavior?** Predictive controller designs will be put forth that combine low-level force control with appropriately selected constraints.

- **Can predictive estimation be used to improve friction compensation relative to additive methods?** A predictive estimator will be developed and implemented to calculate friction disturbance over the prediction horizon based on expected state behavior. It will be compared to traditional friction compensation techniques through feedforward terms based on an instantaneous model, and assessed both analytically and in practice.

- **How does performance of an MPC-controlled system compare to that of benchmark controllers?** The controller is compared to benchmark controllers that offer different balances in the categories of tracking, safety, and ease-of-implementation. Performance is examined using well-established tests and over a range of tuning and operating points in simulation, as well as at discrete checkpoints in hardware, and trade-offs are discussed.

- **How effective is MPC for the multi-DoF hardware implementation?** The predictive controller is applied to a two-DoF robotic arm and subjected to some application-based tasks.

- **What stability and performance statements can be made about the MPC-controlled system?**

CHAPTER III

PNEUMATIC ACTUATOR DYNAMICS & MODELS

A thorough understanding of system dynamics and key behavior exhibited by the plant is critical to successful model-based control. This chapter first introduces the standard nonlinear approach to modeling pneumatic actuator dynamics, then addresses detail variations for high fidelity models. The resultant model has been used to develop a MATLAB/Simulink simulation for controller development and analysis, which will be validated and will continue to be referenced throughout the thesis. Lastly, some simplifying assumptions that improve the model for use in control will be explored.

3.1 Core State Equations

A model for the actuator was developed using the "standard" academic model for a pneumatic actuator [2, 77, 84], based on a physical setup consisting of a valve and cylinder with position and pressure feedback.

The actuator model has four primary components: a conversion of voltage input to orifice area, an equation for mass flow through an orifice, an energy balance to relate mass flow to change in pressure, and a force balance to derive output force from the pressure and friction. The resulting model is a nonlinear, third-order system. A block diagram is shown in Figure 3.

3.1.1 Pressure Dynamics

Mass flow, $\dot{m}$, is calculated as a function of upstream and downstream pressure, P_u and P_d, respectively, orifice area, $A(u)$, a discharge coefficient, c_d, and several predefined constants, and is of the form

$$\dot{m} = A(u) c_d \Psi(P_d, P_u) \tag{1}$$

The choice of upstream and downstream pressure is dependent on the direction of flow, as defined by the valve command. For each chamber, the mass flow can be defined as

$$\text{Discharging (flow out of the chamber): } P_d = P_{Supply}, P_u = P_{Chamber} \tag{2}$$

$$\text{Charging (flow into the chamber): } P_d = P_{Chamber}, P_u = P_{Atmosphere} \tag{3}$$

The function $\Psi(P_d,\, P_u)$ is piecewise-defined and depends on critical pressure ratio, $P_{cr} = (P_d/P_u)_{cr}$, which determines whether the system is experiencing choked or un-choked flow ($P_{cr} = 0.528$ for air):

$$\text{For } P_d/P_u > P_{cr} \text{ (un-choked flow): } \Psi(P_d, P_u) = C_1 \frac{P_u}{\sqrt{T_u}} \left(\frac{P_d}{P_u}\right)^{1/k} \sqrt{1 - \left(\frac{P_d}{P_u}\right)^{(k-1)/k}} \tag{4}$$

$$\text{For } P_d/P_u <= P_{cr} \text{ (choked flow): } \Psi(P_d, P_u) = C_2 \frac{P_u}{\sqrt{T_u}} \tag{5}$$

where T_u and T_d refer to upstream and downstream temperature. These quantities are calculated using the ideal gas law and the instantaneous total mass and pressure in the cylinder

$$T = \frac{PAx_{abs}}{mR} \tag{6}$$

A is the cross-sectional area of the chamber and x_{abs} is the absolute position of the rod, defined as the distance of the piston from the end caps, i.e. the chamber length (this distance includes the dead space inherent to each chamber of the cylinder). The constant terms C_1 and C_2 are functions of the universal gas constants R and the ratio of specific heats k, (k = 1.4 and R = 287 J/Kg K for air):

$$C_1 = \sqrt{\frac{2k}{R\,(k-1)}} \quad \text{and} \quad C_2 = \sqrt{\frac{k}{R\left(\frac{k+1}{2}\right)^{(k+1)/(k-1)}}} \tag{7}$$

The previous equations govern the dynamics of the valve. The next step, an energy balance, defines the link that connects mass flow through the valve to changes in pressure within the chambers of the cylinder. Under an adiabatic assumption, the change in pressure in each chamber of the cylinder can be found using the ideal gas law:

$$\dot{P} = \frac{kRT\dot{m}}{x_{abs}A} - \frac{P\dot{x}}{x_{abs}} \left(\frac{kR}{c_p} + 1\right) \tag{8}$$

where c_p = 1012 J/Kg K is the specific heat of room temperature air. Equation (8) is applied to both sides of the cylinder and integrated to get rod-side and cap-side pressures .

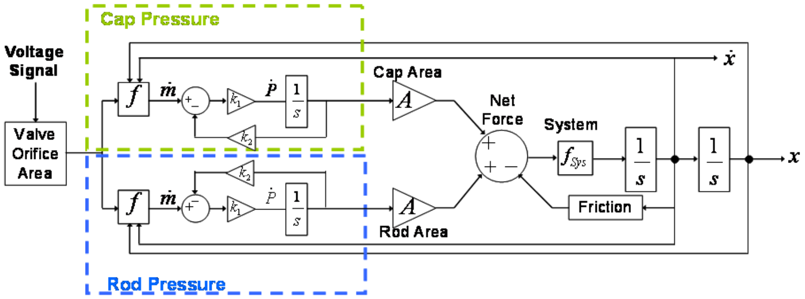

Figure 3: Actuator block diagram.

3.1.2 Force and Position Dynamics

A force balance is used to determine the amount of force exerted by the actuator. There are some losses that affect the actual force experienced by the system. The most critical of these is friction, which is dependent on the internal construction and lubrication of the cylinder, as well as the rates at which the system is excited, and will be discussed further in section 3.2.2. The net force is found using equation (9):

$$F_{net} = P_{cap}A_{cap} - P_{rod}A_{rod} - P_{atm}A_{piston} - F_{friction} \tag{9}$$

where A_{cap} and A_{rod} refer to the cap-side and rod-side cylinder cross-sectional areas, A_{piston} refers to the area of the piston, and the pressures used are absolute.

3.2 Detail Modeling Variations

While the core state equations are fixed, accurate representation of pneumatic dynamics also relies heavily on a several components that can be subjected to significant tuning and adjustment. Notably, these include valve and friction dynamics.

First, the discharge coefficient, c_d, and orifice area, $A(u)$ are closely related and essentially describe valve behavior. Additionally, constraints on valve bandwidth and response speed must be incorporated into a realistic simulation model. Friction also plays a significant role in the position dynamics, and can be described in a number of ways that generally require a balance of accuracy and ease-of-implementation.

3.2.1 Valve Dynamics

In section 3.1.1, the valve dynamics are described by a discharge coefficient, c_d, a signed orifice area, $A(u)$, and a flow function, $\Psi(P_u, P_d)$. The pressures depend on direction of flow, as in equation (2), and are chosen using an overlapped valve assumption: the spool is larger than the orifice, so each chamber can be hooked up to either the supply or the exhaust; never both.

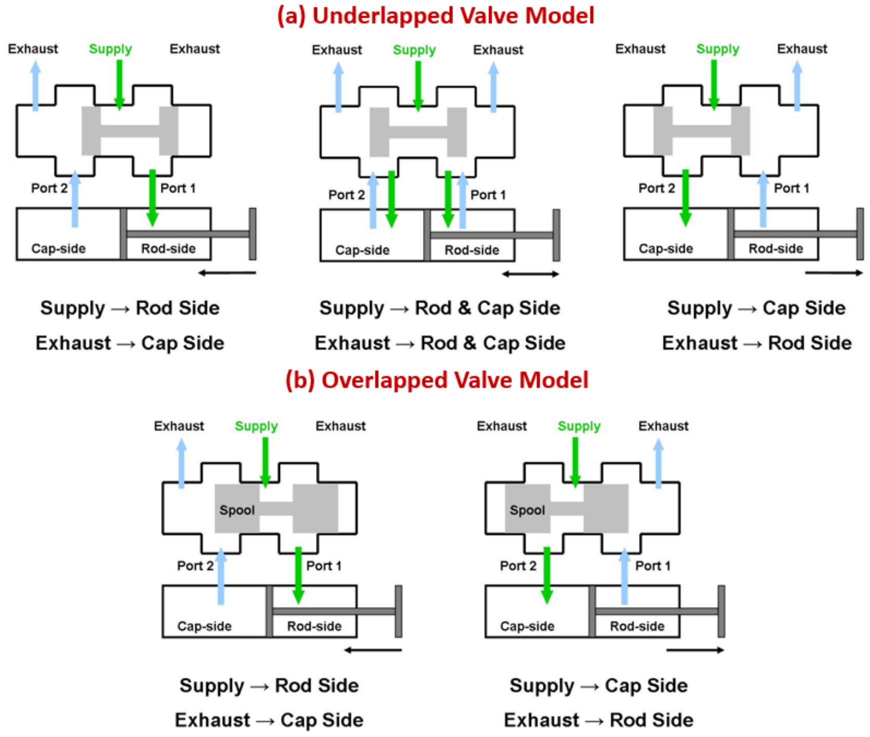

Figure 4: Comparison of underlapped (top) and overlapped (bottom) valve models.

$A(u)$ is modeled as a function of voltage that is zero at the offset voltage (the voltage about which the valve is centered), and is then signed based on the direction of flow. However, because the valve is under-lapped, this approach fails to cover the range of actual dynamics, as seen in Figure 4. Instead, Figure 5, which displays a characteristic pressure response to a voltage step input near the offset voltage: chamber pressures settle to steady state values before the limiting values – supply and exhaust – are reached. This behavior requires that (1) $|A(u)|$ not equal zero at any time, and (2) $\dot{P} = 0$ before either supply or

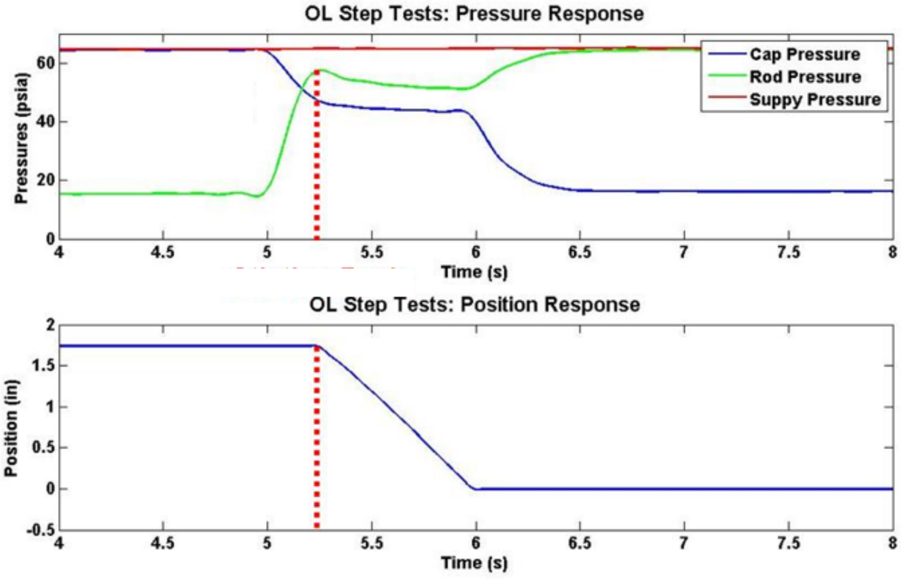

Figure 5: Characteristic pressure & position response to a voltage step. The red line shows the point at which stiction is overcome and motion begins.

exhaust are reached, which is impossible to achieve using an overlapped model.

Past researchers have dealt with this problem in different ways. Many simply ignore it, especially those constructing a model for control, rather than simulation. Others use a linear estimate of $A(u)$ and an offset area, as in Figure 6, which reduces error. However, the affected region spans approximately +/- 0.75 V about the valve offset and has been shown to be very active in servo control tasks, so its dynamics cannot be simplified without first investigating effects on model integrity.

Since the exact geometry of the changing orifice area, $A(u)$, was initially unknown, an equivalent orifice area, A_{eq}, was used instead. A_{eq} is the product of discharge coefficient and signed orifice area, $A_{eq} = c_d A(u)$, which is easily fitted to the known model data.

The area modeling approaches must also drive $\dot{P}$ to zero before supply or exhaust are reached. Four situations were investigated to model this loss of mass flow:

1. Assume a constant mass flow loss and offset the overall $\dot{m}$ function by this amount
2. Assume zero mass flow loss except when cylinder does not fully charge/discharge
3. Model net mass flow as a sum of positive and negative flows through a more accurate

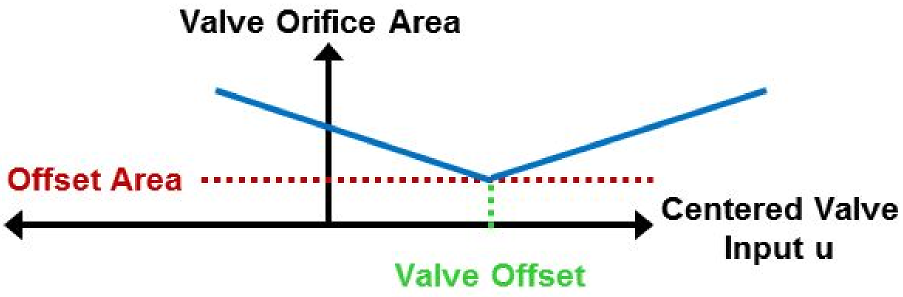

Figure 6: Linear area model with offset area.

valve orifice model

4. Ignore the transients and focus on the steady state values by changing the magnitude of the valve's input port pressures.

All of the modeling approaches were tried. Methods (1) and (2) worked consistently only for a small range of pressures; trends were difficult to observe. Method (3) relies on an exact knowledge of the valve geometry and the inner flow dynamics, which was unavailable. Ultimately, method (4) proved to be the most successful. This approach treats supply and exhaust pressures as continuous functions of the voltage instead of discrete values that switch at the offset voltage.

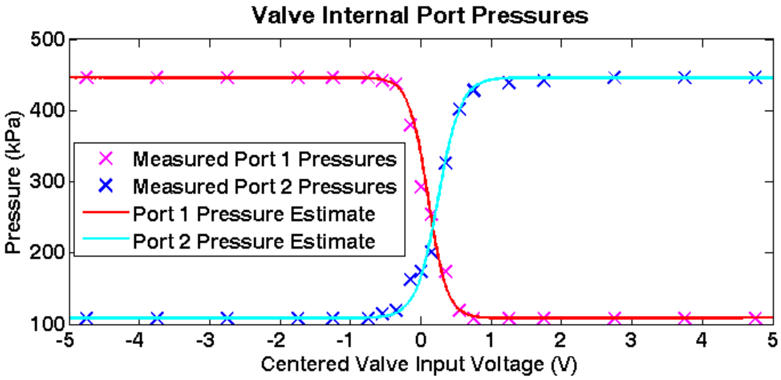

Figure 7: Input pressure trends.

Pressure curves were found by measuring the steady state pressure in each chamber that resulted from a series of voltage step commands to a cylinder with a fixed piston (so motion dynamics didn't play a factor). Figure 7 shows the functions used to model pressure and the data points to which they were matched. A hyperbolic tangent function was found to model observed trends:

$$P_{input} = y_{offset} + C_{scale} tanh\left(K_{Press}\left(u - x_{offset}\right)\right) \tag{10}$$

where the constants are defined in terms of the supply pressure, P_S, the exhaust Pressure, P_e, the low Pressure cutoff, C_{LP} (the voltage/x-axis value at which curve transitions from the constant minimum pressure to a changing one), and the high pressure cutoff, C_{HP} (the voltage/x-axis value at which curve transitions from the constant maximum pressure to a changing one). These constants are defined as $y_{offset} = \frac{1}{2}(P_S + P_E)$, $x_{offset} = \frac{1}{2}(C_{HP} + C_{LP})$, $C_{scale} = \frac{1}{2} sign(C_{HP} - C_{LP})(P_S - P_E)$, and $K_{Press} = 2\pi/|C_{LP} - C_{HP}|$.

To determine equivalent orifice area curves, step tests were run on the cylinder test rig, always starting from the opposite voltage limit and then calculating equivalent orifice area from the observed steady state behavior. The resulting equivalent area curves, seen in Figure 8, were fit using fifth order splines for use in simulation. However, the curves also match an orifice area model $A_{eq} = c_d A$ that uses two constant, chamber-specific values for c_d and an orifice area model for A based on expected geometry: A_{geom} is the area of a circle segment:

$$A_{geom} = R_O^2 cos^{-1}\left(\frac{R_O - h_{spool}}{R_O}\right) - (R_O - h_{spool})\sqrt{2R_O h - h_{spool}^2} + Y_{corr} \tag{11}$$

where h_{spool} is the segment width affected by the advancing spool, mapped from input voltage u by constants C_1 and C_2: $h = C_1 u + C_2$. R_O is the orifice radius, and Y_{corr} is a correction factor since the curves are not exactly centered. Values were found using approximate measurements of the geometry as initial values and then computing a least squares fit. From a simulation perspective, either curve produces approximately the same detail-oriented result.

3.2.2 Friction Modeling

To avoid issues with stiff solvers, a continuously differentiable friction model was used to represent Stribeck and viscous effects. This model is based on a hyperbolic tangent curve,

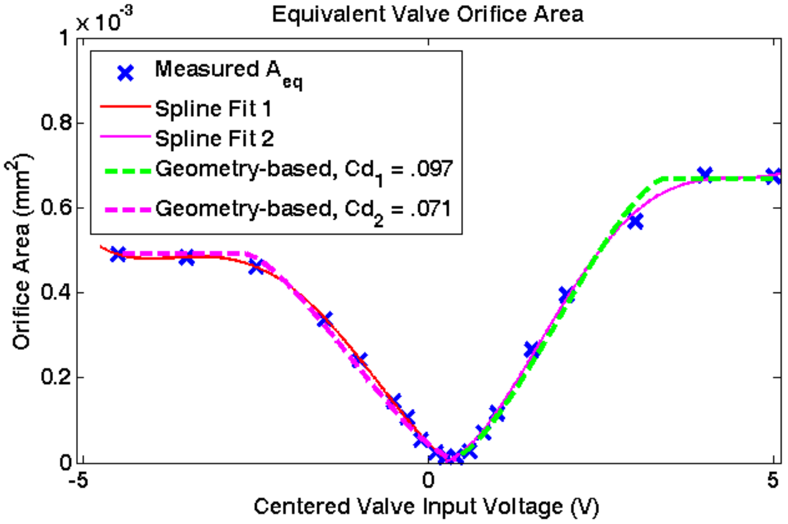

Figure 8: Equivalent area curve fit for one valve port.

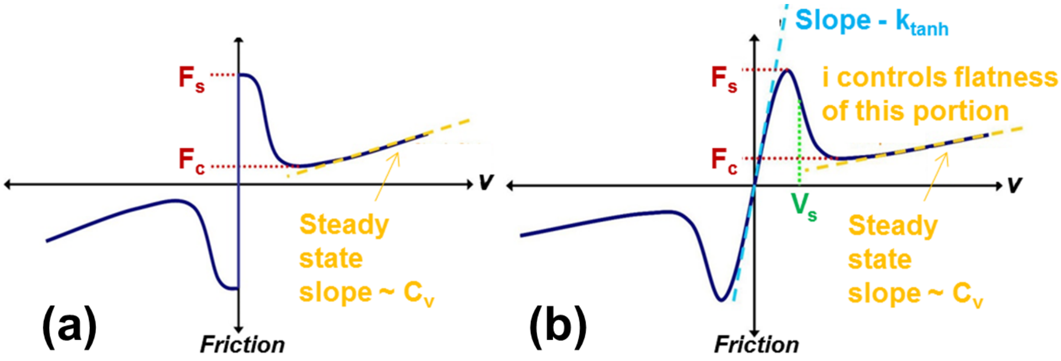

Figure 9: General form of friction as a function of velocity: (a) Stribeck friction model and (b) Stribeck-Tanh friction model.

defined by equation (12):

$$F_{Friction} = \left[F_C + (F_S - F_C)e^{-(|v|/v_S)^i}\right] tanh(k_{tanh}v) + C_v v \tag{12}$$

The effects of the parameters on the shape of the curve are seen in Figure 9. F_C and F_S represent Coulomb and static friction, respectively, v is the velocity, and v_S is the sliding speed coefficient, which sets the velocity at which stiction transitions to other forms of friction. The exponent, i, affects the rate at which slope changes from the post-stiction drop to steady state viscous and Coulomb friction. The slope between the friction peaks is governed by k_{tanh}, and C_v is the coefficient of viscous friction.

F_C, F_S, and C_v were found experimentally using a series of open loop step tests at different supply pressures and in various orientations. A force sensor measured the observed force, and pressure sensors provided the differential force in the cylinder. Stiction measurements were derived from the force difference at the start of the motion, and the viscous friction was plotted as a function of steady state velocity so that curves could be fit to the data. Once Coulomb, Viscous, and Static friction parameters were found, the remaining parameters were matched to the data based on the overall fit, as detailed in [25].

One challenge with friction is that it does not necessarily stay constant in time, due to lubrication and wear. This research spanned several years, and multiple cylinders, and while the model was found to be a consistently strong representation of the stiction behavior, the exact parameters varied. In the original analysis seen in [25], the final model parameters used $F_S = 20.0$ N (4.5 lbf), $F_C = 13.3$ N (3.0 lbf), $C_V = 4.4$ Kg/s (0.5 lbf·s/in), $i = 5$, $V_S = 0.1$, and $k_{tanh} = 1580$ s/m (40 s/in). The accuracy of that friction model is shown in Figure 10. A more recent fit, used in the simulations in the later chapters of this Ph.D., found that the magnitude had decreased: $F_S = F_C = 5.3$ N (1.2 lbf), $C_V = 5.3$ Kg/s (0.6 lbf·s/in), and held all other parameters to their earlier values.

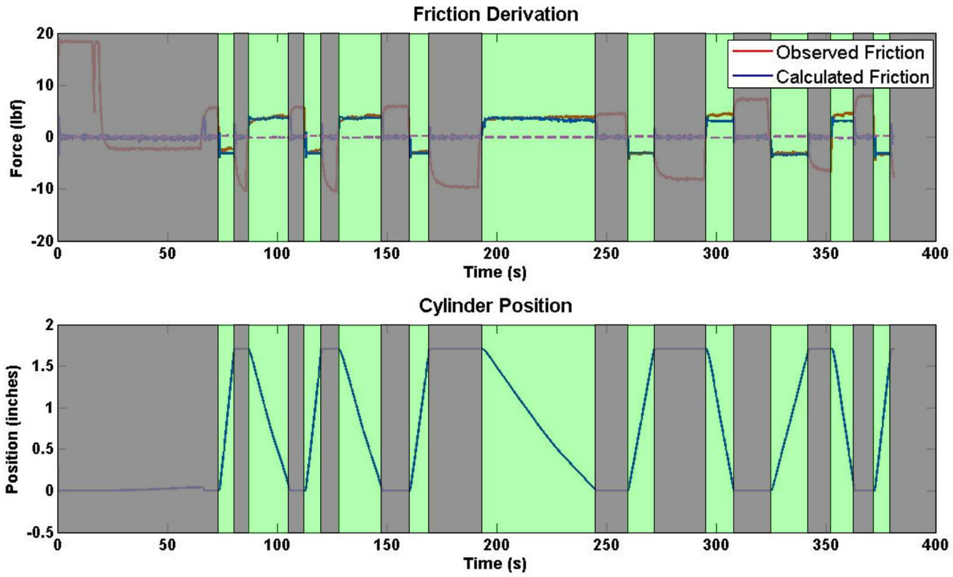

Figure 10: Performance of friction model. Regions during cylinder motion are highlighted in green.

3.3 Model Validation & Hardware Performance

The model was validated against numerous physical comparisons in both open and closed loop settings, exhibiting a strong resemblance between position, velocity, and pressure states, particularly in the operating region close to the valve equilibrium, where most closed loop control took place. Two primary testbeds were used:

- The hardware testbed found in [24–26] and shown in figure 11. This variation was predominantly used for simulation validation, including numerous tests at various levels. The actuator uses a FESTO MPYE-M5 5 port, 3-way valve connected to the chambers of a Bimba PFC low friction cylinder with internal position feedback. Two SSI P51 pressure sensors monitor chamber pressure. A third pressure sensor measures supply pressure near the source.

- The testbed seen in figure 12, cited in [21–23] and used for 1-DoF controller verification in subsequent thesis chapters. This testbed uses a cart mounted atop an air-bearing to ensure that horizontal motion is free of side-loading a friction due to external sources. The same pressure sensors used in the previous testbed are mounted at each chamber (supply pressure is unmeasured, but a regulator is located just before the input to the valve), and the Bimba-PFC actuator has its own internal position sensing, as in the previous testbed. Additionally, a shaker, on the left, may apply external axial loading, and interaction forces are measured by a force sensor. The two are connected by several inches of plastic 10-32 threaded rod, which will bend and snap if the system overloads or fails, to avoid damage to the shaker, sensor, and actuator.

Both platforms were run using a target computer equipped with xPC Target / Real-Time Workshop, MATLAB's real-time operating system. For the most recent set of single degree-of-freedom tests, since tests were performed on a borrowed shaker, a mobile platform needed to be designed, and a separate target was used. Thus, while the first system and multi-DoF platform used a PC104 with 2 GB of RAM and integrated D/A and A/D cards, the single DoF tests were primarily operated on a desktop PC connected to a Quanser Q8 Board. The details of these platforms is provided in appendix A.3.

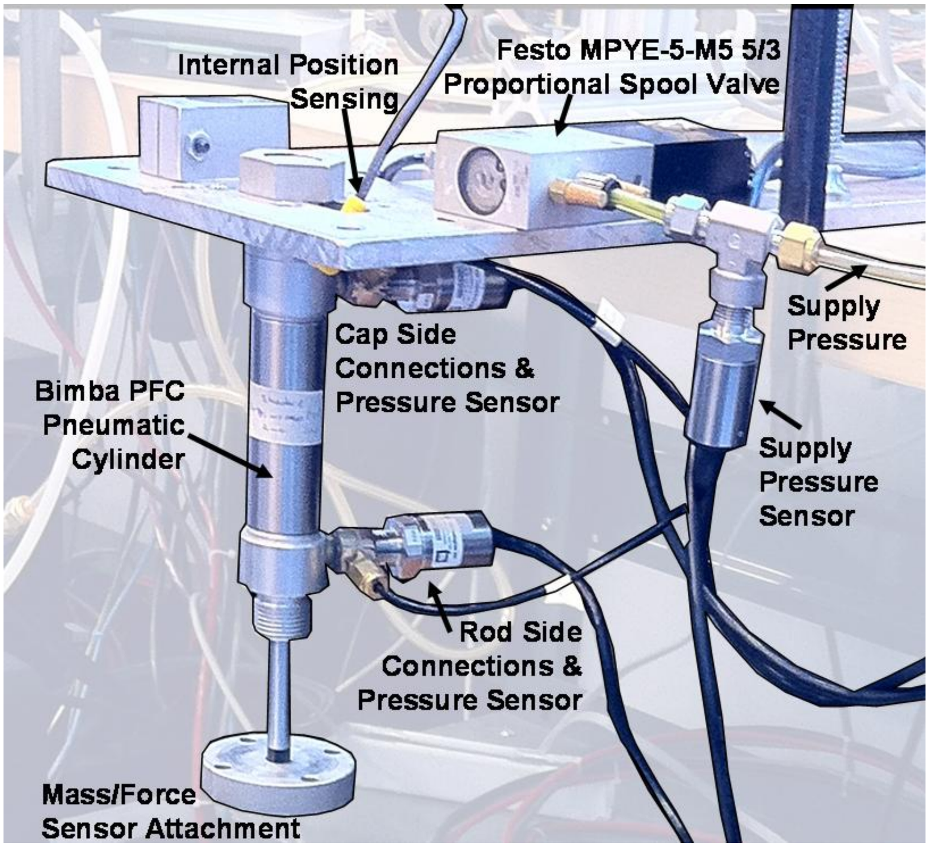
Internal Position Sensing
Festo MPYE-5-M5 5/3 Proportional Spool Valve
Supply Pressure
Cap Side Connections & Pressure Sensor
Bimba PFC Pneumatic Cylinder
Supply Pressure Sensor
Rod Side Connections & Pressure Sensor
Mass/Force Sensor Attachment

Figure 11: Hardware testbed used for original model validation. Modifiable to vertical and horizontal configurations. Not shown: mass and force sensor.

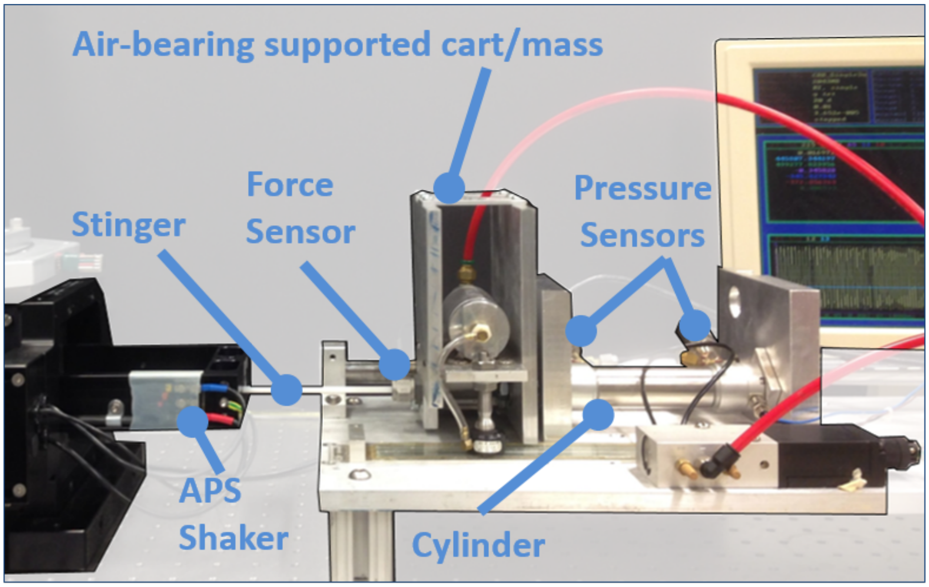

Figure 12: Hardware testbed used for 1-DoF benchmark and MPC tests in later chapters of this thesis.

3.3.1 Primary Simulation Validation Tests

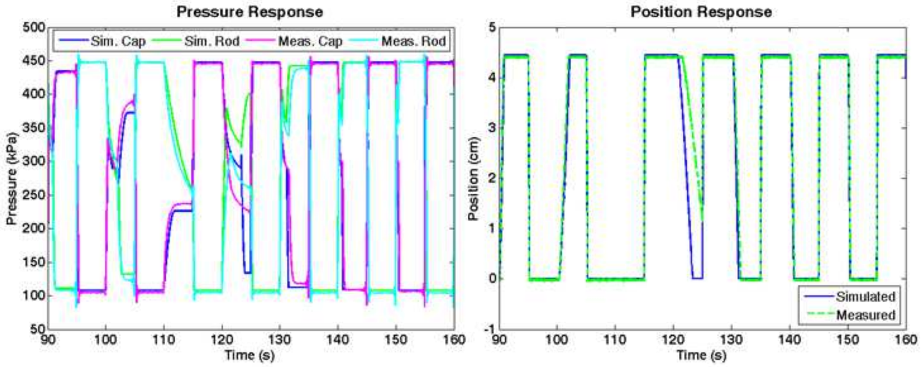

Figure 13: Actuator model validation: response to sequence of open loop voltage steps from 6 to 4 V.

An initial set of simulations focused on the individual actuator model, validated on the original test rig. Figure 13 displays position and pressure response to a series of open loop voltage step inputs, including points not used for curve fitting earlier. These results show that the adjusted orifice area model represents observed trends accurately. Simulated and actual dynamics match closely, even without feedback control, and in regions near the valve offset. The accuracy of the friction model and pressure constants can be viewed by looking at the time it takes for the position to begin changing (or its resistance to change if the pressures are not high enough), the pressures needed to do so, and the duration of the step thereafter. The only place where position correspondence deviates significantly is exactly at the offset voltage, 5.25 V. This is viewed as an outlier, particularly since the curves near it demonstrate good correlation of measured and simulated values. These methods also lead to high accuracy within a PID control loop on position, as seen in Figure 14, which demonstrates simulated and actual results of sine tracking at various frequencies. Figure 15 shows similarly well-matched results for a closed loop position step reference. The model exhibits rise and settling times near those of the actual system, as well as oscillations characteristic to a third order system, as seen in the zoomed-in step response.

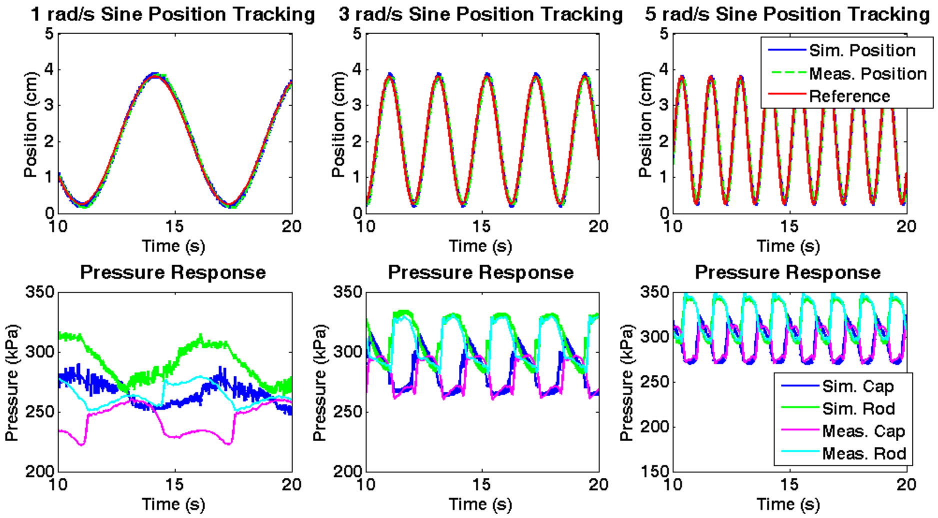

Figure 14: Actuator model validation: sine tracking.

3.3.2 Simulation and Second 1-DoF Hardware Testbed

Because this project lasted for several years, the hardware testbed and properties changed somewhat over time. A secondary validation was carried out to demonstrate the validity of the updated simulation on the newer testbed.

The changes were limited to three main areas:

1. **A transition from a continuous model to an entirely discrete one:** While the simulation above was run at 1000 Hz with continuous simulation states (but continuous and discrete controller states), the later version was entirely discrete for more robust numerical results and controller interaction. To better represent the continuous nature despite the transition, the simulation was run at 5000 Hz, while controllers were simulated at 100 - 300 Hz.

2. **Addition of valve spool dynamics:** In most pneumatic actuator models, it is assumed that the spool can move infinitely fast; that is, it has no dynamics. However, Festo provides a limiting spool bandwidth of 125 Hz for free motion (presumably without any pressurized air flow), and in frequency response experiments, it has been observed that an additional integrator must be present above 30 Hz. Therefore, the

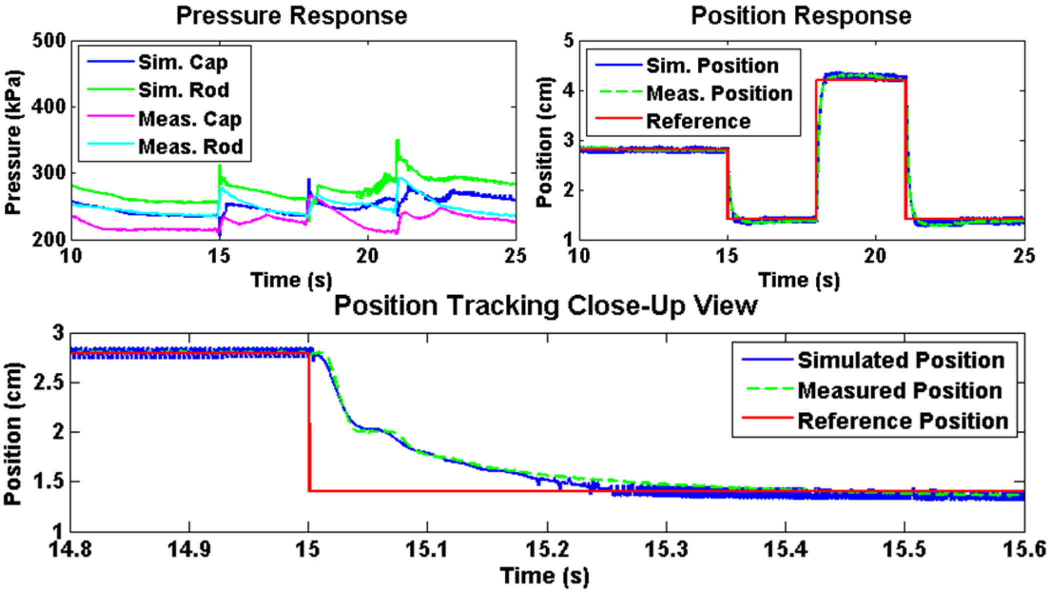

Figure 15: Actuator model validation: step reference.

spool subject to airflow was assumed to act like a first order low-pass filter with a 45 Hz cutoff frequency, found by best fit over several frequency response tests. In general, 45 Hz is well outside the achievable bandwidth of a pneumatic controller, but it may still affect simulation validity, so the dynamics were included in the model.

3. **Updated friction parameters:** Because seal and lubrication properties vary with time, friction parameters are subject to change. The old values were tested and updated based on time-domain tests, resulting in new Coulomb and Static friction magnitudes (the underlying velocity relationships were left as originally derived): $F_S = F_C$ $= 5.3$ N (1.2 lbf), $C_V = 5.3$ Kg/s (0.6 lbf·s/in).

The simulation was validated by comparing the outputs to measured hardware data. Figures 16 shows the position tracking responses of the actuator controlled by a PID controller with identical gains in hardware and simulation. It is evident that the behaviors are very similar and, in particular, the reader should note the qualitative behavior: the time constant and overshoot of the step response match, as do the magnitude and time of the delay caused by friction in the step and sinusoidal responses. A more comprehensive validation is seen in Figure 18, which shows measured and simulated frequency response plots

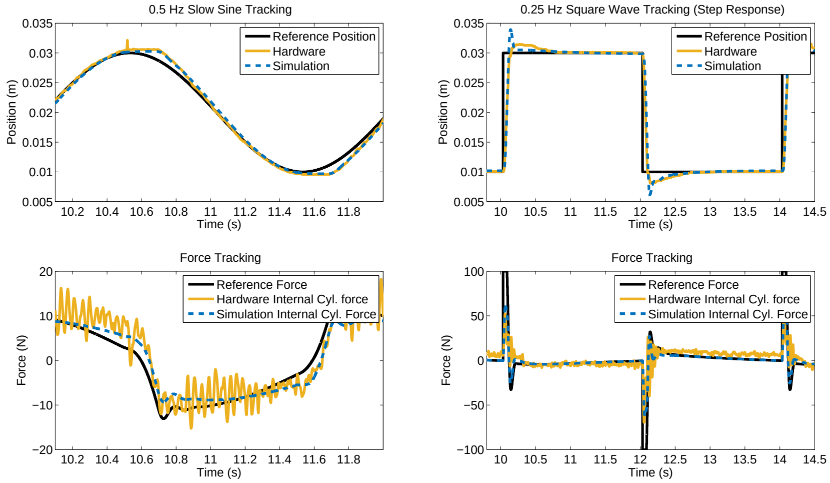

Figure 16: Position PID tracking step response.

for several important transfer functions, found using MATLAB's `tfestimate()` function. First, in Figure 17, the open loop force dynamics are validated: by exciting the voltage with a chirp signal and measuring the corresponding force signal, the transfer function for the force dynamics is obtained. A simple PD controller was used to shift the voltage signal to keep the actuator piston from hitting the end stops. Figure 18 illustrates the similarity in performance of the closed loop system. The PID controller is the same as that used in Figure 16, and which will be used as a PID controller later in this paper. The controller has two loops – the first controls force; its transfer function is displayed on the left in Figure 18. The second PID control loop is on position, and has similarly well-matched dynamics between the hardware and simulation, as seen in the figure on right. In all cases, the magnitude behaviors match well over the frequency spectrum, and while phase is also similar, it tends to deviate in the high frequency range. These effects are likely due in part to the effect of the hardware testbed support on the hardware; the hardware stand is clamped to a table that, at medium-high frequencies, begins to shake with the hardware stand. However, the behavior of the system at high frequencies is unimportant: in this thesis, the focus is on human-like tasks (under 5 Hz bandwidth) and pneumatic controller performance – which is typically under 10 Hz (in literature, high-performance pneumatic tracking can be found up to about 15 Hz).

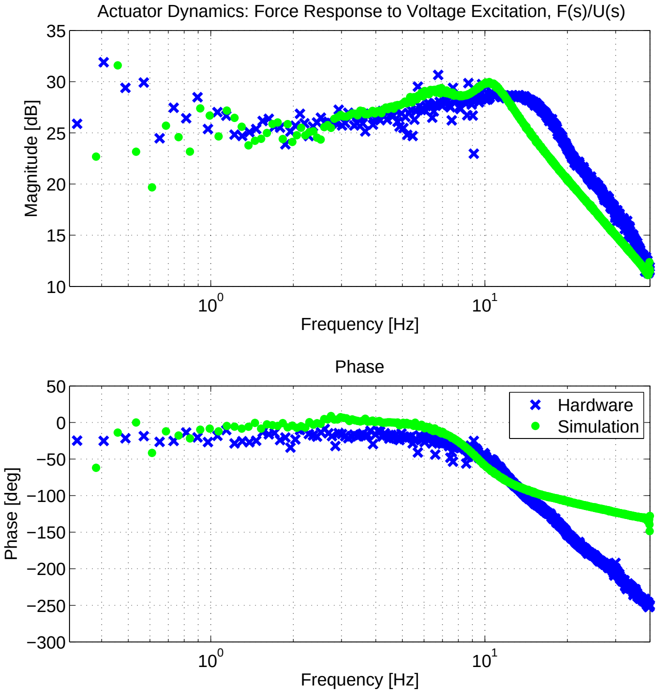

Figure 17: Effective open-loop frequency response validation: force/voltage. A PD control loop is used to prevent the system from hitting the end stops.

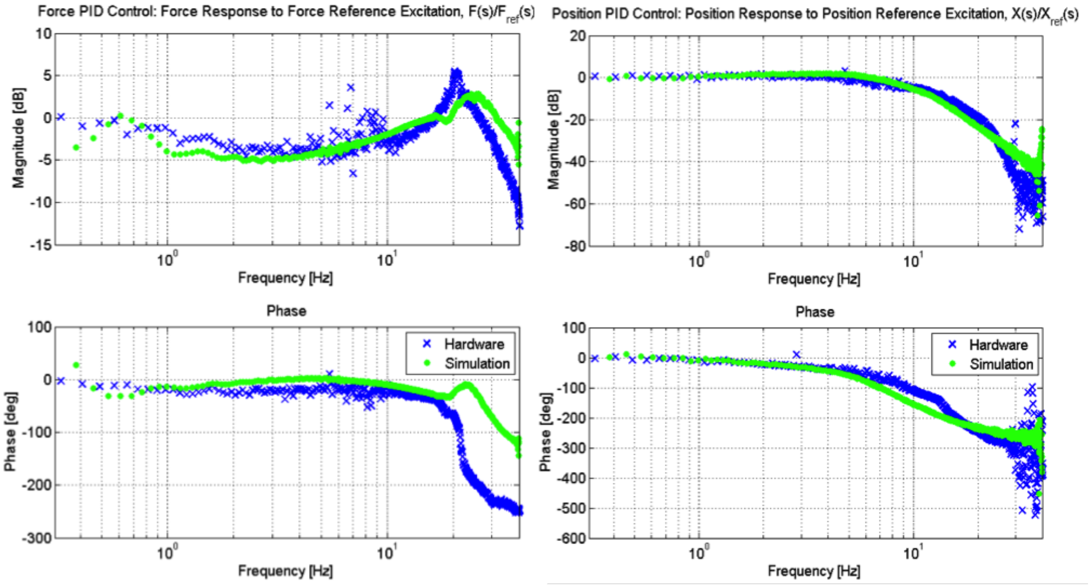

Figure 18: Closed loop frequency response validation – hardware results are shown in blue, while simulation data is in green. From left to right: (1) force PID (F/F_{ref}), and (2) position PID (X/X_{ref}) frequency response.

3.4 Alternative Models for Control

The standard model works well to accurately describe the behavior of a single-ended pneumatic cylinder controlled by a 5/3 servo-valve, following the typical configuration. However, the model includes multiple discrete components and other nonlinearities that make it inappropriate for control application. Instead, models are desired that are linear, continuous, and represent the true dynamics as closely as possible.

To derive these models, three primary steps are necessary: (1) eliminate unnecessary nonlinearities, (2) address friction, and (3) determine appropriate methods for system linearization.

The complete state dynamics for a system connected to chambers a and b can be described by a system with states $x_1 = x_a = L - x_b$, $x_2 = \dot{x}_a = -\dot{x}_b$, P_a, and P_b, representing the piston position and velocity, and pressures in chambers a amd b. The input is given by voltage, u. In abbreviated form, this system is shown to be:

$$\dot{x}_1 = x_2 \tag{13}$$

$$\dot{x}_2 = (P_a - P_b)/M_{sys} \tag{14}$$

$$\begin{aligned}\dot{P}_a &= \frac{kRT\dot{m}_a}{x_a A} - \frac{P_a \dot{x}_a}{x_a}\left(\frac{kR}{c_p} + 1\right) \\ &= \frac{kRT_a}{A_a x_1} g_a(u) c_{d,a} \psi_a(P_d, P_u) - \frac{P_a x_2}{x_1}\left(\frac{kR}{c_p} + 1\right)\end{aligned} \tag{15}$$

$$\begin{aligned}\dot{P}_b &= \frac{kRT\dot{m}_b}{x_b A} - \frac{P_b \dot{x}_b}{x_b}\left(\frac{kR}{c_p} + 1\right) \\ &= -\frac{kRT_b}{A_b(L - x_1)} g_a(u) c_{d,b} \psi_b(P_d, P_u) + \frac{P_b x_2}{(L - x_1)}\left(\frac{kR}{c_p} + 1\right)\end{aligned} \tag{16}$$

In the above equations, M_{sys} is the mass of the system being actuated, and the signed orifice area, $g_a = -g_b$, is a function of the input, u. Temperatures can be assumed constant or may be state-dependent. For the latter case, the following equations are needed:

$$T_a = \frac{P_a A_a x_a}{mR} = \frac{P_a A_a x_1}{mR} \tag{17}$$

$$T_b = \frac{P_b A_b x_b}{mR} = \frac{P_b A_b (L - x_1)}{mR} \tag{18}$$

3.4.1 Isolation of Unnecessary Nonlinearities

In the equations above, there is a direct, one-to-one mapping from the signed orifice area, $g_a(u) = A_{eq}(u)$, to the input voltage, u, as defined in Section 3.2.1 with splines and equation 11. Since the mapping is clearly defined, it is sufficient to treat the equivalent orifice area as the input in lieu of the voltage and simply convert the input as necessary. In practice, this is easily done using zero-finding methods, such as Newton-Raphson.

3.4.2 Friction

Rather than address friction in the model, the novel controllers developed in this thesis will use explicit friction compensation: additive or MPC-based compensation will be used to counter the effects of friction. Therefore, friction can be removed from the model used for control.

3.4.3 Linearization

There are two common approaches for linearization. The first is to use a closed loop controller to attempt to cancel out the nonlinearities. The closed loop system can then be approximated by some third or fourth order system (depending on the order of the valve dynamics) obtained via a system identification procedure or model-based transfer function. This will be discussed in Chapter 4 using input-output linearization and PID control. While straightforward, however, low-level feedback control is a non-ideal solution: it may alter the system dynamics, reduce bandwidth, and can reduce the capability of an outer-loop controller to affect actuator performance by creating state dependencies through feedback control.

Alternatively, the model can be linearized about an operating point using classical methods such as a Taylor Series approximation. At first glance, this may seem daunting for a pneumatic system, which appears to have prominent discontinuities due to the choked/unchoked flow transition. However, a closer look will demonstrate that mass flow, while non-smooth, is a continuous function of input and pressure, and can be quite nicely characterized using third- to fifth-order polynomial representations, as well as local linearizations

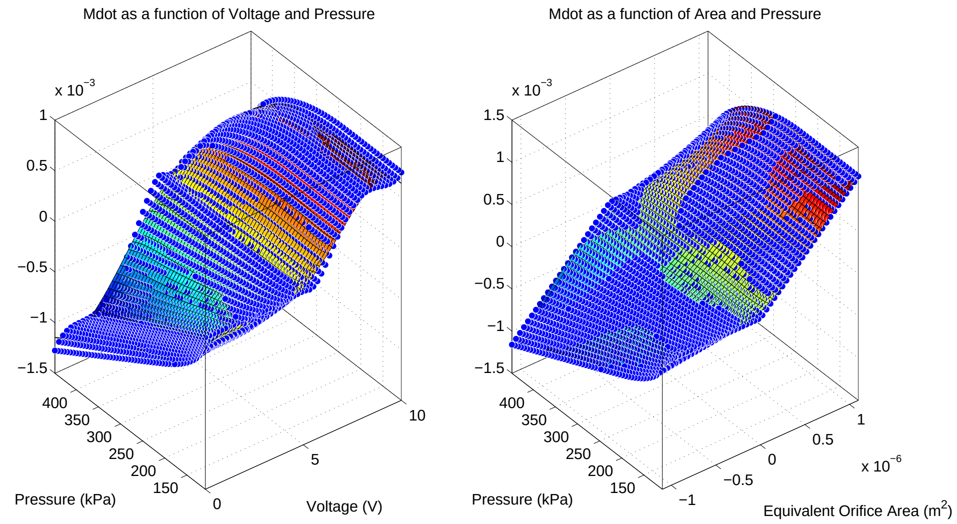

Figure 19: 5th order polynomial fit to mass flow data.

thereof (2nd order polynomials), as shown in Figure 19.

As defined in equation (1), mass flow is a function of u, P_u, and P_d, and while the assignments of these pressures (to supply, exhaust, and chamber pressure) may vary, it can be equivalently stated that $\dot{m}$ is a function of $P_{chamber} = P_{(a,b)}$, P_S, and P_E, where a and b refer to each of the cylinder chambers. If supply pressure, P_S and exhaust pressure, P_E are presumed constant, then $\dot{m}$ can be defined as a function $\dot{m}_{(a,b)} = f(u, P_{(a,b)})$.

The figure shows mass flow parametrized two ways: on the left, in terms of chamber pressure ($P_{(a,b)}$) and input voltage, u, and on the right, in terms of chamber pressure and equivalent orifice area $A_{eq}(u) = g(u)$. The difference further supports the assertion stated in section 3.4.1 – that the mapping from input to orifice area makes the model much more workable. In fact, the surface is actually quite smooth; the only non-differentiable points occur in a line near the offset voltage (when $A_{eq} = 0$). Further, change in slope is gradual. Using the parametrization on the right, a fifth order polynomial is used to approximate mass flow.

$$
\begin{aligned}
\dot{m}(A,P) = c_{00} & \\
& + c_{10}A + c_{01}P \\
& + c_{20}A^2 + c_{11}AP + c_{02}P^2 \\
& + c_{30}A^3 + c_{21}A^2P + c_{12}uP^2 + c_{03}P^3 \\
& + c_{40}A^4 + c_{31}A^3P + c_{22}A^2P^2 + c_{13}AP^3 + c_{04}P^4 \\
& + c_{41}A^4P + c_{32}A^3P^2 + c_{23}A^2P^3 + c_{14}AP^4 + c_{05}P^5
\end{aligned} \tag{19}
$$

It is easy to obtain symbolic linearizations of the above polynomial representation that are accurate at any operating point. Further, the simulation can be used to verify that both the polynomial and linear representations are good approximations over the prediction horizon, as seen in Figure 20.

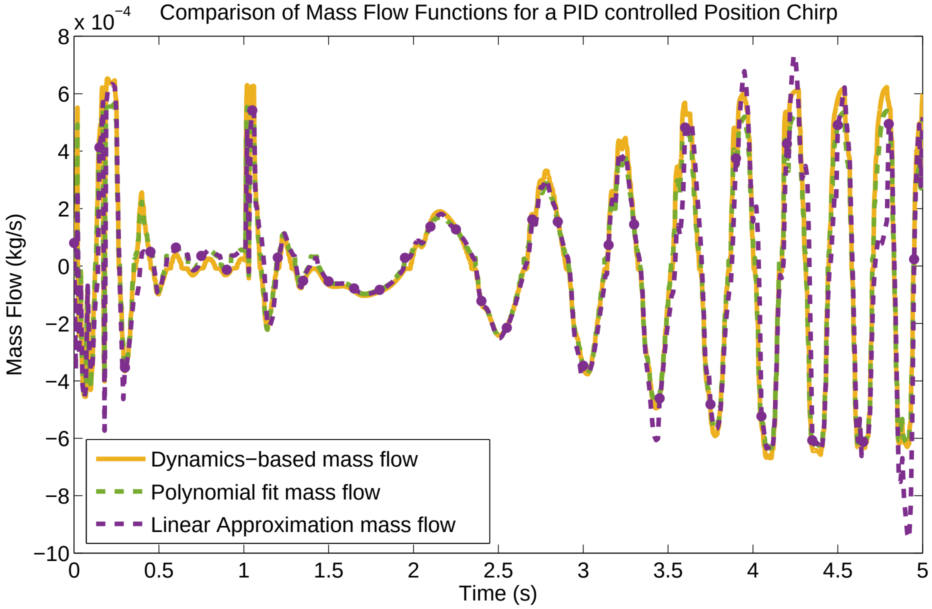

Figure 20: Comparison of mass flow functions.

The figure shows the mass flow for a simulated PID-controlled system following a position chirp signal. The yellow line depicts the mass flow, based on dynamics introduced earlier, while the green dashed line shows the mass flow as defined by a polynomial fit of equivalent orifice area and chamber pressure. Further, the dashed purple line illustrates the behavior of the linearized mass flow, re-initialized every 15 time steps (0.15 s for the 100 Hz controller) from the polynomial model. The time horizon is chosen to demonstrate that the linear model will be sufficient for use in a model predictive controller with $N_P = 15$. It is evident that the polynomial model is an accurate approximation for the exact dynamics-based approach, and that the linearized model is well-defined over the prediction horizon, so the models will be well suited for all the desired controllers.

Given these mass flow dynamics and the equations provided in section 3.1, the system can be compactly written as

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ P_a \\ P_b \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{P_a A_a - P_b A_b}{m} \\ \left(P_a A_a x_2 + \dot{m}(A(u), P_a) RT\right) \frac{1}{x_1 A_a} \\ \left(P_b A_b x_2 + \dot{m}(A(u), P_b) RT\right) \frac{1}{(L - x_1) A_a} \end{bmatrix} \tag{20}$$

where mass flow is approximated by local linear functions $\dot{m}(A(u), P_{a,b}) = K_1 A(u) + K_2 P_{a,b}$. K_1 and K_2 are constants that are easily obtained from equation (19). Then Taylor Series methods can be used to obtain a linear state representation. In practice, non-dimensionalization and scaling may be required to ensure numerically robust results.

CHAPTER IV

PERFORMANCE ANALYSIS OF BENCHMARK CONTROLLERS

Several performance measurement tools were developed to properly understand the challenges of controlling pneumatic and similarly compliant systems. The tools were used to quantify and compare the performance of several benchmark controllers and the proposed novel model predictive control strategies.

4.1 Benchmark Tools

It is important to establish quantitative metrics to use to assess target qualities of comparable controllers. Using metrics for similar force-control-oriented actuation analyses as a guide [70, 80], benchmark tests were determined to analyze properties of tracking performance, safety & compliance, and disturbance response. Four tests were designed using the hardware and simulation tools described in Chapter 3: Tracking bandwidth, Tracking of a 1 Hz sinusoidal signal, impedance frequency response, and static disturbance & release. The first two tests measure tracking performance, while the latter tests measure safety & compliance, and disturbance response, respectively. In each case, several hardware tests were carried out, and simulation was used to run wide-ranging parameter variation studies. The following sections will define each benchmark test and detail its purpose, describe the hardware/simulation configuration used to obtain it, and discuss the metrics extracted from the test.

4.1.1 System Bandwidth

The closed loop tracking bandwidth is obtained from the frequency response function of the reference tracking transfer function

$$F_{BW}(s) = \frac{Y(s)}{R(s)} \tag{21}$$

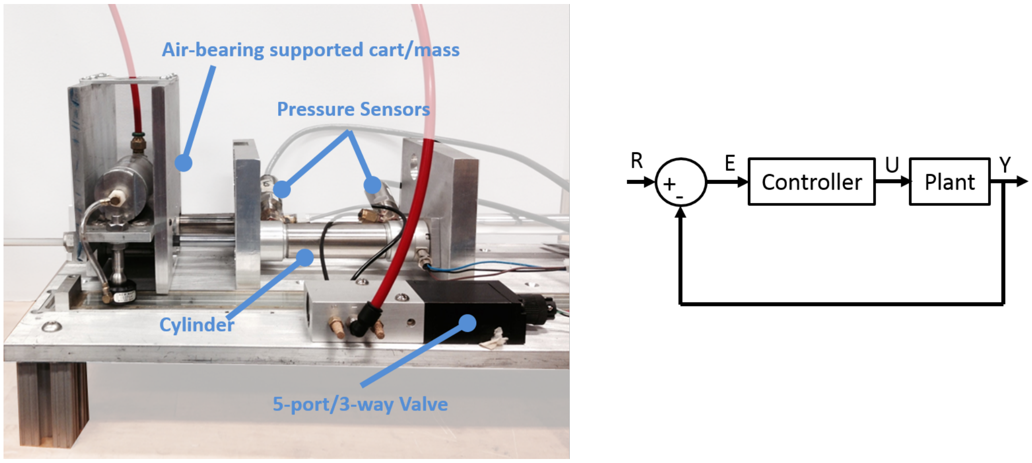

Figure 21: Position tracking hardware configuration (left) & block diagram (right).

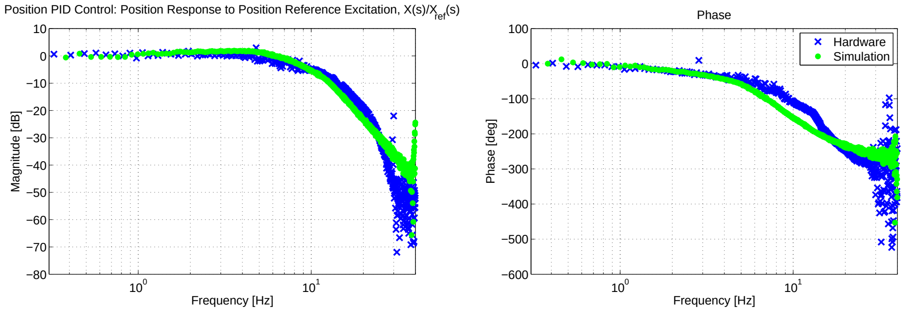

Figure 22: Example bandwidth plot result: Bode plot of hardware (blue) and simulation (green) Y/R transfer function for a position-PID controlled pneumatic cylinder.

where Y is the system output, and R is the desired reference trajectory. To obtain this transfer function, several sine sweeps were provided as a reference to the single-DoF hardware system, seen in Figure 21, and a spectral analysis, performed in MATLAB (detailed in the Appendix in sections A.1 and A.2), is used to obtain the transfer function of the averaged resultant data. An example output is shown in Figure 22, which compares hardware and simulation tracking of a position PID controller. The bandwidth is defined as the frequency at which the magnitude of the transfer function moves outside of the [-3 dB, 3 dB] range. The process was automated by applying MATLAB's `find()` function on the vector of magnitude values.

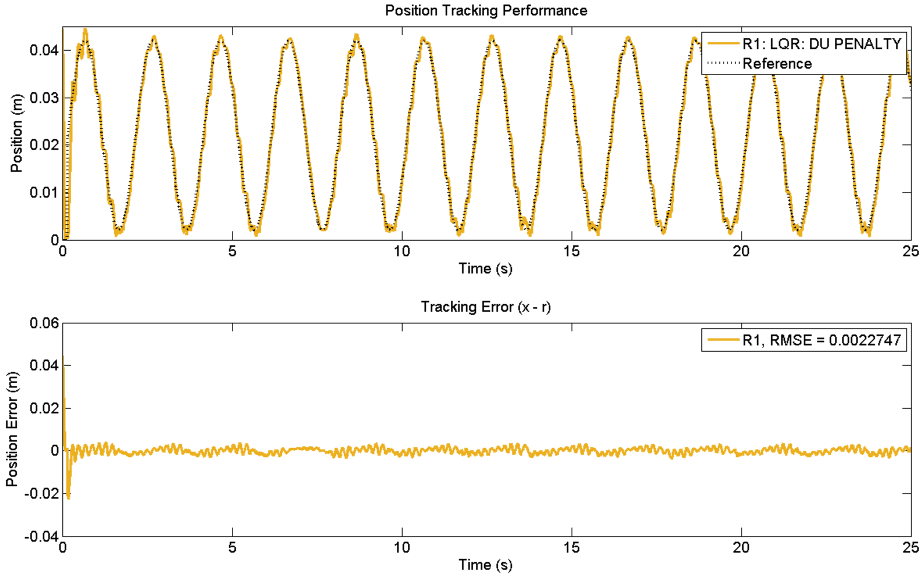

Figure 23: Example 1 Hz sinusoidal reference tracking plot. The LQR controller is used to track a sine wave, and RMSE is calculated from t = 12.5 to 25 s.

4.1.2 Tracking a 1 Hz Sinusoidal Reference

While the bandwidth offers a convenient measure of the range of signals that can be tracked, it is not a useful judge of accuracy. Though it is possible to examine the root mean squared error (RMSE) of sine sweeps, such results are inherently dependent on the bandwidth. Instead, the system illustrated in Figure 22 was used to track a 1 Hz sine wave – a slow, continuous, and changing signal that should be well within the bandwidth of any well-suited pneumatic controller, as defined by the numeric targets set in section 2.2.4. The RMSE of the tracked signal was measured and stored for each run, and used as an accuracy comparison. The aim of this metric is to provide comparable accuracy metrics for tracking of a changing reference; therefore, it is desirable to ignore any error caused by initial conditions and startup. Accordingly, only the second half of the time series data is used to calculate the RMSE. An annotated example output is shown in Figure 23. In addition to the RMSE of the 1 Hz sine, the RMSE of the different frequency sweep runs are also recorded, but data from sine sweeps that exceed the measured bandwidth is generally less reliable as a general tracking accuracy measure. Instead, the two tests provide independent measures of complementary performance attributes.

It is worth noting that RMSE alone is not always a great indicator of tracking performance. As the controller tuning parameters approach an unstable region, tracking can become increasingly oscillatory without significantly affecting RMSE. Similarly, an oscillatory system can have a similar or lower position RMSE than a system that simply has a small phase delay. To get rid of these problem cases, a metric to assess the velocity error, termed *normalized velocity error*, was used. The metric, abbreviated as nRMSvE is the root mean squared velocity error, normalized by a standard factor proportional to the maximum reference velocity. For a position measurement signal x and reference r with N samples,

$$nRMSvE = \frac{1}{F_{norm}} \left(\frac{1}{N} \sum_{i=1}^{N} (\dot{x}_i - \dot{r}_i)^2 \right)^{1/2} \tag{22}$$

The normalizing factor was chosen by looking at the maximum reference velocity, which is found by differentiating a reference position signal with amplitude A_{ref} and frequency f_{ref}, $r = A_{ref} \sin(f_{ref} t)$:

$$\dot{r}_{max} = max \left(\frac{d}{dt} (A_{ref} \sin(f_{ref} t)) \right) \tag{23}$$

$$= max \left(A_{ref} f_{ref} \cos(f_{ref} t) \right) = A_{ref} f_{ref} \tag{24}$$

For the 1 Hz sine tests, the frequency is constant, $f_{ref} = 2\pi(1)$, so the only the reference amplitude, A_{ref} affects the maximum reference velocity, and it is chosen as the normalizing factor in equation (22), $F_{norm} = A_{ref}$.

4.1.3 Output Impedance

The safety, force properties, or compliance of a system can be generally understood by examining its output impedance. The output impedance is characterized by the transfer function

$$Z(s) = \frac{F_l(s)}{X_l(s)} \tag{25}$$

where F_l is the load force due to external environmental effects (i.e., external forces acting on the load, not necessarily equivalent to the actuator input force), and X_l is the load position, which refers to the load displacement. An ideal force source has zero output impedance

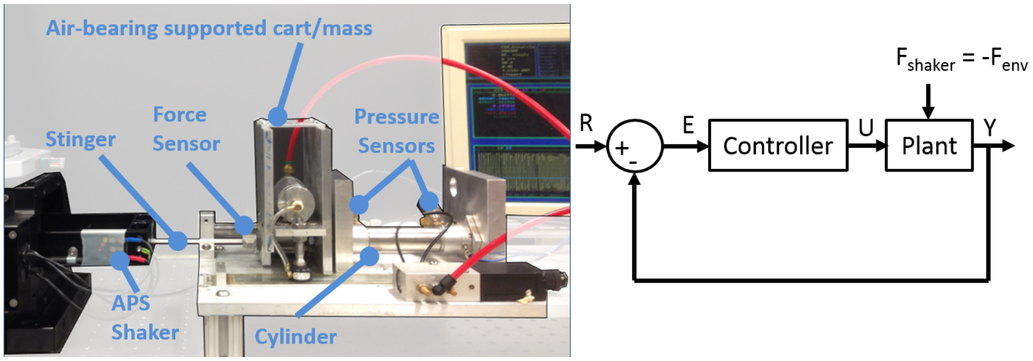

Figure 24: Position disturbance hardware configuration (left) & block diagram (right).

(i.e., the applied force is independent of any applied perturbation) and provides perfect force tracking; generally, mechanical actuators can be described as mass-spring-damper systems. The displacement is taken with respect to some equilibrium point, which, in a closed loop system, is likely the reference. For the block diagram shown in Figure 24, the load force is the reaction force incurred by the shaker, and the load displacement is the error from the position controller setpoint caused by shaker perturbation, $X_l = R - Y$, when the shaker is active.

While it is common to define mechanical output impedance in terms of velocity, $Z(s) = F_l(s)/V_l(s) = F_l(s)/sX_l(s)$, the distinction between position- and velocity-based impedance is merely that of an added integrator, and for pneumatic systems, it is more convenient to define position as the input, as defined in equation (25).

Among papers that focus on impedance properties, it is common to examine the closed loop impedance transfer function of a system [13,70,80], which can be obtained analytically or experimentally and visualized using frequency domain methods. For linear systems with well-defined controller and plant models, obtaining the impedance transfer function is simply a matter of writing out the closed loop system transfer function and rearranging the terms to related load force and load position. The resultant transfer function can then be visualized by drawing a corresponding Bode Plot.

For nonlinear systems and controllers, a comprehensive frequency-domain form is more difficult to acquire. The primary issue is that nonlinear systems can be input-dependent, and therefore an experimentally acquired Bode plot will only be a fair comparison within some input amplitude range. Additionally, as Buerger noted in [13], the magnitude plots are much more informative than phase plots for this application and suffice to perform impedance assessments. Therefore, it is necessary to compare numerous magnitude responses and compare them. This process was achieved by fitting the responses to an expected low-frequency transfer function and then comparing individual parameters.

The impedance transfer function was obtained using the system shown in Figure 24. The test may be run two ways – using either position or force as the input:

1. **Position error as the input:** In the first case, the controlled pneumatic actuator is first given a position setpoint. Then a position chirp is provided to the PD-controlled shaker, so that it perturbs the mass from its initial setpoint value along a chirp disturbance, and the corresponding environmental interaction force is obtained from force sensor attached to the cart on the piston, $F_{env} = F_{shaker} = F_l$. The cylinder's internal position sensor measures the corresponding position, X, which can in turn be used to calculate load displacement, $X_l = E = X - R$. Then the impedance is obtained from the relationship of measured interaction force to error from the pneumatic actuator's desired setpoint.

2. **Force as the input:** Once again, the position controlled pneumatic actuator is first given a position setpoint. Next, a voltage chirp is converted to a current by an amplifier, which produces a force chirp on the shaker. The impedance is obtained from the relationship of the measured interaction force (which should be a chirp with a slowly varying DC value) and the error caused by the disturbance from the actuator setpoint.

The position input case is more systematic and easily enforced in simulation, while the force-input method is a fairly standard approach to approximating interaction behavior of a nonlinear system with linear impedance transfer functions. However, the first version is difficult to implement over a broad frequency range in hardware[1], while the second version

[1] This problem requires accurate PD control of a shaker, subject to disturbance by a pneumatic actuator,

is hard to represent consistently and systematically in simulation (it requires some hand-tuning and several manual inputs). Therefore, simulation was predominantly conducted using the first method, and hardware tests were conducted using force-input. A comparison of the methods showed that they provided similar results, and that the hardware results were verified in simulation by both types of experiment. A detailed overview of this problem, the testing methods, and comparative results with the different tests is provided in the appendix in section A.4.2.

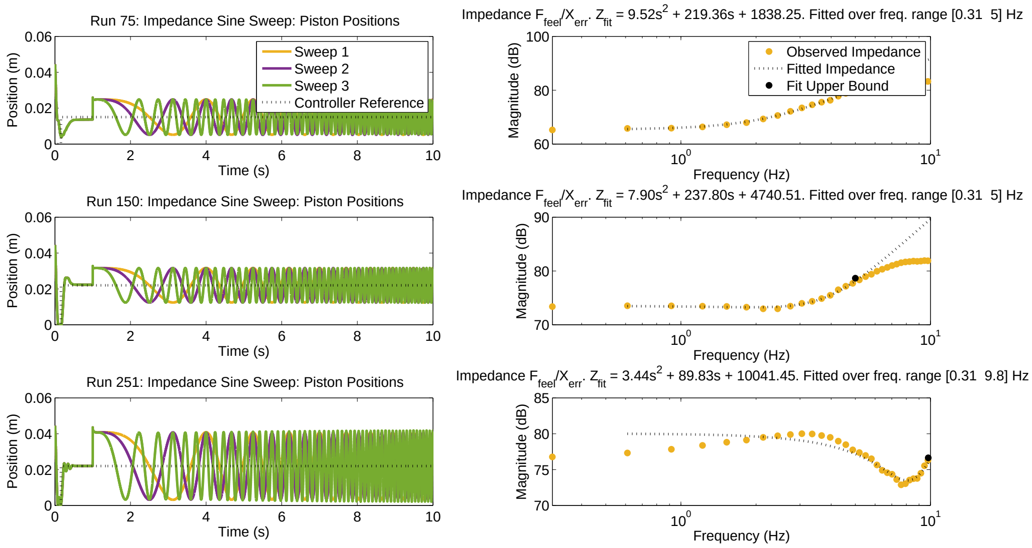

Figure 25: Example impedance fits (system under impedance control). Left: position data during three sine sweep trials. Right: corresponding impedance transfer function and automated fit.

Unlike the metrics posed in sections 4.1.1 and 4.1.2, there is no single quantitative value that results from the impedance transfer function. However, most output impedances take the form of a mass-spring-damper within the target frequency range (under 10 Hz):

$$Z(s) = \frac{F_l(s)}{X_l(s)} = (K_{fit} + B_{fit}s + M_{fit}s^2) \tag{26}$$

which means that it is possible to extract parameter fits for stiffness, damping, and inertial

with a bandwidth up to 15 Hz, which is straightforward in simulation using an ideal force source with infinite input capability, but challenging on current-limited hardware.

terms (K_{fit}, B_{fit}, and M_{fit}, respectively). This was achieved in MATLAB using a least-squares best fit approach over a range from 0 to 5-10 Hz, using the algorithm provided in section A.2. The frequency upper bound was determined by best fit to a magnitude matching equation (26), as judged by root mean square error.

Some sample outputs are seen in Figure 25. For most trials (over numerous controllers), this fit is quite good (within the 5-10 Hz range), especially for stiffness and damping matching. Periodically, however, fits are observed that do not match the mass-spring-damper model very well, which can cause misleading fits to be recorded. An example is seen in the bottom plot of Figure 25, which fits a much higher stiffness value at the low frequency than the controller actually exhibits. These cases are largely outliers[2], but are nonetheless a caveat worth considering. Additionally, in the sliding mode case, the observed transfer function moves considerably in the low frequency range, as shown in Figure 26, which often leads to low stiffness estimates. For these reasons, it's desirable to have a secondary measure by which to check impedance properties; the disturbance/release test, discussed in section 4.1.4 does so by examining impact force, rather than overall impedance.

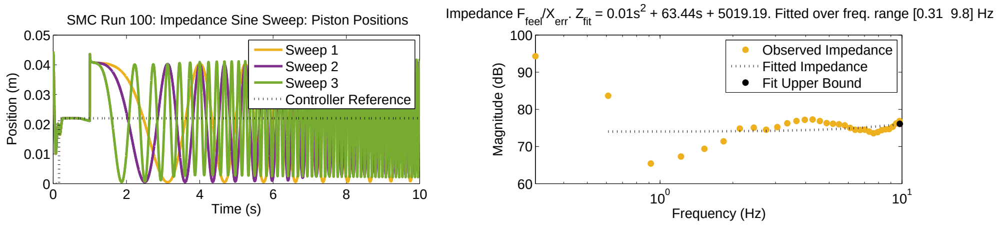

Figure 26: Example impedance fits for the system subject to sliding mode control. Left: position data during three sine sweep trials. Right: corresponding impedance transfer function and automated fit.

4.1.4 Steady Disturbance & Release

In past research efforts focusing on force or impedance qualities, researchers have often bypassed the issues resulting from frequency-domain tests by using time-domain tests that relate impedance to actual outputs: force, or perturbation. For example, Buerger [13]

[2] This statement can confidently be made by observing hundreds of available datasets

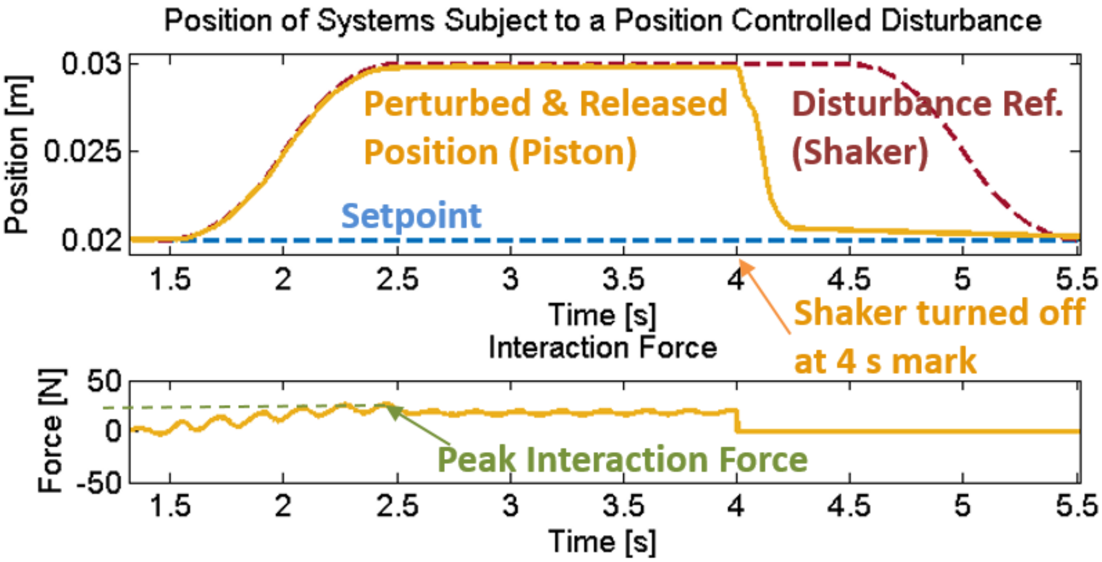

Figure 27: Example disturbance release simulation). Top plot: set point (blue), disturbance reference (red), and perturbed/released position (orange). Bottom plot: interaction force (due to shaker).

featured a test that used a commanded set point, and then provided low velocity disturbances, measured corresponding reaction forces, and used the force-perturbation distance relationship to estimate a stiffness. Richardson et al. used a similar approach to study the effectiveness of an admittance controller [67]. They provided sinusoidal force disturbances at a number of amplitudes and compared the observed disturbance to the expected disturbance with the target impedance. The equivalent approach with impedance control would be to administer sinusoidal position disturbances of varying amplitude and frequency, and plot observed load force against expected load force.

An example application of this concept is shown in Figure 27. The position behavior is shown in the top plot, while the bottom plot shows the axial interaction force. In the test, the controller was initially commanded to move the actuator piston to a set point, shown in blue. However, after a standard initialization period, the PD-controlled shaker perturbed the piston/mass to instead follow a disturbance reference. The shaker controller was designed using classical methods and a model obtained via a system identification of the shaker and attached unpressurized cylinder/cart, as detailed in the appendix in section A.4.3. Following completion of the test, automated scripts were run to define the results in terms of individual, comparable parameters, such as the average interaction force, as well as the response to unexpected interaction: post-release RMS error, overshoot, settling time, and steady state error.

A continuous disturbance was used for two reasons: (1) in simulation, collisions are not modeled, so a square acceleration profile is used, and (2) the force sensor used in hardware is a piezo-based sensor with built-in filtering, meaning it has a significant time constant – i.e., when subjected to impulsive loads, the sensor's transient behavior is not representative of the true interaction forces.

4.1.5 Other Methods for Measuring Stiffness

In a mechanical system, stiffness may be easily approximated by the presence of elastic elements. By contrast, a pneumatic system's stiffness is characterized by the pressures in either chamber.

Following the derivation in & Goldfarb [76], $K = -\frac{\partial F_{net}}{\partial X_1}$ can be found by differentiating the force balance and assuming that $\frac{\partial F_{Friction}}{\partial X_1} = 0$. For a cylinder with chambers a and b, denoted by corresponding subscripts,

$$\frac{\partial F_{net}}{\partial X_1} = -A_a \frac{\partial P_a}{\partial X_1} + A_b \frac{\partial P_b}{\partial X_1} \tag{27}$$

$P_{(a,b)}$ can be redefined as a function of x using the ideal gas law,

$$P_{(a,b)} = \frac{m_{(a,b)} RT}{A_{(a,b)}(L/2 \pm x)} \tag{28}$$

where L is the stroke length, x is the position measured from center of the stroke, and m_a and m_b are mass of the air in chambers a and b of the cylinder. By expanding the differentiation, it is possible to solve for stiffness as a function of mass and position, both measurable simulation states:

$$K = RT \left(\frac{m_a}{\left(\frac{L}{2} + x\right)^2} + \frac{m_b}{\left(\frac{L}{2} - x\right)^2} \right) \tag{29}$$

In practice, however, mass is not easily measured. Instead, a second implementation of the ideal gas law enables the stiffness to be rewritten in terms of pressure and position, which can both be measured using sensors on the testbed:

$$K = \frac{P_a A_a}{\left(\frac{L}{2} + x\right)} + \frac{P_b A_b}{\left(\frac{L}{2} - x\right)} \tag{30}$$

4.2 Baseline & Benchmark Controllers

Four position controllers were used as benchmarks: position-integral-derivative (PID) control, linear quadratic regulation (and its tracking extension, LQR/LQT), impedance control, and Sliding Mode Control (SMC). Since the desired performance is a trade-off between precise positioning, compliant actuator dynamics, and ease of implementation, the benchmark controllers were selected to span a broad spectrum of this range. PID control is easy to design, non-model-based, and should span a range of operation. LQR is similarly straightforward to implement and offers a baseline comparison for model predictive control: whereas the proposed MPCs use constrained optimal methods to achieve compliant, accurate tracking, LQR offers a linear and computationally lightweight benchmark comparison, and also provides an assessment of the necessity and benefits of constraints. Impedance control prioritizes a certain force interaction profile and should therefore supply the lowest achievable compliance profile, while Sliding Mode Control (SMC) uses robust control methods to minimize error, without regard for the effects on compliance.

4.2.1 PID Control

The PID controller is designed to have two loops: one is placed on net force in the cylinder, obtained from measurements of the pressure in each cylinder chamber, and the other is placed on position, found using a linear potentiometer in the cylinder. A block diagram is shown in Figure 28. Gains were designed first for a force control loop based on the nonlinear dynamics discussed in Chapter 3 and valve dynamics. Once a stable, responsive force control loop was achieved, it was modeled with first or second order closed loop dynamics (since the spool was shown to have first order dynamics with a 45 Hz cutoff frequency, a 2nd order approximation was possible as well), and used to design a PID controller that ensured stable, responsive system (rise times to a step input within 0.1 s and zero steady-state error), good tracking of continuous reference signals, and minimal valve chatter.

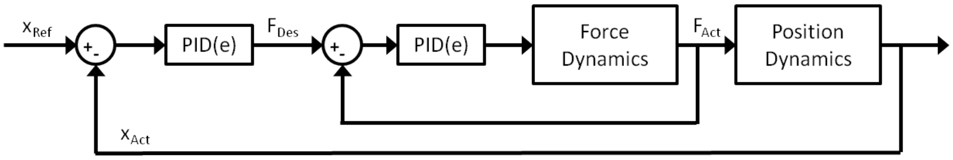

Figure 28: PID control block diagram.

4.2.2 Force Control Variations

Several of the benchmark controllers require a low-level force controller. In general, a PID controller was shown to suffice, but feedback-linearization, a model-based approach was also explored for comparison.

4.2.2.1 Input-Output Linearization

Input-output linearization can be used to replace the nonlinear force dynamics with a linear behavior by choosing an input that cancels out the undesired dynamics. The basic principle is to replace some of the output dynamics with some desired (linear) dynamics. In this case, the nonlinear cylinder internal force dynamics, $\dot{F}_{Press} = \dot{P}_a(x, \dot{x}, P_a, A_{eq})A_a - \dot{P}_b(x, \dot{x}, P_b, A_{eq})A_b$, are replaced by some desired force tracking dynamics, w. An appropriate input is then found by taking derivatives until the control input appears, at which point it can be defined algebraically in terms of modeled and desired behavior.

The desired dynamics, w, were set such that the force error dynamics have first order convergence,

$$\dot{e} = -\beta_0 e \quad \text{where} \quad e = F - F_d \tag{31}$$

Then each individual actuator is described by:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= F/m \\ \dot{F} &= w = \dot{F}_d - \beta_0 \left(F - F_d\right) \end{aligned} \tag{32}$$

where x_1, x_2, and F are the position, velocity, and force due to differential pressures, respectively.

The input appears after one derivative of the force dynamics: $\dot{P}$ is replaced by an input-dependent term from the pressure dynamics, such that:

$$\dot{F}_{Press} = \left(\frac{RT}{V_a(x)} A_{eq,a}(u)(c_d)_a \Psi_a(P_d, P_u) - \frac{P_a}{V_a(x)} \dot{V}_a\left(\dot{x}\right) \right) A_a \tag{33}$$

$$- \left(\frac{RT}{V_b(x)} A_{eq,b}(u)(c_d)_b \Psi_b(P_d, P_u) - \frac{P_b}{V_b(x)} \dot{V}_b\left(\dot{x}\right) \right) A_b \tag{34}$$

$$= \dot{F}_{Des} - \beta_0 \left(F_{Des} - F_{Press} \right) = w \tag{35}$$

where w represents the desired force dynamics in eqn. (35). An overlapped valve configuration is assumed for control, so that flow direction is determined by spool position and is opposite in each of the two chambers (since each chamber is always connected exclusively to the supply or exhaust). Then the orifice area, $A_{eq}(u)$, is signed according to flow direction, with $A_{eq,a}(u) = -A_{eq,b}(u)$. Setting $A_{eq}(u) = A_{eq,a}(u)$,

$$A_{eq}(u) = \frac{w + \frac{P_a}{V_a(x)} \dot{V}_a\left(\dot{x}\right) A_a + \frac{P_b}{V_b(x)} \dot{V}_b\left(\dot{x}\right) A_b}{\frac{RT}{V_a(x)} A_a (c_d)_a \Psi_a(P_d, P_u) + \frac{RT}{V_b(x)} A_b (c_d)_b \Psi_b(P_d, P_u)} \tag{36}$$

The values of c_d and Ψ in the denominator of equation (36) will vary depending on flow direction. However, since the terms in the denominator are all positive, flow direction of $A_{eq}(u)$ is determined by the numerator, which must be calculated first. The desired orifice area, $A_{eq}(u)$, is converted to a voltage input u using the known, one-to-one mapping A_{eq}. This can be achieved using geometry-based equations or spline approximations, as discussed in section 3.2, or with a simple linear approximation. In practice, the spline and geometry fits were shown to be effectively the same, and a zero-finding method was used to quickly map singed orifice area to voltage.

4.2.3 Linear Quadratic Tracking

In theory, an unconstrained MPC operating on a linear system should be equivalent to an LQR controller that is truly optimal for the given cost function. Accordingly, an LQR controller provides a baseline of achievable optimality. This system is of course not linear, but the LQR controller still offers a glimpse at what type of response might be expected for a given cost function and set of weights. In general, the LQR problem is defined as follows:

given a dynamical system

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ x(T_0) &= x_0 \end{aligned} \tag{37}$$

find the optimal control input $u^*(t)$ that minimizes a cost function $J(T_0, T_f, x_0, u(t))$. Since the controller used in this thesis was implemented as a discrete controller, the LQR realization varied slightly from the continuous representation. Instead, the problem is to find a control input u_k for the system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \\ x(T_0) &= x_0 \end{aligned} \tag{38}$$

to minimize the quadratic cost function:

$$J_k = \frac{1}{2}(Cx_N - r_N)^T P(Cx_N - r_N) + \frac{1}{2}\sum_{k=1}^{N-1}(Cx_k - r_k)^T Q(Cx_k - r_k) + u_k^T Ru_k \tag{39}$$

A control input is found to satisfy

$$u_k = -K_k x_k \tag{40}$$

where K_k is the optimal gain, found using the discrete Riccatti Equation:

$$\S_k = A_k^T(S_{k+1} + S_{k+1}B_k(B_k^T S_{k+1}B_k + R_k^{-1})B_k^T S_{k+1})A_k + C^T Q_k C \tag{41}$$

where $S_N = P$ is given. Then the gain matrix is found via:

$$K_k = (B_k^T S_{k+1}B_k + R_k)^{-1}B_k^T S_{k+1}A_k \tag{42}$$

For general, finite-time LQ regulation, this control is executed by first solving equation (41) backwards in time for S_k. Variations exist for discrete, infinite horizon, and tracking cases.

Since this thesis uses a time-invariant model, it is sufficient to follow the infinite time horizon application. Further, we are interested in a tracking application, rather than reference. It is important to distinguish the tracking case, where the reference, $\tilde{y}(t)$ is known $\forall t \in [0, T_f]$ but has not been fitted to a model, from the servo case, where $\tilde{y}(t)$ has known dynamics that may be estimated. It can be shown that a feedforward term will solve the modified problem of tracking. A term containing a time-dependent variable v_k is added to the input

$$u_k = -K_k x_k + K_k^v v_k \tag{43}$$

where v_k is the solution to

$$v_k = (A - BK_k)^T v_{k+1} + C^T Q r_k \quad v_N = C^T P r_N \tag{44}$$

and r_k is the reference.

However, this application still requires that the reference be known over some prediction horizon. The most general case, used in this controller, is to simply reorganize system states into tracked states, X_T, and other states, X_N:

$$X = \begin{bmatrix} X_T \\ X_N \end{bmatrix} \quad \text{and} \quad u_k = -K\left(\begin{bmatrix} X_T \\ X_N \end{bmatrix} - \begin{bmatrix} r_k \\ 0 \end{bmatrix}\right) \tag{45}$$

It is further important to include integral compensation. Therefore, the system is augmented to included integral states in the model and cost function. A system with tracking and integral compensation would have the form:

$$\begin{bmatrix} X_T \\ X_N \\ X_I \end{bmatrix}_{k+1} = \begin{bmatrix} A_{TT} & A_{TN} & 0 \\ A_{NT} & A_{NN} & 0 \\ dt & 0 & I \end{bmatrix} \begin{bmatrix} X_T \\ X_N \\ X_I \end{bmatrix}_k + \begin{bmatrix} B_T \\ B_N \\ 0 \end{bmatrix} \left(-K \begin{bmatrix} X_T - r \\ X_N \\ X_I - \int r \end{bmatrix}_k \right) \tag{46}$$

The state penalty in the cost function, Q, is augmented to include a penalty on the integral state as well.

4.2.4 Impedance Control

Impedance control is a type of *indirect force control.* The idea is that a manipulator's impedance completely describes how it will interact with a variety of environments; in fact, ideally, the environment doesn't matter [13]. More broadly, if an arbitrary impedance can be achieved, then an arbitrary behavior can be achieved [43]. Further, unlike stiffness control, which affects stationary behavior, impedance control imposes an objective on *dynamic* behavior, specified by a mass-spring-damper system [92].

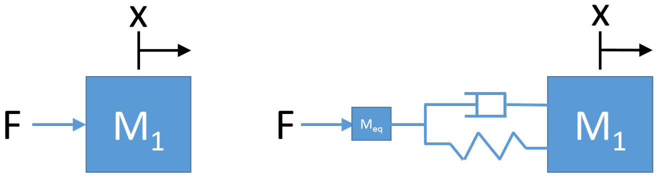

Figure 29: Pure force source (left) vs. system with inherent impedance (right).

Many impedance-based controllers are designed with the aim of lowering the apparent (closed-loop) impedance of the controlled system. Generally, the ideal goal is zero impedance, such that the system should behaves as a pure force source. However, most systems have some inherent impedance, resulting in a resistance to motion, as illustrated in Figure 29. Further, impedance control of systems is affected by a number of properties, including the inherent impedance in the actuator (generally related to its inertia, among other things), which can affect achievable apparent inertia. Achieving low impedance using high-force actuators can be particularly difficult, as noted by Stephen Buerger [13], in his dissertation on the topic. Therefore, it is common to set achievable target impedances using some combination of mass, spring, and damper elements:

$$Z_{des} = K_{des} + B_{des}s + M_{des}s^2 \tag{47}$$

As noted in section 4.1.3, position-based impedance will be used here for consistency.

There are two types of impedance control: *position-based impedance control*, also known as *admittance control*, and *force-based impedance control*. In position-based impedance control, an inner position loop is used and positions are specified based on a measured or observed exerted force & environment force, and in force-based impedance control, an inner force loop tracks a force commanded based on position tracking error.

Admittance control is used when the goal is to select a compliant position based on a measured force and a desired position. The advantage of this approach is that it leads to robust position control (robust to noise and non-environmental disturbances) that also exhibits the desired impedance properties. However, the causality of admittance control is reversed from impedance control, meaning that if the aim is backdriveability, the use of admittance control essentially opposes controller goals [13]. Therefore, the benchmark controller is a force-based impedance controller.

The basic impedance controller can be demonstrated on a system with dynamics

$$M(x)\ddot{x} + B(x, \dot{x}) + K(x) + F_n(x, \dot{x}) = F_a + F_e \tag{48}$$

where M, B, and K are inertial, damping, and stiffness terms, and F_n represents system friction. F_a is the desired actuator force and F_e is the force from interaction with the environment. Then F_a is chosen to obey some impedance relationship, e.g. $F_a = K(x_a - x)$, where x_a is a "virtual reference" – a suggested but not necessarily realizable trajectory that is the basis for the position error. The aim of the virtual reference is to influence the "feel" of the impedance control, i.e. which direction the user feels resistance from. Then the aim is for the controlled system to exhibit some target impedance behavior

$$F_e = sZ_{des}(s)(X_a - X) \tag{49}$$

where F_e is the force due to the environment, also called the load force, F_l, as in section 4.1.3. Impedance control in this simple format (no additional terms to compensate for inherent system dynamics) is commonly termed *simple impedance control*, and is effectively an extension of PD control. It has been used by several past researchers, such as [80] to control SEAs. However, it is clear from equation (48) that the inherent actuator dynamics, if significant, will preclude the controller from achieving the target performance:

To reduce the effects of the open-loop system dynamics, compensatory terms are added to the control law, as in [100]. This strategy leads to the control law defined by equation (50), which results in closed dynamics defined by equation (51).

$$F_d = (\tilde{M} - \hat{m})\ddot{x} + (\tilde{B} - \hat{b})\dot{x} + \hat{m}\ddot{x}_a + \hat{b}\dot{x}_a + \hat{k}(x_a - x) + \tilde{F}_f \tag{50}$$

$$\begin{aligned} F_e = {} & \hat{m}(\ddot{x} - \ddot{x}_a) + \hat{b}(\dot{x} - \dot{x}_a) + \hat{k}(x - x_a) \\ & - \Big((\tilde{M} - m)\ddot{x} + (\tilde{B} - b)\dot{x} + (\tilde{F}_f - F_f)\Big) \end{aligned} \tag{51}$$

$\tilde{M}$, $\tilde{B}$, and $\tilde{F}_f$ are the estimated mass, damping, and friction parameters. As long as these are reasonably close to the true values, F_e will exhibit the desired impedance characteristics.

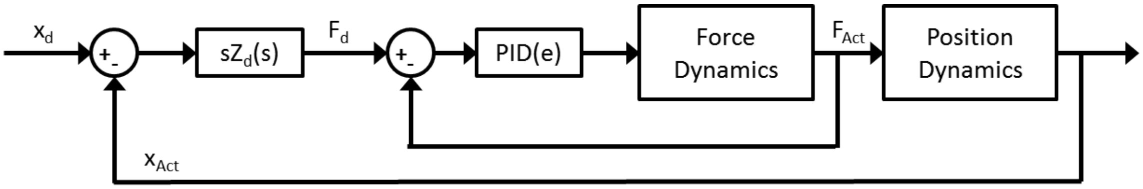

Figure 30: Simplified impedance control block diagram, adapted from [80].

It is worth nothing that impedance control may be improved through the addition of force feedback [43]. A useful form for incorporating force feedback distributes the accompanying force gain, K_f, over several impedance terms:

$$F_a = K(x_a - x) + B(\dot{x}_a - \dot{x}) + K_f\left[F_e + K(x_a - x) + B(\dot{x}_a - \dot{x})\right] \tag{52}$$

where x is actuator position and r is the virtual position (the end-effector equivalent of virtual reference). Substituting F_a into equation (48), it is evident that the force gain will serve to minimize effects caused by the initial system impedance:

$$\frac{m}{1+K_f}\ddot{x} + \frac{b}{1+K_f}\dot{x} + \frac{F_n(x,\dot{x})}{1+K_f} + K(x - x_a) + B(\dot{x} - \dot{x}_a) = F_e \tag{53}$$

However, the addition of force sensors for each degree-of-freedom, which is outside the scope of this thesis, which instead seeks to investigate the impact that improved control can make on a standard system, with position and pressure sensing. Presumably, most controllers could be improved via expanded sensing capabilities.

4.2.5 Sliding Mode Control

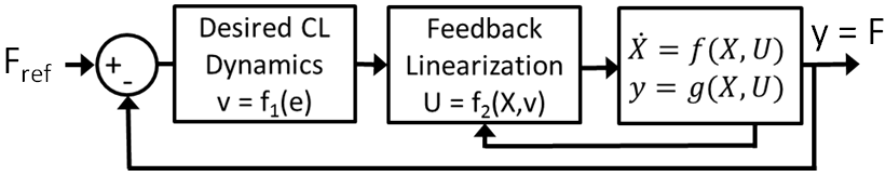
F_ref
+
-
Desired CL Dynamics v = f_1(e)
Feedback Linearization U = f_2(X,v)
Ẋ = f(X,U)
y = g(X,U)
y = F

Figure 31: Sliding mode control block diagram.

The goal of Sliding Mode Control is the relate the problem of tracking for n^{th} order systems to the much simpler challenge of stabilizing a 1^{st} order system, as described in [78].

The idea is to represent the error dynamics as a surface, $s = 0$, known as the *sliding surface*. s is defined as a function of the error state and time:

$$s = \left(\frac{d}{dt} + \lambda\right)^{n-1} e \tag{54}$$

where λ is a strictly positive constant and n is the number of times that the output must be differentiated to obtain the input term. Then a control law is devised such that the states stays on the sliding surface, i.e. $x - x_d = s = 0$, and $\dot{s} = 0$ once $s = 0$. This control term, an equivalent input u_{eq}, may be found by setting $s = 0$ and substituting equations of motion, differentiating until the input appears, and then solving for u.

For the pneumatic system, which relates orifice area to position, $n = 3$ (even though the spool has dynamics, they may be ignored, since the expected bandwidth is well under 45 Hz). Then $s = \ddot{e} + 2\lambda\dot{e} + \lambda^2 e$ where $e = x - x_d$. By setting $\ddot{e} = \ddot{x} - \ddot{x}_d$ and substituting for $\ddot{x}$, the sliding surface is defined in terms of system states:

$$s = \frac{1}{M}\left(P_a A_a - P_b A_b - P_{atm} A_r - B\dot{x}\right) - \ddot{x}_d + 2\lambda\dot{e} + \lambda^2 e \tag{55}$$

This equation is differentiated, and equations for $\dot{P}_{(a,b)}$ are substituted in so that the input appears in the equation. The system is solved for the signed equivalent input area, $u_{eq} = A_{eq} = c_d A$, and a control law u is defined:

$$u = A_{eq} - K_{sat}\left(\frac{s}{\phi}\right) \tag{56}$$

$$A_{eq} = \frac{M\left(x_d^{(3)} - 2\lambda\ddot{e} - \lambda^2 e\right) + B\ddot{x} + \left(\frac{P_a A_a^2}{V_a} + \frac{P_b A_b^2}{V_b}\right)\dot{x}}{RT\left(\frac{\Psi_a A_a}{V_a} + \frac{\Psi_b A_b}{V_b}\right)} \tag{57}$$

In this equation, K_{sat} is a gain that drives the error state back onto the sliding surface if it deviates, i.e. if $s \neq 0$. The use of the saturation term and ϕ variable serve to enforce a boundary layer: outside the range of ϕ, the control is saturated, and within it, the added term is effectively a proportional controller. This reduces the instance of chatter in the system. It can also be shown that λ affects error dynamics: the related quantity $1/\lambda$ is the system's time constant. Exact controller parameters vary, but a good fit was found using the parameters $K = 3e - 7$, $\Phi = 10$, and $\lambda = 50$.

As noted in sections 4.2.2.1, 3.2.1, and 3.4.3, the equivalent orifice area may be mapped to voltage input using either polynomial fits or geometric methods. Further, the sign can be determined from the numerator alone, since the denominator is always positive. Finally, in theory, the discharge coefficient varies depending on the chamber side, but a comparison of the approaches in both hardware and simulation showed that level of detail to have negligible performance benefits, and that an averaged discharge coefficient provides the same tracking results.

4.2.6 Filtering and Derivatives

The benchmark (and MPC) controllers use various derivatives of the references and position states – up to third order, in the sliding mode case. In simulation, it is sufficient to approximate these with two-step derivatives and – in the case of high order derivatives, some low pass filters, enforced via Butterworth filters.

In hardware, these filtering methods were largely insufficient. While they were acceptable for reference derivatives, they failed on state measurements, which of course are obtained from the hardware. These signal processing challenges are primarily due to error propagation: even though the position signal is generally very clean (the potentiometer uses a wiper built into the actuator piston), very small errors and discrete realization propagate and cause errors in the second and third order derivatives. Additionally, cylinder force is measured by pressure sensors, meaning that it only encompasses differential force (i.e., it does not include friction), and is notably noisier than position. Therefore, a Kalman Filter was used to obtain acceleration and jerk, using a linear model that approximated the static friction with high coulomb-viscous terms. Velocity was obtained by differentiating position, but acceleration was measured via the Kalman filter and a four-step averaging filter. These methods markedly improved performance of all the controllers requiring higher-order derivatives (all controllers other than PID).

4.3 Benchmark Results

To better define the state-of-the-art in compliant tracking of pneumatic systems, the controllers introduced in section 4.2 were applied to the single-DoF pneumatic actuator. Hardware tests were used to assess performance and establish trends, while parameter variation studies, conducted in simulation were used to more extensively verify performance trends and demonstrate controller limits and capabilities.

4.3.1 Controller Tuning

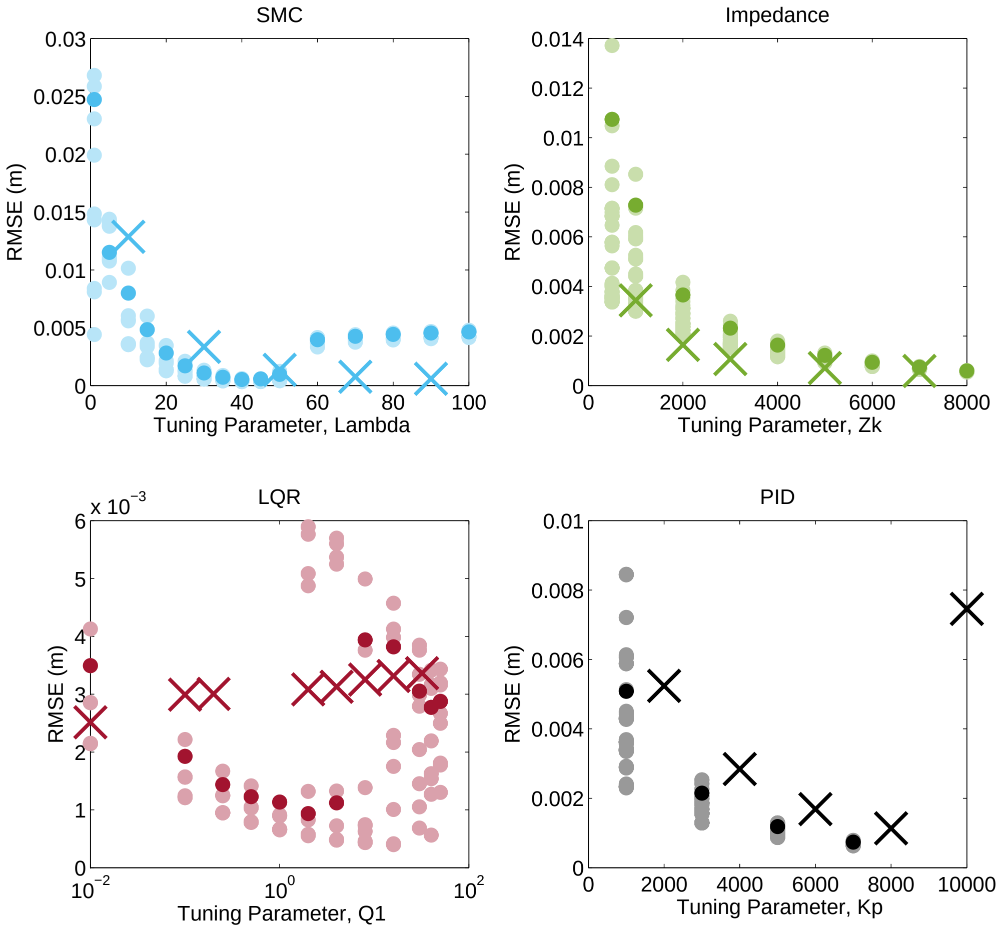

Figure 32: Benchmark controller fine-tuning. Clockwise, from top left: sliding mode control, impedance control, PID control, and linear quadratic tracking. Gray dots: all simulated points. Colored simulated dots: points meeting the tuning requirements, and used in later results. Hardware validation represented by X's.

To ensure that the hardware results provided a fair controller comparison, it was necessary to first ensure that an appropriate range of tuning parameters was used. The simulation was used to rapidly explore the space of controller parameters over several operating points, which included parameters such as reference bias and amplitude, and – in some cases – other controller tuning parameters. After several runs, good parameters and acceptable parameter ranges were identified, and dominant tuning parameters – selected based on theory and observation – were determined and varied to demonstrate performance impact. Additionally, hardware tests, indicated by 'Xs', were used to verify performance trends. The targeted parameter tuning ranges are illustrated in Figure 32, which shows how error is reduced as the dominant parameter, indicated on the x-axis, is increased. In the figure, the y-axis RMS error is taken from the 1 Hz sine reference. The pastel-colored dots show the range of all selected simulated parameter combinations, while the bold dots indicate parameter combinations that match the runs observed in hardware (with some slight differences on exact tuning parameter values in some cases, as evident in the figure).

4.3.2 Data Selection & Parameter Variation

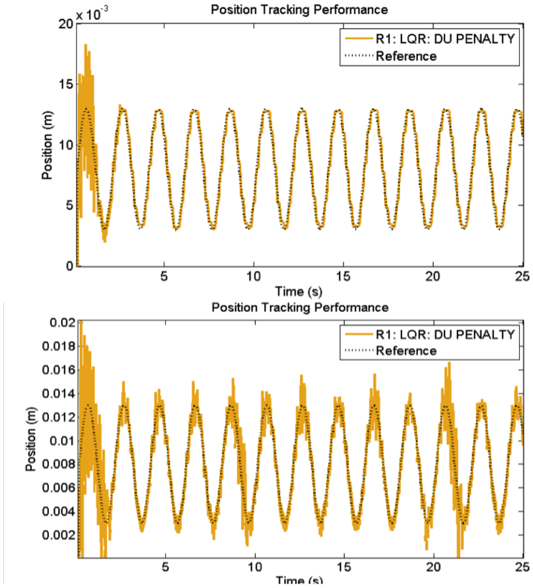

Figure 33: Usage of normalized RMS velocity error (nRMSVE) to get rid of undesirable cases. Top plot: nRMVE = 15.4. Bottom plot: nRMVE = 18.9.

In some cases, over-tuning the parameter can lead to chatter/oscillation, which is undesirable. While this ultimately leads to diverging error, it is important to also remove false positives – tracking in which the position oscillates about the reference, but the RMS error remains quite small. Normalized velocity error was used as a bound to remove these cases: by observation, a bound on root mean square normalized velocity error, $\langle(\dot{e}/\dot{r})^2\rangle^{1/2}$ was set at 15.5. An example application of this metric is shown in Figure 33; effectively, it removes marginally stable cases using a simple numerical test.

Further, both these tests and the later ones are tested on 10 operating points, defined by different bias and amplitude values of the respective sinusoid, chirp, or disturbance reference. These include small and large amplitudes and biases near the actuator center position and extremes, with respect to the stroke length.

The hardware tests were performed over less extensive ranges and operating points than the simulation, and were primarily in place to validate the simulation results. In general, trends observed in simulation were verified in hardware, though actual controllers slightly underperformed the simulated ones. In the SMC case, the error divergence seen in simulation didn't occur until approximately $\lambda = 300$, though audible valve chatter began right near $\lambda = 50$ – the point at which simulation error began to diverge again. These differences are likely due to differences in the friction model; harsher friction conditions in practice would lead to lower performance, but reduced impact of valve chatter on error divergence, as observed in the SMC case.

The benchmark controller performance was evaluated using the selected operating points and controller parameter ranges indicated by the respective colored dots in Figure 32, and assessed on the basis of the performance metrics introduced in section 4.1. A complete overview of the collected data and individual results is provided in section A.4. The relative performance of the controllers among different metrics was observed using a parametric plotting approach detailed in Section A.4.5.

4.3.3 Compliant Tracking with Benchmark Controllers

Figures 34 - 37 depict the performance of the benchmark controllers in the areas of compliance, accuracy, and safety. From the impedance fits, the stiffness measurement was shown to be parameter that most dominantly defined system behavior in the low-bandwidth regions; therefore, stiffness was selected as a choice parameter to evaluate compliance.

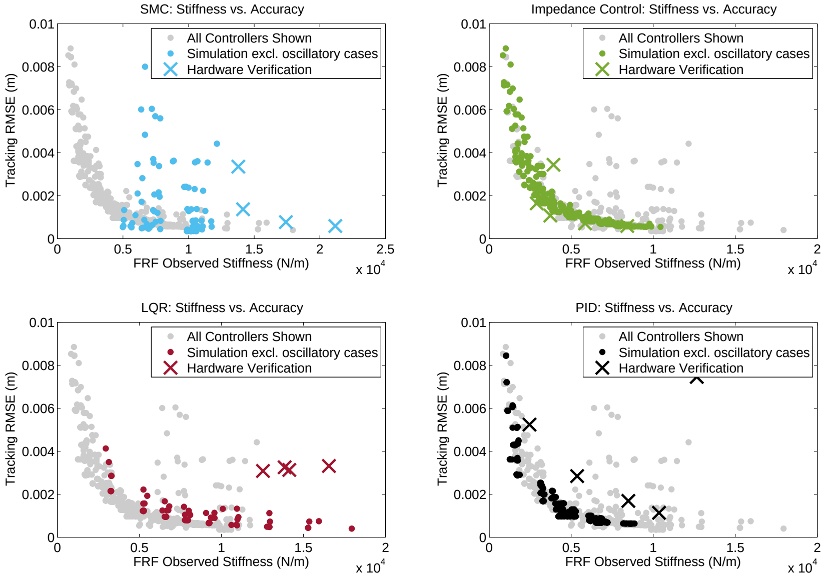

Figure 34: Accuracy vs. compliance of benchmark-controlled systems. Depicted by RMS error of 1 Hz sine tracking (x-axis) and observed stiffness parameter from the impedance frequency response fits (y-axis).

First, the evident trade-off of compliance and accuracy is illustrated in Figures 34 and 35. Accuracy is represented by RMSE from the 1 Hz sine tracking test, while compliance is illustrated by stiffness – obtained from and impedance frequency response fit in Figure 34, and using the stiffness from the force-perturbation ratio in Figure 35. As anticipated, sliding mode control is always stiff, never dropping below 5000 N/m, and with no clear trade-off, though the majority of results are accurate, falling below 4 mm RMS error. By contrast, impedance, PID, and LQR control all fall along a distribution that demonstrates a clear trade-off of accuracy and compliance, and appear to primarily be different means to the same end. Impedance control more clearly spans the range (likely due to the ease of

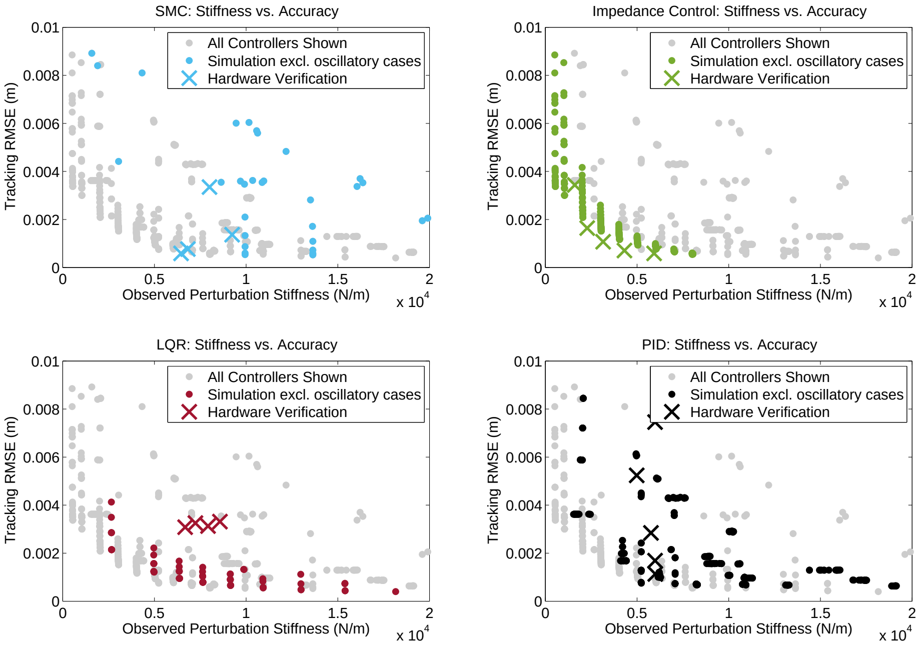

Figure 35: Accuracy vs. compliance of benchmark-controlled systems. Depicted by RMS error of 1 Hz sine tracking (x-axis) and observed stiffness parameter from the force-position error relationship in the perturbation test (y-axis).

targeting a desired performance), while LQR – the only full-state feedback used – is simply unable to achieve very low stiffness tracking with any reasonable normalized RMS velocity errors.

In hardware, it was impossible to recreate the low errors observed with LQR seen in simulation, but the trend that Q had only a minimal impact on performance and plateaued after a certain point – seen in Figure 32 – was clearly observed. The discrepancies between hardware and simulation are caused by un-modeled effects (e.g., the difference in friction representation between hardware and simulation), and suggested that a more aggressive controller is needed to bypass these effects. One asset of MPC is the ability to effectively solve an overly aggressive optimal control problem, but limit the input and input rates via constraints, potentially bypassing this issue. The LQR controller also demonstrated that a stiff closed-loop system was required to even approach desired target tracking performance. The behavior of the LQR controller is important: it implies that optimality alone is insufficient to create a compliant system.

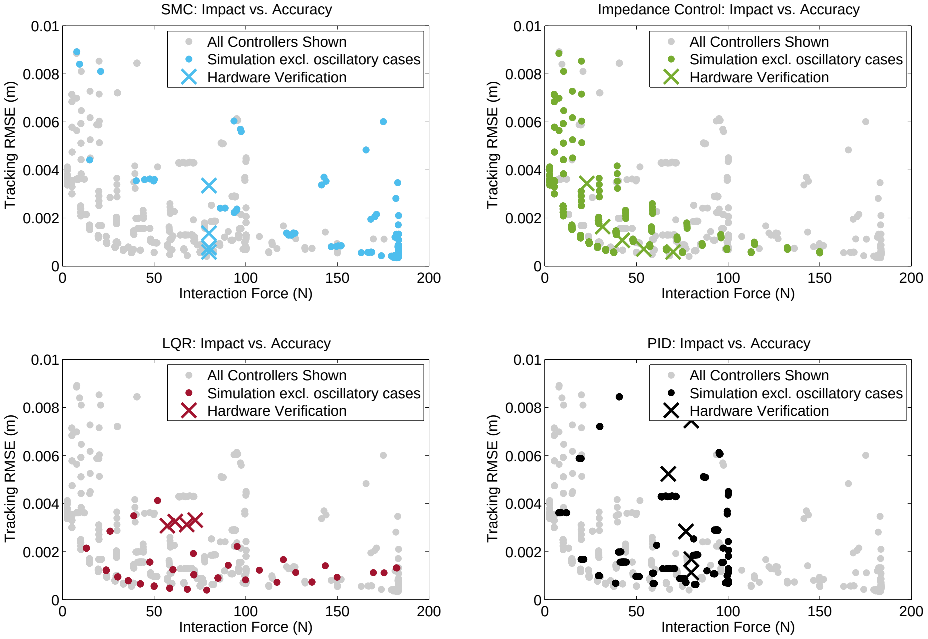

Figure 36: Force response to unexpected disturbance for benchmark controllers. Gray dots show performance of all benchmark controllers, while colored dots highlight performance of the selected controller. On the x-axis: RMS error from 1 Hz sine tracking. On the y-axis: Disturbance force to unexpected disturbance.

Similar behaviors are evident by simply examining the average disturbance force when these benchmark controlled systems are subjected to an unexpected disturbance. In Figure 36, it is seen that the most accurate systems produce the highest interaction force, on the order of 150-200 N. In the figure, sliding mode Control is shown to be subject to the same trade-off of accuracy and compliance observed in the other controller performance, though the trend is less well-defined. This test was impossible to verify in hardware: 4 of the 5 sliding mode controllers caused the shaker to exceed its maximum current allowance and thereby fail at the maximum allowable force, as seen in the figure. In general, sliding mode controllers result in a very stiff behavior, as verified in several tests throughout this thesis. Additionally, PID and impedance control have lower upper bounds, and LQR has a very narrow performance region, though the simulation produced larger values for different operating parameters.

Another desirable performance quality is a good return to tracking following a disturbance. Figure 37 quantifies the behavior of a tracked system after it has been subjected to a disturbance. The position-controlled actuator returns to the setpoint, eventually reaching a steady state error, but has some overshoot in doing so. The figure compares the overshoot (on the x-axis) to the steady-state RMS error (on the y-axis). Since the response to setpoint is from a positive disturbance, a negative overshoot value indicates an under-damped response (and an actual overshoot), while a positive value indicates an over-damped response. Here, the target behavior would be near zero. Unfortunately, this test could not truly be verified in hardware: to perfectly recreate it, the shaker would need to be completely disconnected from the system (e.g., by cutting the stinger). However, this would damage the shaker and is potentially dangerous, so instead, the shaker was merely turned off. Since the shaker is a sizable mass (more than twice that of the cart) and generally acts as a damped system, it effectively impedes the response; therefore, hardware results were overly conservative. Nonetheless, the simulation responses provide an overview of behavior varieties, and demonstrate that only the impedance controller can really approach the target desired performance response following a perturbation.

Finally, the tracking ability is inherently correlated with the system bandwidth. Figure 38 shows how the bandwidth varies as a function of stiffness gleaned from an impedance fit. While there is no clear relationship between stiffness and bandwidth for sliding mode controller, it is always quite stiff: the observed controller stiffness never falls below 5000 N/m. Similarly, LQR always falls in the 3-5 Hz bandwidth domain, but is never able to achieve stiffnesses below 3 kN/m. Impedance and PID control both display a trend of increasing bandwidth with increasing stiffness, culminating in a 4-10 Hz observed bandwidth, easily satisfying the target range of 2-5 Hz. However, among the impedance and PID controllers, there is a low-bandwidth, low-stiffness region that is very sparse compared to the remainder of the accessible range, indicated by the dashed orange circle in figure 38. This opening supports the hypothesis that there is some room for improvement in the compliance-accuracy trade-off in exchange for reduced bandwidth, but that it is effectively inaccessible using the benchmark control methods.

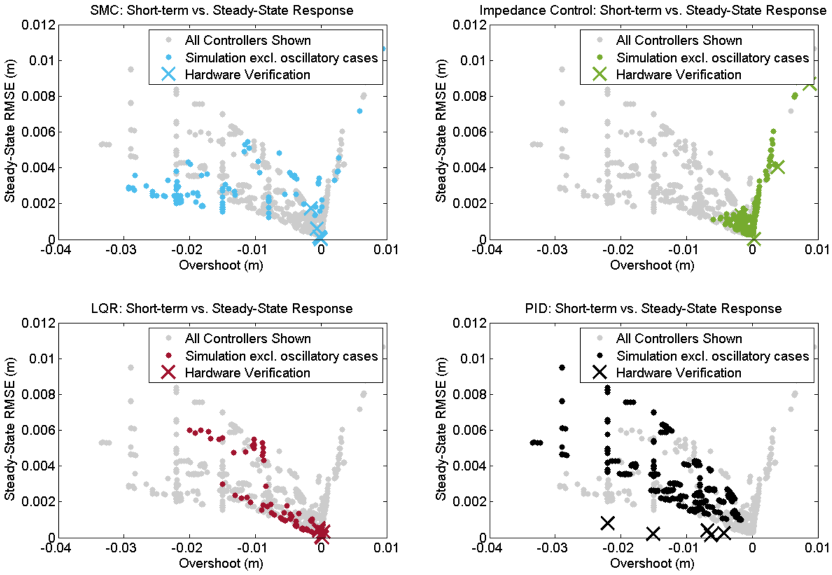

Figure 37: Response of benchmark controllers following an unexpected disturbance. Controlled systems are perturbed to a point above the setpoint, then released. The x-axis shows overshoot (positive values indicate an over-damped response, and negative values indicate under-damped response), and steady-state error following release is shown on the y-axis. Clockwise, from top left: sliding mode, impedance, PID, and LQR control.

4.3.4 Performance Gaps

The benchmark results suggest that there is room for improvement. It is clear from Figures 34 and 35 that there is a maximum stiffness above which the increase in accuracy is negligible. For compliant control, it would be desirable to limit the stiffness so that it reaches, at most, this maximum value, ideally without explicitly including the value in control. Furthermore, if a known stiffness limit is provided, the tracking should be comparable to the best result observed by the benchmark controllers at a similar stiffness, as observed in section 4.3.3. Finally, there are certainly some performance gaps that would be desirable to close. Specifically, it is difficult to impossible for existing controllers to reach stiffnesses below 5000 N/m while also maintaining tracking RMS error below 4 mm, or about 10% of the stroke. Additionally, for safety, a 50 N interaction force is generally a well-regarded upper bound: injury from static collisions often occurs in excess of 50 N [48]. This is also 33% of the maximum force for chest impact based on ISO standards for

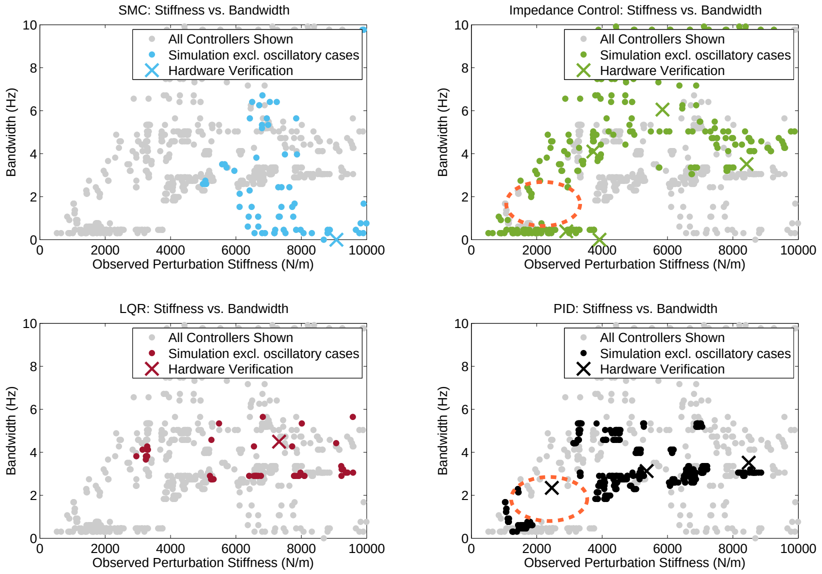

Figure 38: Stiffness vs. bandwidth of benchmark-controlled systems. Bandwidth found using frequency response tests, and stiffness taken from impedance fits.

safe industrial robots [1, 40]. These improvements are certainly plausible; for example, the performance of the benchmark controllers could be improved upon by trading bandwidth for lower stiffnesses at high accuracy, as noted at the end of section 4.3.3.

The stiffness and tracking error target performance is corroborated by looking to an alternative solution approach: improved performance though actuator design. Shen & Goldfarb [76] used two valves to simultaneously control force and stiffness of a pneumatic actuator. They observed control from 2000-3000 N/m to about 14000 N/m, while tracking a position signal over a -20 to 20 mm range, using an 80 psig supply pressure, with position errors in the 5 - 10% range. In this thesis, the aim is to verify that control can be used in lieu of actuator modifications to achieve similar performance gains. Therefore, for the new controller, the most accurate tracking should be achievable with a stiffness of at most 5000 N/m. Additionally, with constraints, a controller should be designed that can achieve lower stiffnesses, ideally as low as 2000 N/m is desirable, with tracking error under 5-10%, or under 2-4 mm.

CHAPTER V

COMPLIANT TRACKING OF A 1-DOF PNEUMATIC SYSTEM WITH MODEL PREDICTIVE CONTROL

Chapter 4 examined the performance of a number of controllers for pneumatic actuators with respect to the trade-off between accurate position control and compliant actuator dynamics. In this chapter, a Model Predictive Controller is designed that couples impedance constraints with optimal control to handle this trade-off and explicitly define the desired performance on a single Degree-of-Freedom (DoF) system.

First, an overview of Model Predictive Control (MPC) is provided, and the particular implementation tools and their limitations are discussed. Next, the MPC architecture for compliant control is introduced using two approaches to compliant constraints: one that uses impedance constraints, and a second one that uses admittance constraints. A dual mode structure is proposed as a means to provide theory-backed stability guarantees. Friction compensation strategies are compared, and an appropriate friction compensation method is selected for use with the MPC. Finally, the controller performance is compared to benchmark controllers and among architecture variations.

5.1 Model Predictive Control

In its general form, illustrated in Figure 39, MPC computes solutions to a constrained, finite-time, optimal control problem over a prediction horizon of length N_p time steps. The state equations are supplied with N_c control inputs, where N_c denotes the control horizon, though in the execution, only the first calculated control input is used. After each execution, states are updated based on plant sensor readings and estimator outputs, and the process starts over.

The controller can be written as a constrained optimal control problem at each sampling

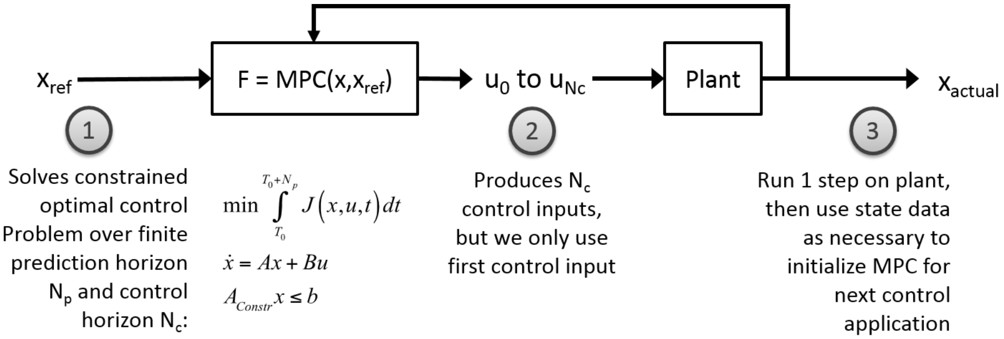
x_ref
F = MPC(x,x_ref)
u_0 to u_Nc
Plant
x_actual
1
Solves constrained optimal control Problem over finite prediction horizon N_p and control horizon N_c:
$\min \int_{T_0}^{T_0+N_p} J(x,u,t)\,dt$
$\dot{x} = Ax + Bu$
$A_{Constr} x \le b$
2
Produces N_c control inputs, but we only use first control input
3
Run 1 step on plant, then use state data as necessary to initialize MPC for next control application

Figure 39: Overview of MPC structure.

instant, consisting of a cost function, state equations, and system constraints:

$$min \quad J \quad = \quad min \sum_{k=T_0}^{N_p} F_k(x,u) \qquad \text{subject to} \quad \begin{cases} x_{k+1} = f(x_k, u_k) \quad k = 1..N_c \\ u_k = 0 \quad k = N_c + 1..N_p \\ g_1(x_k) \le b_{constr} \\ g_2(u_k) \le d_{constr} \end{cases} \tag{58}$$

5.2 Implementation Tools and Limitations

An added challenge is incurred by the tools used for on-line optimization. While general MPC is not restricted to any particular subset of systems, for practical application, it is necessary to use numerical optimization tools that can be solved quickly in real-time

There are a number of optimization tools that are free for academic use, which are of varying ease-of-use and speed. CVXgen [58] is a tool that generates custom, high-speed convex optimization scripts for problems that can be represented in Quadratic Problem (QP) form. This requires quadratic cost functions and constraints that are either convex inequalities or affine equalities (i.e., linear), resulting in the following, slightly more restricted, generalized form:

$$min \sum_{k=T_0}^{N_p} \left(x_k^T Q_k x_k + u_k^T R_k u_k \right) \quad \text{subject to} \quad \begin{cases} x_{k+1} = G x_k + H u_k \quad k = 1..N_c \\ x_{k+1} = G x_k \quad k = N_c + 1..N_p \\ A_{constr} x_k \le b_{constr} \\ C_{constr} u_k \le d_{constr} \end{cases} \tag{59}$$

Q_k and R_k are positive semi-definite matrices, x_k is the state vector, u_k is the input vector, and (A_{constr}, C_{constr}) and (b_{constr}, d_{constr}) are matrices and vectors, respectively, governing state and input limits.

These form requirements can easily be limiting. For example, a direct realization of the posed compliant control problem of optimal tracking with impedance constraints would include constraints of the form $Z \leq Z_{max}$, or potentially a cost function including minimization of an impedance term, $Z^T Q Z$. However, any direct impedance or stiffness definition requires state division: $Z = F/x_2$ or $K = F/x_1$. Similarly, a time-domain realization of the frequency-domain property necessitates nonlinear operators, and even physical interpretations, e.g. pneumatic stiffness based on properties of the air, at least requires some nonlinear relation, such as quadratic terms and state division (see eq. (30) in section 4.1.5). Furthermore, simple term manipulation, e.g. $F \leq K_{max} x$, is not an acceptable substitute since the sign of the state, $sign(x)$ at an arbitrary time is unknown. Instead, constraints have to be designed that approximate this behavior over the prediction horizon, which is achieved by limiting state quantities (e.g. force, where the force limit is updated based on impedance constraints) or by relating states in a way that enforces an impedance/admittance relationship.

5.3 MPC Formulations for Compliant Pneumatic Tracking

Given the afore-mentioned limitations, several problems must be solved in order to feasibly implement a constrained MPC for pneumatic tracking. First, the low-level system needs to be describable by a linear system. Next, the optimal control problem must be defined using a quadratic cost function with constraints that are convex, concave, or affine. Additional cost and constraint terms can be added to ensure stability.

5.3.1 Linear Plant Model for MPC

For a pneumatic system, where most of the nonlinearities arise in the force dynamics, linearization is achieved via numerical or analytical methods, or via closed-loop force control, as discussed in section 4.2.2. The MPC realizations here use a PID force controller, and represent the closed loop dynamics with an under-damped second-order system:

$$\underbrace{\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ F_1 \\ F_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1/m & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ F_1 \\ F_2 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_n^2 \end{bmatrix}}_{B} \underbrace{F_d}_{u} + \underbrace{\begin{bmatrix} 0 \\ F_{frict}(x, x_1, F_1)/m \\ 0 \\ 0 \end{bmatrix}}_{D} \tag{60}$$

As will be discussed in section 5.4, friction is explicitly compensated for, so it can be effectively removed from consideration in the model provided to the predictive controller, resulting in a linear system:

$$\dot{x} = Ax + Bu, \text{ where } A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1/m & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_n^2 \end{bmatrix} \tag{61}$$

$\dot{x} = Ax+Bu$. Since the MPC is a discrete controller, the system is discretized by partitioning the matrix exponential of the augmented state transfer matrix:

$$\phi = \begin{bmatrix} A_{(n\,x\,n)} & B_{(n\,x\,m)} \\ 0_{(m\,x\,n)} & 0_{(m\,x\,m)} \end{bmatrix} \tag{62}$$

$$\Phi = e^{\phi dt} = \begin{bmatrix} G_{(n\,x\,n)} & H_{(n\,x\,m)} \\ \Phi_{n+1:n+m,1:n} & \Phi_{n+1:n+m,n+1:n+m} \end{bmatrix} \tag{63}$$

This results in a discrete, linearized system of the form

$$x_{k+1} = Gx_k + Bu_k \tag{64}$$

An alternative approach that was also investigated was the use of an analytical linearization of the full state dynamics, which would eliminate the use of a restrictive low-level controller and replace the free variables in optimization with terms that directly affect pneumatic dynamics. This extension will be explored in slightly further detail in section 7.3.1.

Next, constraints are defined – first on system performance, and then on compliant tracking, and associated with an appropriate cost function.

5.3.2 Pneumatic System Input & State Constraints

An over-constrained problem is numerically challenging and potentially impossible to solve, so it is necessary to only impose constraints that directly improve performance. As a result, performance constraints were imposed on the input (force reference), u, in the form of a rate constraint, $||\Delta u||_{max}$ and saturating values, u_{min} and u_{max}. This limit effectively enabled the MPC to be over-tuned without leading to instability, i.e., it eliminated the need for a well-selected R matrix. Other operating limits, such as position and velocity constraints, were instead implicitly enforced via feasible reference generation.

Upper and lower bounds for force reference can actually be found from the pressure supply: $u_{max} = P_s A_{cap} - P_e A_{rod}$ and $u_{min} = P_e A_{cap} - P_a A_{rod}$. In theory, these could be higher, since u is a reference and not the actual force state, F, though it was found that treating them as the same and imposing constraints on u was a more robust, if potentially more conservative (and thus limiting) approach. Additionally, since u was used in place of F to enforce constraints on the actuator force, the constraints on u were largely overridden by the impedance/admittance constraints, discussed in sections 5.3.3 and 5.3.4.

A heuristic for $||\Delta u||_{max}$ was found by treating closed loop force tracking as a feedback linearized system, discretizing the system, and then calculating limits on u_{k+1} at each time step as a function of the known maximum valve orifice area, the valve bandwidth, and system dynamics, as detailed in Appendix B. While it would have been possible use dynamic constraints that update Δu at each time step, such constraints proved to be unreliable and problematic, rather than performance enhancing. Instead, a fixed value of $\Delta u = 10/dt$ was selected as a rate limit, based on observed changing rate limits and fine-tuning.

5.3.3 Version 1: Impedance Constraints

Due to the limits on implementation configuration discussed in section 5.2, it is not possible to directly apply compliance constraints. However, since compliance is simply a relation of force and position/velocity states, it is possible to update state constraints that indirectly affect system compliance.

One approach is to use an impedance constraint, enforced via a maximum internal actuator force. With this approach, the internal actuator force reference at each time step is subjected to a state constraint, $F_{min} \leq F_d \leq F_{max}$. The maximum and minimum force values are found using the force limits that would be provided by the instantaneous error, e, and desired maximum impedance (K_{max}, B_{max}, M_{max}):

$$min\ J = min \sum_{i=T_0}^{T_f} e_{act,i}^T Q e_{act,i} + F_{d,i}^T R F_{d,i} dt$$

$$\begin{cases} \dot{x} = Ax + Bu \\ x \leq b_{constr} \quad u \leq u_{constr} \\ e = [e_0\; e_I], \quad e = x_{1,ref} - x_1, \quad \dot{e}_I = e_0 \\ F_d \leq ||Z_{max} e||_{t=0} \\ \qquad = ||K_{max} e_{t=0} + B_{max} \dot{e}_{t=0} + M_{max} \ddot{e}_{t=0}|| \end{cases} \tag{65}$$

Note that the error has been augmented with an integral error term, meaning that Q is at minimum a 2 x 2 matrix that penalizes position and integral error. The error terms can be written as an output in state space form by augmenting the state to include an integral term, $\dot{x}_I = x$, and treating the known reference and integrated reference values as inputs:

$$\begin{bmatrix} e \\ e_I \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} + \begin{bmatrix} x_{ref} \\ x_{ref,I} \end{bmatrix} \tag{66}$$

In practice, a clamping method is used to counteract integral windup, and the integration acts over a moving horizon.

With this constraint approach, the force limits are constant over the prediction horizon. While the method directly enforces the desired behavior at each initial time step, the impedance relation could vary over the prediction horizon as error margins change, but the force limit stays the same. As a result, the constraint is less appropriate over the duration of the prediction horizon. This can also lead to unexpected or undesired results, such as a delayed force response over the prediction horizon.

5.3.4 Version 2: Admittance Constraints

The main issue with the impedance constraint introduced in section 5.3.3 is that it does not enforce the condition over the prediction horizon. One way improve upon this constraint is to implement an admittance relationship over the prediction horizon, in which the tracking error is directly tied to the force limit:

$$min\ \ J = min \sum_{i=T_0}^{T_f} e_{act,i}^T Q e_{act,i} + F_{d,i}^T R F_{d,i} dt$$

$$\text{subject to} \quad \begin{cases} \dot{x} = Ax + Bu \\ x \leq b_{constr} \quad u \leq u_{constr} \\ ||Ke + B\dot{e} + M\ddot{e}|| \leq F_{max}|_{t=0..N_p} \\ e = [e_0\ e_I], \quad e = x_{1,ref} - x_1, \quad \dot{e}_I = e_0 \\ F_{max} = ||Z_{max}e||_{t=0} \\ \qquad\ \ = ||K_{max}e_{t=0} + B_{max}\dot{e}_{t=0} + M_{max}\ddot{e}_{t=0}|| \end{cases} \tag{67}$$

This implementation uses the same cost function and error states introduced in the previous section, but with the proposed constraint changes. The force limit is, of course, still determined using the instantaneous error at the start of each time step, so to some extent, the problem persists. However, the added admittance relation also provides redundancy, which can improve the MPC's chance of finding a feasible solution that satisfies compliance constraints, despite being provided only indirect constraints on compliance relations.

5.3.5 Terminal Cost and Constraints for Stability

A common method for ensuring a stable predictive controller is through the use of an infinite-horizon cost, using a method known as the *dual mode predictions* [56,71]. The basic premise is to design a realizable controller that may be evaluated over an infinite prediction horizon despite requiring a finite number of MPC variables. This is achieved by splitting the predictions into two modes (hence, "dual mode"):

$$u_k = \begin{cases} \text{mode 1: } u_k \text{ determined by MPC optimization, } k = 1..N_p \\ \text{mode 2: } Kx_k, \quad k > N_p \end{cases} \tag{68}$$

Mode 1 acts over the length of the prediction horizon, and mode 2 is in place thereafter, effectively splitting the prediction horizon into near transients (mode 1) and asymptotic behavior (mode 2) [71]. The second mode is enforced using a terminal cost, and terminal constraints are applied to ensure recursive feasibility.

It is important to clarify that the dual mode formulation exists only in the predictions, not the control: the second mode is implicit; it is required to improve stability, and to provide a 'hot' start for the optimal control problem [59,71]. The dual-mode prediction has the effect of allowing a finite horizon optimization to be used to solve an optimal control problem over an infinite horizon. It also ensures that the *tail* – the parts of the predictions made at the previous sample which have yet to take place – are included in the current prediction class [71].

5.3.5.1 General Implementation & Stability Theory

The generalized form of a MPC with terminal cost and constraints based on dual-mode predictions (for the linear case) is seen in equation (69):

$$min\ J = min \sum_{k=T_0}^{N_p} \left(x_k^T Q_k x_k + u_k^T R_k u_k \right) + x_{N_p}^T \bar{Q} x_{N_p}$$

$$\text{subject to} \begin{cases} x_{k+1} = Gx_k + Hu_k \quad k = 1..N_c \\ u_k = 0 \quad k = N_c + 1..N_p \\ A_{constr} x_k \leq b_{constr} \\ C_{constr} u_k \leq d_{constr} \\ A_x x_{N_p} \leq b_x \\ A_u x_{N_p} \leq b_u \end{cases} \tag{69}$$

While any state feedback law may be used for mode 2, it is common and convenient to apply an LQR controller. Since the MPC must remain finite, this is done by choosing a terminal cost that is equivalent to the infinite horizon cost of the mode 2 controller. Then the terminal cost weighting matrix, $\bar{Q}$ is found to be the solution of the Lyapunov equation [56]:

$$\bar{Q} - (A + BK)^T \bar{Q} (A + BK) = Q + K^T RK \tag{70}$$

The second requirement is *recursive feasibility*: since the LQR controller acting over the tail does not inherently obey performance constraints, it is possible for an incarnation of the MPC with dual-mode predictions to be infeasible at a given time-step. However, since the controller is stable and converges asymptotically, as long as the LQR starts from a sufficiently constrained region, it will remain in that region for $t = T_0..\infty$. Since the second controller in the dual-mode prediction – the LQR controller – is state-dependent, its control action, and the state at any time past the prediction horizon $k_{x,u} = t - N_p$ can be explicitly defined as $u = K(A+BK)^{k_u} x_{N_p}$ and $x_k = (A+BK)^{k_x} x_{N_p}$, respectively. Then the aim is to find the number of terminal constraint applications (the values of k_x and k_u) such that the infinite horizon controller inherently satisfies operating constraints for all time $T > N_p + max(k_x, k_u)$. This is found recursively, by solving two linear programs at each time step beyond the prediction horizon, k, until k_u and k_x are found:

$$\begin{aligned}
&\text{Given input constraints } u_{min} \leq u \leq u_{max} \\
&\text{find } k_u = \min_k \text{ such that} \\
&u_{max} \geq \left(\max_x K(A+BK)^{k+1} x \text{ s.t. } u_{min} \leq K(A+BK)^k x \leq u_{max}, k = 1..k_u \right) \\
&u_{min} \leq \left(\min_x K(A+BK)^{k+1} x \text{ s.t. } u_{min} \leq K(A+BK)^k x \leq u_{max}, k = 1..k_u \right)
\end{aligned} \tag{71}$$

and for state constraints:

$$\begin{aligned}
&\text{Given state constraints } x_{min} \leq x \leq x_{max} \\
&\text{find } k_x = \min_k \text{ such that} \\
&x_{max} \geq \left(\max_x (A+BK)^{k+1} x \text{ s.t. } x_{min} \leq (A+BK)^k x \leq x_{max}, k = 1..k_x \right) \\
&x_{min} \leq \left(\min_x (A+BK)^{k+1} x \text{ s.t. } x_{min} \leq (A+BK)^k x \leq x_{max}, k = 1..k_x \right)
\end{aligned} \tag{72}$$

Stability can be shown in the Lyapunov sense by using $J(k)$ as a Lyapunov function: $V(x(k)) = J(k)$. Provided J(k) is a positive definite function of x(k) and the terminal weight is chosen so that $J(k)$ is equal to the infinite horizon cost, and the optimal predicted input sequence at time k is feasible at each subsequent time step, then the optimal predicted cost is non-increasing and satisfies $J(k+1) - J(k) \leq -[x_k^T Q x_k + u_k^T R u_k]$ along closed-loop trajectories. Together, these conditions ensure asymptotic convergence , i.e. $-[x_k^T Q x_k + u_k^T R u_k] \longrightarrow 0$ as $k \longrightarrow \infty$. A clear, thorough overview of the stability approach used here may be found in [15], though it is also more thoroughly documented in sources such as [56, 71].

5.3.5.2 Implementation for Compliant Control

For the compliant control MPCs introduced in sections 5.3.3 and 5.3.4, a predictive controller was designed that uses dual-mode predictions mode 2 defined by the LQ-optimal feedback control law, as discussed in the previous section. Since input constraints exist on the input and its rate change, the state was augmented to treat Δu as the input, and include u as a state:

$$x_k = [\begin{array}{ccccc} x_1 & x_2 & F_1 & F_2 & F_d \end{array}]^T, \quad u_k = \Delta F_d \tag{73}$$

Then the state relation was found using an augmented matrix:

$$\hat{x}_{k+1} = \underbrace{\begin{bmatrix} G & H \\ 0 & 1 \end{bmatrix}}_{\hat{G}} \hat{x}_k + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\hat{H}} \hat{u}_k \tag{74}$$

$$min\ \ J = min \sum_{k=T_0}^{N_p} \left(e_k^T Q_k e_k + u_k^T R_k u_k\right) + e_{N_p}^T \bar{Q} e_{N_p}$$

$$\text{subject to} \quad \begin{cases} \hat{x}_{k+1} = \hat{G}\hat{x}_k + \hat{H}\hat{u}_k \quad k = 1..N_c, \quad N_c = N_p = 15 \\ A_{constr} x_k \le b_{constr} \\ u_{min} \le u_k \le u_{max} \\ A_x x_{N_p} \le b_x \\ u_{min} \le A_u x_{N_p} \le u_{max} \end{cases} \tag{75}$$

Ideally, the linear programs would be solved on-line at each time-step, but such a solution is computationally expensive and very possibly unnecessary. Instead, for time-varying constraints (F_{max}, Z_{max}), low values were provided and used to estimate a maximum set of constraints, for which it was found that, at 100 Hz, $k_u = 4$ and $k_x = 3$, using the script found in the Appendix in section B.2.

5.3.5.3 Effect on Performance

The use of a terminal cost and added constraints did increase the prevalence of CPU overload problems with the target. As detailed in the Appendix in section B.3, CPU overloads occur when hardware delays cause unexpected jumps in the computational workload,

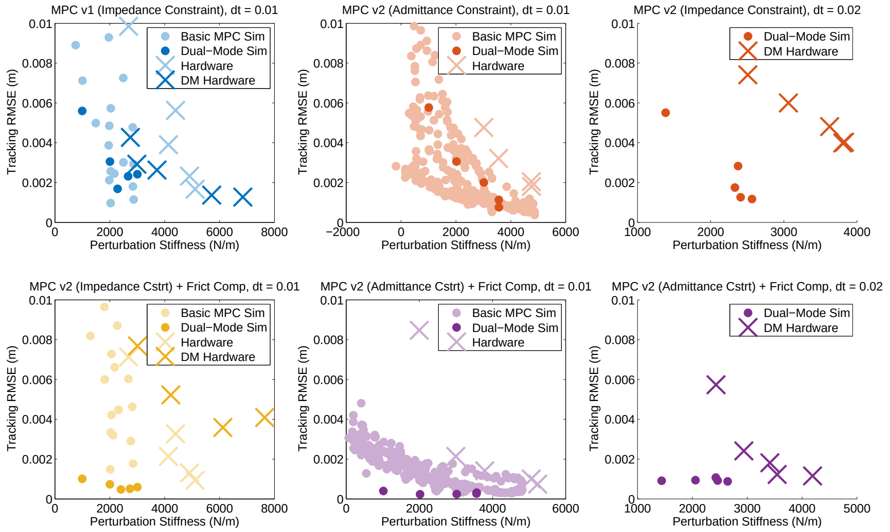

Figure 40: Comparison of MPC with and without dual-mode predictions (terminal cost and constraints) in simulation and hardware.

leading to sudden program failure. It is somewhat random and hardware dependent, and part of the problem is that – due to constraints on the project – the targets used for this work are relatively old, less robust than newer targets running Real-Time Workshop, and have fairly modest specifications. However, CPU overloads are also affected by general computational load, and the increase in CPU overloads suggests that the added cost and constraints make the optimization harder to solve for an arbitrary initial state.

As a result, while simulated results using the predictive controller with a terminal cost and constraints were easily obtained, hardware results were limited. Frequency-fitted stiffness values are unavailable, so instead, the stiffness was found using the force perturbation test (by dividing the observed interaction force by the position error incurred by the perturbation). Additionally, the admittance constrained version (v2) could only be run consistently in hardware at a slower sampling rate of 50 Hz. Therefore, the overall results are compared across three tests: simulation and hardware at 100 Hz, simulation at 100 Hz, and simulation and hardware at 50 Hz.

Figure 40 illustrates how the addition of the dual-mode cost and terminal constraint

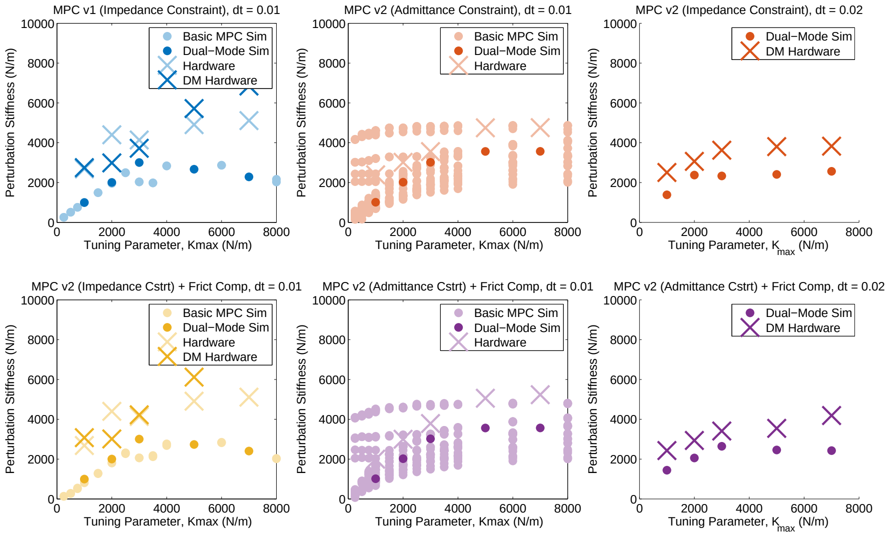

Figure 41: Comparison of compliance constraints on MPC with and without dual-mode control configuration in simulation and hardware.

affected basic performance. A simulation comparison showed that the MPC without dual-mode characteristics already behaved in a stable manner, though it did improve simulated accuracy when coupled with friction compensation. Additionally, it improved the simulated Version 2 (Admittance Constraint), making it behave more like the expected version, meeting an impedance limit until the maximum impedance asymptote was achieved.

While the control with dual-mode predictions did improve accuracy in simulation, it was again shown that at the recorded stiffnesses, these results were optimistic. Primarily, the addition of the terminal constraint served to make the simulated admittance constraint more reflective of the trend seen in hardware – that MPC obeys the performance constraints, given some modeling error, as illustrated in Figure 41. In general, the MPC with dual-mode predictions didn't change performance significantly relative to the lighter MPC versions without added terminal constraints. In other words, the addition of the terminal cost and constraints was overly conservative for this system, a trend that has been observed on a number of systems documented in literature. One explanation is that with a sufficiently long prediction horizon, the impact of the mode 2 controller on the predicted states will be small and the terminal constraints effectively enforced by the horizon length, thus making its addition negligible [71]. This suggests that for this system, a 15 step prediction horizon at 100 Hz is sufficient to adopt infinite horizon properties and ensure system stability. The implications for the broader set of general systems is discussed next in section 5.3.5.4.

5.3.5.4 Limitations and Alternatives

The predictive controller formulation introduced in sections 5.3.5.1 and 5.3.5.2 has all the elements necessary for stability guarantees in the Lyapunov sense for the most general MPC case, as outlined by Mayne, Rawlings, and Scokaert in their seminal paper [59]: an appropriate terminal cost or, if possible, a sufficiently large horizon, and terminal constraints. Naturally, these guarantees require that several assumptions are satisfied: the prediction assumptions must be reproducible at subsequent sampling instants, the references must be feasible, and the hardware/software must have the capability to solve the larger optimization problem.

First, in order for the infinite horizon cost to be accurate, the system model must be a good representation of the actual state dynamics, such that it can fairly be represented with a linear time-invariant model, and the reference should be sufficiently smooth to treat the tracking problem like an error regulation one. Furthermore, the supplied references must have feasible solutions with the provided state and input constraints. While input constraints are easily satisfied, state constraints are more difficult to satisfy and may conflict with input constraints [71]. If the constraints cannot be satisfied, the stability proof is invalid. Finally, while a well-designed dual-mode prediction configuration is effective (as observed with the system tested in this thesis), the addition of terms to the cost function and added constraints can make computation more difficult, which in turn requires better solvers or more computational power [74].

In the absence of applicability of the dual-mode criteria, there are a number of approaches that would provide suitable alternatives. For example, early MPC researchers showed that by choosing a sufficiently long horizon, a predictive controller acting on a stable plant achieves the properties associated with an infinite horizon [59]. Alternatively, if the underlying plant is stable, the system may be treated as a "reference governor" [7,32,37,50]:

the optimal component (MPC) is used to generate feasible references that the underlying system, a stable plant, can follow. If that system is linear, then it is stable independent of the provided reference, and the governor's primary aim is to ensure performance. If the system is nonlinear, the reference governor may use an algorithm to select a feasible reference that is both satisfactory and preserves stability of the modeled system. In these cases, successful stability guarantees necessitate that the system model is well-defined and representative of the actual behavior (e.g., that the system dynamics are fairly described by an LTI model), and the constraints on performance must be well-characterized and clearly communicated in the controller to ensure generation of feasible references.

In general, researchers have tended towards the explanation that feasibility is sufficient [74]. Chen [17] offers a more relaxed stability criterion based on the observation that many predictive controllers without terminal constraints are nonetheless stable. He points out that while in the classic case the terminal constraint is intended to cover the cost-to-go for the remaining horizon, a simpler condition requires only that the stage cost at time $k + N$ be less than at time k, which relies on the monotonicity of the associated Lyapunov function. From there, a number of theorems are developed that span most controller variations – from unconstrained, linear MPC to constrained, nonlinear MPC – that essentially define feasibility conditions required for stability. In general, this result – feasibility implies stability – appears to provide a good guiding intuition.

5.3.5.5 Summary of Stability for Compliant MPC

The dual-mode formulation provides theory-backed stability guarantees for the predictive controllers introduced in sections 5.3.3 and 5.3.4, while preserving the benefits of compliant control seen without its addition. However, parameter variation studies show that its addition is unnecessary, and that the original MPC formulation is already stable. The lack of necessity is effectively explained by less restrictive MPC stability theory, which suggests that feasibility implies stability. Furthermore, the underlying system consists of a stable force controller and position dynamics that are marginally stable the absence of damping, but are stable in any practical system with friction, viscous damping, etc. (as was used in both the model and in hardware). The stability of the internal model dynamics, coupled with the conservative constraints on feasible force references defined in section 5.3.2, is sufficient to ensure stability and verify the performance shown in simulation and practice.

For general systems, it is advisable to design low-level dynamics that are inherently stable, either via mechanical design or feedback control. Then a reference governor can be designed to ensure stability, or a sufficiently large prediction horizon may be found that provides stable performance. Alternatively, if the dynamics are well described by an LTI model and the system is sufficiently capable, a dual-mode formulation provides stability guarantees while preserving compliance constraints.

5.4 Friction Compensation

Friction is one of the most evident disturbances that exists in pneumatic tracking. It is inherently in the system due to seal dynamics, and while most manufacturers have made strides in reducing contact friction, it is still prevalent – e.g., the Bimba PFC cylinders used in this thesis were "low-friction" cylinders, yet they still experience 1 -5 N of friction even under idealized operating conditions, as discussed in Chapter 3. Additional friction sources may exist due to interacting platform components, system geometry, etc.

Friction is commonly offset using an additive feed-forward term based on a friction model – here termed "additive" compensation. The method is especially effective when the friction model is well-sourced, consistent, and the system is responsive, i.e. a fast system can offset sudden stiction disturbances better than a slow system. The most significant problem, then, is static friction, or stiction, which arises suddenly, at key times (changes in direction and starting motions), and is typically relatively large in magnitude, compared to viscous friction. Alternatively, methods such as high-frequency dither – where the actuator is pulsed a small distance at a high frequency – may be used to attempt to prevent the system from ever really stopping, and thereby keep the system out of the stiction regime. This method can be effective on repeatable, motion control platforms, but necessitates a constant input, and its effect tends to be dampened by significant actuated geometry (e.g.,

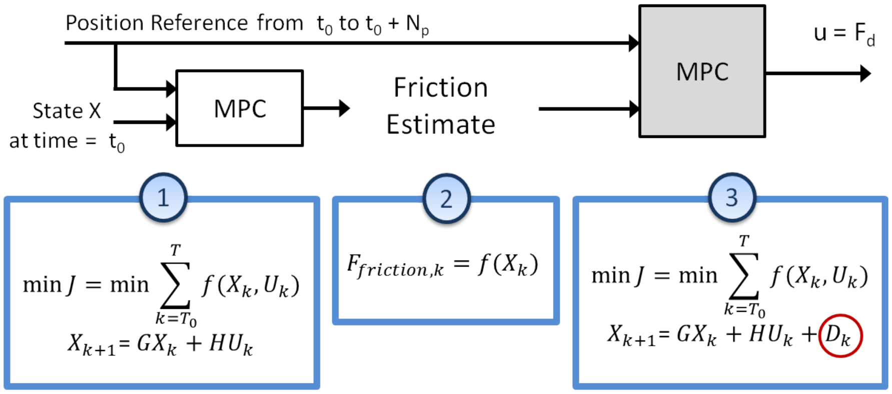
Position Reference from t_0 to $t_0 + N_p$
State X
at time = t_0
MPC
Friction
Estimate
MPC
u = F_d
1
$\min J = \min \sum_{k=T_0}^{T} f(X_k, U_k)$
$X_{k+1} = GX_k + HU_k$
2
$F_{friction,k} = f(X_k)$
3
$\min J = \min \sum_{k=T_0}^{T} f(X_k, U_k)$
$X_{k+1} = GX_k + HU_k + D_k$

Figure 42: Friction prediction & compensation.

a robot arm).

In model predictive control, disturbances are treated as additional actuator inputs with some estimated known value over the prediction horizon, D_k:

$$x_{k+1} = G_k x_k + H_k u_k + D_k \tag{76}$$

Given a well-constructed estimated signal, this approach allows the system dynamics to be included in the compensation effort, which improves compensation for slow systems. In fact, predictive control can be used to design an estimator in this way that improves the compensation capability of pneumatic systems in certain cases, as will be shown in section 5.4.1. However, the effect comes at added cost, and does not extend well to systems that are already operating at their constrained optimal limits – as is the case with compliant control – though, it could be improved with possible architectural changes. Instead, an additive compensation method, detailed in section 5.4.2, was used with satisfactory results.

5.4.1 Cascaded Compensation

In the cascaded MPC approach to friction compensation, two predictive controllers are sequenced, so that the first one acts as an estimator, and the second as a controller. Outputs from the first controller generate state estimates over the prediction horizon that are used by a friction model to estimate the friction dynamics over that horizon. The second controller then produces an output that considers these predicted friction effects.

A diagram of this approach is shown in Figure 42. There are three steps:

1. The first MPC minimizes the specified cost function, subject to linear discrete system dynamics, i.e. it solves the optimal control problems defined by either (65) or (67), depending on constraint choices, subject to model dynamics $x_{k+1} = Gx_k + Hu_k$.

2. A friction model uses the N_p time steps of state information to approximate friction over the prediction horizon, which is passed on to the second MPC.

3. The second MPC again minimizes the same cost function as in step 1, but subject to slightly modified state dynamics; the state equation is described by

$$x_{k+1} = G_k x_k + H_k u_k + D_k \tag{77}$$

   where D_k is an offset vector, consisting of the expected friction values. The second MPC now produces an input that will compensate for the disturbance effect caused by friction, but still subject to optimality and performance constraints as specified in the predictive controller design.

One way to look at this system is as a linear system with a prescribed input, i.e. rather than treating D_k as a separate term in equation (77), it could be lumped into the vector $H_k u_k$ as a prescribed value at each time step.

It is relatively straightforward to demonstrate the value of a cascaded MPC, both analytically and in simulation.

5.4.1.1 Value Demonstrated by Analysis

A standard derivation of unconstrained MPC, including friction as an unknown disturbance term, D_k, shows how a disturbance estimate affects inputs for slow systems.

For a system that satisfies

$$x_{k+1} = Ax_k + Bu_k + D_k, \quad y_{k+1} = Cx_k \tag{78}$$

the goal is to find an input, u_k^*, that minimizes a cost function, J:

$$J(X) = (Y_c - \tilde{Y})^T (Y_c - \tilde{Y}) + U^T RU \tag{79}$$

where Y and $\tilde{Y}$ are the actual and expected outputs for the next N_p time steps, and U is the matrix of corresponding inputs. N_p is the prediction horizon.

For the unconstrained case, the input is easily found by setting $\frac{\partial J}{\partial U} = 0$ and solving for U. However, this requires defined expected outputs $\tilde{Y} = \left[\tilde{y}_{k+1}\ \tilde{y}_{k+2}\ \tilde{y}_{k+3} \ldots \tilde{y}_{k+N_p}\right]^T$ over the prediction horizon, which are presented in matrix form:

$$\tilde{Y} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}}_{K_{CA}} x_k + \underbrace{\begin{bmatrix} CB & 0 & 0 & \ldots & 0 \\ CAB & CB & 0 & \ldots & 0 \\ CA^2B & CAB & CB & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \ldots & CB \end{bmatrix}}_{M_{CAB}} \underbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \\ \vdots \\ u_{k+N_p-1} \end{bmatrix}}_{U}$$

$$+ \underbrace{\begin{bmatrix} C & 0 & 0 & \ldots & 0 \\ CA & C & 0 & \ldots & 0 \\ CA^2 & CA & C & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1} & CA^{N_p-2} & CA^{N_p-3} & \ldots & C \end{bmatrix}}_{M_{CA}} \underbrace{\begin{bmatrix} D_k \\ D_{k+1} \\ D_{k+2} \\ \vdots \\ D_{k+N_p-1} \end{bmatrix}}_{D} \tag{80}$$

Using this syntax, the cost function derivative – a vector derivative – can be expanded:

$$\frac{\partial J}{\partial U} = 0 = \frac{\partial f}{\partial Z}\frac{\partial Z}{\partial U} + \frac{\partial g}{\partial U} \quad \text{where} \quad J = \underbrace{Z^T Z}_{f(U)} + \underbrace{U^T R U}_{g(U)} \tag{81}$$

$$= Z^T(I + I^T)(-M_{CAB}) + U^T(R + R^T) \tag{82}$$

$$= 2(Y_c - \tilde{Y})^T(-M_{CAB}) + 2U^T R \tag{83}$$

$$= 0/2 = (Y_c - K_{CA}x_k - M_{CAB}U - M_{CA}D)^T(-M_{CAB}) + U^T R \tag{84}$$

$$= 0/2 = (-M_{CAB})^T(Y_c - K_{CA}x_k - M_{CAB}U - M_{CA}D) + R^T U \tag{85}$$

$$= -(M_{CAB})^T(Y_c - K_{CA}x_k - M_{CA}D) + (M_{CAB}^T M_{CAB} + R^T)U \tag{86}$$

Since R is a scalar, $R^T = R$. Making this substitution, and solving for U,

$$U = \underbrace{(M_{CAB}^T M_{CAB} + R)^{-1} M_{CAB}^T}_{K_{MPC}}(Y_c - K_{CA}x_k - M_{CA}D) \tag{87}$$

K_{MPC} is the MPC gain. It is possible to extend this solution to consider a state cost, Q, which results in the modified result:

$$U = \underbrace{(M_{CAB}^T Q M_{CAB} + R)^{-1} M_{CAB}^T Q}_{MPC}(Y_c - K_{CA}x_k - M_{CA}D) \tag{88}$$

Clearly, U is affected by the presence of the disturbance, D.

To obtain a better understanding of how much a jump discontinuity affects a slow system vs. a fast one, a simple system with first order force tracking (defined by a time constant, τ: $\dot{F} = \frac{1}{\tau}(F_{ref} - F)$) is assumed. Using a (standard) single term matrix exponential approximation of $e^{A\Delta t} \approx I + A\Delta t$:

$$\begin{bmatrix} x_1 \\ x_2 \\ F \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 - \frac{\Delta t}{\tau} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ F \end{bmatrix}_{k} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t}{\tau} \end{bmatrix} u_k + \begin{bmatrix} 0 \\ d_k \\ 0 \end{bmatrix} \tag{89}$$

$$y_{k+1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_{k+1} \tag{90}$$

As an example, U is calculated for $t = 1 \ldots N_p = 4$:

$$\begin{aligned}
U = \begin{bmatrix} u_{11} & u_{12} & 0 & 0 \end{bmatrix}^T \quad & \text{where} \\
u_{11} = & \frac{\left(R\,dt^3\,y_3 + 3\,R\,dt^3\,y_4\right)\,\tau^3 + \left(-R\,dt^4\,y_4\right)\,\tau^2 + \left(dt^9\,y_3\right)\,\tau}{R^2\,\tau^4 + \left(11\,R\,dt^6\right)\,\tau^2 + \left(-6\,R\,dt^7\right)\,\tau + \left(dt^{12} + R\,dt^8\right)} \\
& - \frac{dt^4\,\tau\,\left(dt^6 - 2\,R\,dt\,\tau + 7\,R\,\tau^2\right)}{R^2\,\tau^4 + R\,dt^8 - 6\,R\,dt^7\,\tau + 11\,R\,dt^6\,\tau^2 + dt^{12}}\,D_2 \\
& - \frac{dt^4\,\tau\,\left(3\,R\,\tau^2 - R\,dt\,\tau\right)}{R^2\,\tau^4 + R\,dt^8 - 6\,R\,dt^7\,\tau + 11\,R\,dt^6\,\tau^2 + dt^{12}}\,D_3 \\
& - \frac{dt^4\,\tau\,\left(2\,dt^6 - 3\,R\,dt\,\tau + 11\,R\,\tau^2\right)}{R^2\,\tau^4 + R\,dt^8 - 6\,R\,dt^7\,\tau + 11\,R\,dt^6\,\tau^2 + dt^{12}}\,D_1 \\
u_{12} = & \frac{\left(R\,dt^3\,y_4\right)\,\tau^3 + \left(dt^9\,y_4 - 3\,dt^9\,y_3\right)\,\tau + dt^{10}\,y_3}{R^2\,\tau^4 + \left(11\,R\,dt^6\right)\,\tau^2 + \left(-6\,R\,dt^7\right)\,\tau + \left(dt^{12} + R\,dt^8\right)} \\
& - \frac{dt^4\,\left(dt^7 - dt^6\,\tau + 2\,R\,\tau^3\right)}{R^2\,\tau^4 + R\,dt^8 - 6\,R\,dt^7\,\tau + 11\,R\,dt^6\,\tau^2 + dt^{12}}\,D_2 \\
& - \frac{dt^4\,\left(dt^6\,\tau + R\,\tau^3\right)}{R^2\,\tau^4 + R\,dt^8 - 6\,R\,dt^7\,\tau + 11\,R\,dt^6\,\tau^2 + dt^{12}}\,D_3 \\
& - \frac{dt^4\,\left(2\,dt^7 - 3\,dt^6\,\tau + 3\,R\,\tau^3\right)}{R^2\,\tau^4 + R\,dt^8 - 6\,R\,dt^7\,\tau + 11\,R\,dt^6\,\tau^2 + dt^{12}}\,D_1
\end{aligned} \tag{91}$$

As $R \to 0$ and input is not penalized, the significance of the time constant becomes more evident:

$$U = \begin{bmatrix} \left(\frac{y_3}{dt^3} - \frac{1}{dt^2}\,D_2 - \frac{2}{dt^2}\,D_1\right)\tau \\ \frac{y_3}{dt^2} - \frac{3\,y_3 - y_4}{dt^3}\,\tau - \frac{\tau}{dt^2}\,D_3 - \frac{dt-\tau}{dt^2}\,D_2 - \frac{2\,dt - 3\,\tau}{dt^2}\,D_1 \\ 0 \\ 0 \end{bmatrix} \tag{92}$$

For comparison, in the direct (additive) compensation case, the inputs are found assuming $D_k = 0$, and then D_k/dt is added to each step of the input, u_k. As $\tau \to dt$, these values become closer to each other. However, if τ is very large, then the compensation required by terms is significantly higher than what will be provided by direct compensation, and the lack of compensation for later disturbance terms has notably more impact.

5.4.1.2 Value Demonstrated by Simulation

Alternatively, the impact of predictive friction compensation for systems with slow dynamics may be tested by implementing the controller on systems with different levels of responsiveness. A generic system is defined with dynamics:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= F/m \\ \dot{F} &= \dot{F}_d - \beta_0\,(F - F_d) \end{aligned} \tag{93}$$

where x_1, x_2, and F are the position, velocity, and force due to differential pressures, respectively. Compensation methods are compared for three values of β_0, with three approaches to friction compensation:

1. MPC with no friction compensation (baseline)

2. MPC using a feed-forward friction compensation term based on an instantaneous estimate of the friction

3. MPC with predictive friction compensation, termed "Cascaded MPC" for its use of sequenced predictive controllers

In all cases, the friction models used for simulation and control – a Stribeck-tanh model – were the same, so the simulation is essentially a best-case scenario for the simplest version

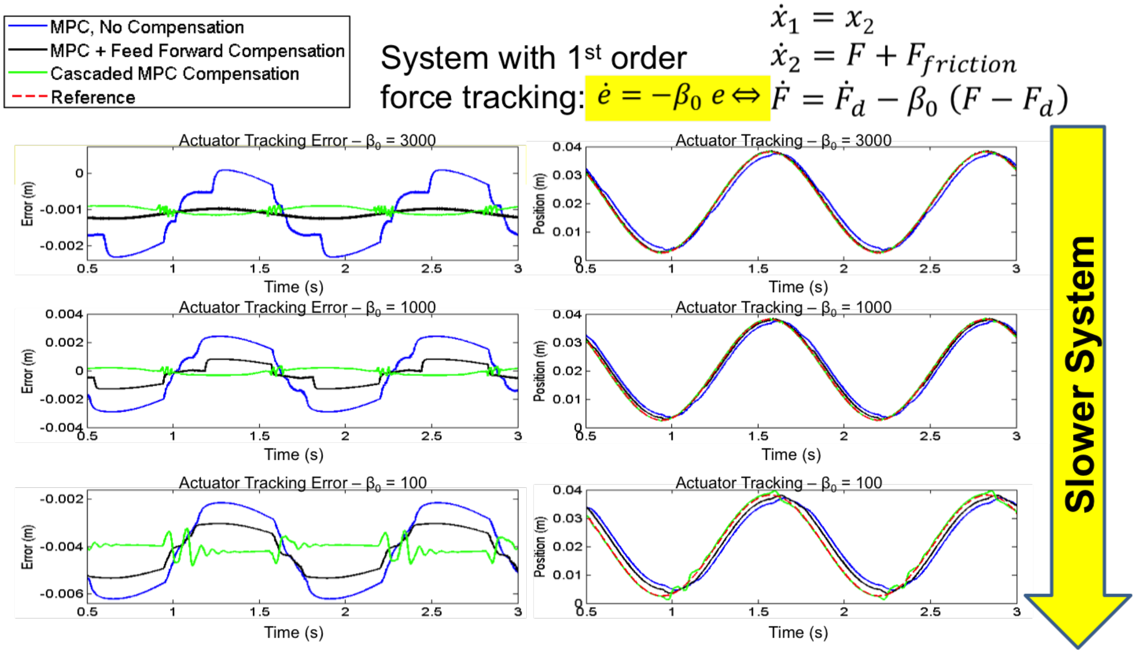

Figure 43: Friction compensation comparison for a system with varying β_0.

of each compensation technique. The results are illustrated in Figure 44. The plots on the left show tracking error, while those on the right give an idea of the general shape of system tracking. It can be clearly seen that as β_0 is decreased and the system becomes slower, the cascaded compensation approach becomes superior to the additive compensation method, resulting in better tracking and significant error reduction.

5.4.1.3 Application to Pneumatic Systems

The cascaded compensation method was applied to the pneumatic actuator model in simulation, which verified that it behaves like a slow system, and that the predictive friction compensation strategy provides similarly promising results, as seen in Figure 44. It is worth pointing out that the error levels observed by the predictive friction in simulation are unprecedented; the 1-mm order tracking accuracy achieved by the feed-forward (Additive) compensation is comparable to the best state-of-the-art, and certainly sufficient. Therefore, the added gains, while significant, are possibly unnecessary, given the additional computational cost. When applied the the compliant control architecture presented in section 5.3, the predictive compensation failed to significantly improve error tracking. Instead, it

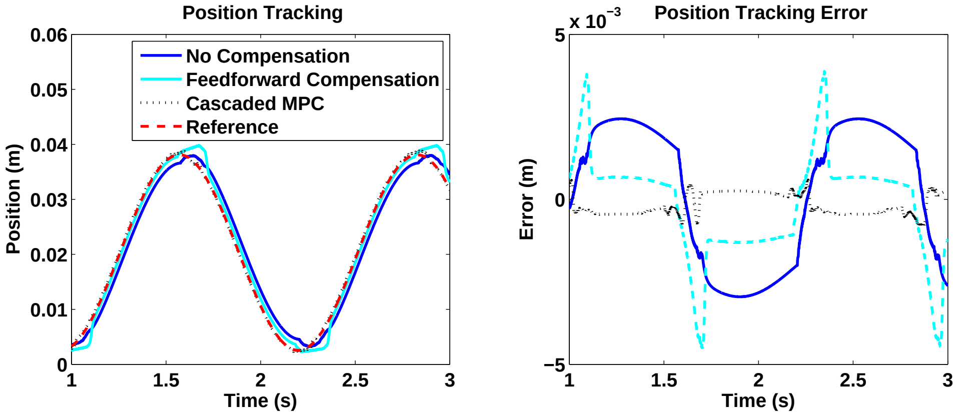

Figure 44: Friction compensation comparison for a simulated pneumatic positioning system.

simply reached the same error as achieved by the system without predictive compensation, as illustrated in Figure 45. Presumably, this is because the MPC for compliant tracking is already subjected to stringent constraints, thereby limiting freedom for dealing with predictive compensation. An alternate strategy would be to also adjust constraints at each operating point to reflect friction needs, though that would potentially violate impedance limitations. Additionally, the predictive compensator doubles the computational load due to on-line optimization, which can have a significant impact on performance. Ultimately,

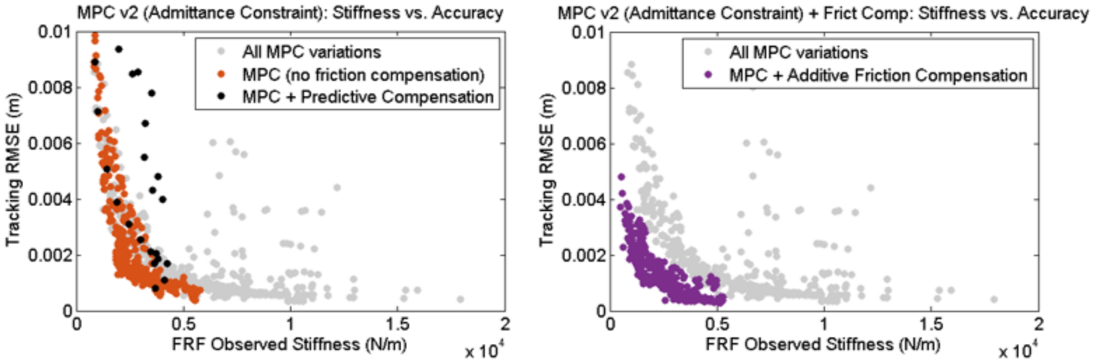

Figure 45: Left: MPC (v2) without friction compensation is compared to MPC with predictive friction compensation. Right: MPC with additive friction compensation. All results shown are from simulation.

the predictive compensator was viewed as an asset for general application, but was not further applied for compliant tracking, which instead used an additive approach.

5.4.2 Additive Compensation

In direct compensation, friction is simply estimated using a pre-existing friction model, and then a counter-term is provided to the force reference signal:

$$(F_d)_{act} = F_d - F_{frict} \tag{94}$$

Then the extra term will cancel out friction, as seen by substituting equation (94) into equation (60). Several models were investigated, including velocity dependent models like Stribeck friction and approximate versions, discussed in Chapter 3, and an exponential decay based approximation previously used by [89]:

$$F_{friction} = -sign(\dot{x}) \left(a_0 + a_1 e^{-b|\dot{x}|} \right) \tag{95}$$

In equation (95), the sum of a_0 and a_1 is the stiction force, and b is the slip constant, and simple Coulomb friction: A force-dependent model was also tested:

$$F_{friction} = -sign(\dot{x}) min \left(F_{Press}, F_s \right) \tag{96}$$

where F_s is a constant representing the stiction force. In simulation and hardware, however, the Stribeck-Tanh model provided the best results, and thus was used throughout.

The results observed in hardware in section 5.5 actually use the reference as a model source rather than the measured sensor readings. This need arises from a lack of relevant sensors: since the friction models are velocity-based, they require velocity measurements, but the hardware only achieves velocity signals via differentiation of the actuator position measurement, which incurs both added noise and time delays. This hypothesis was verified by the fact that the actual signal was effective in simulation – which is free of the delay and noise problems encountered in hardware – and with the reference-based signals, which are equivalent to a phase-lag-free, clean sensor measurement for the observed tracking. Further implementation would be improved through velocity measurements, or potentially integrated accelerometer measurements, coupled with velocity estimation.

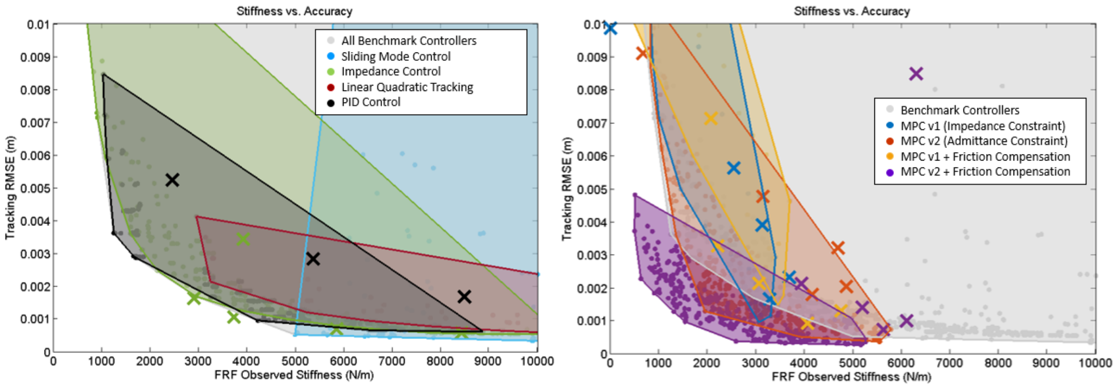

Figure 46: Left: summary of benchmark stiffness vs. accuracy. Right: MPC stiffness vs. accuracy. Legends show colors that correspond to each controller. In both plots, dots represent simulation data from parameter variation studies, and X's represent hardware data. The convex hull of simulation results from each respective controller is indicated by the shaded region in the corresponding color.

5.5 Results: Comparison to Benchmarks

The goal of MPC for compliant control is to ensure that the system follows an explicit approach to handling accuracy and compliance trade-offs: safe interaction is defined as a bound, and optimal tracking is desired within that prescribed limit. Good performance is achieved if the constraints are enforced. Additionally, the MPC should optimize tracking at each performance bound, so it should improve or match tracking achieved by benchmarks subject to a compliance bound. Finally, if there is a critical minimum impedance needed for optimal tracking, the MPC should obtain it automatically, i.e. the impedance constraint should act as an upper limit, not a target impedance.

Results showed that most of these aims were achieved. Figure 46 shows the overall trend of stiffness vs. accuracy for benchmarks and MPC-controlled systems. As the benchmark analysis in Chapter 4 demonstrated, most controllers experience a clear trade-off of accuracy and compliance. MPC is no exception, but the compliance is low, and the best tracking capability at least matches – or even improves upon – benchmark tracking.

The performance for each individual MPC is laid out in Figures 47 - 51. From Figure 47, it is evident that while hardware validated the observed trends seen in simulation, the simulated accuracy was generally lower than what was measured in hardware – a trend

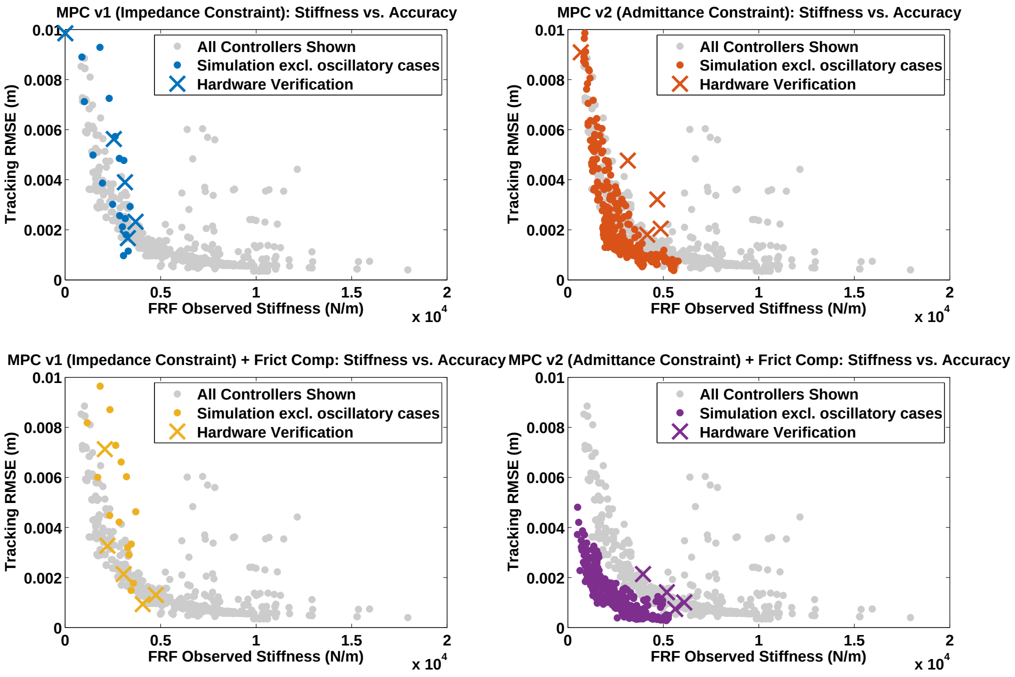

Figure 47: Breakdown of MPC variations: stiffness vs. accuracy.

mostly matched among benchmark controllers. The one exception was impedance control, which did better in practice than in simulation, due predominantly to well-tuned parameter compensation terms.

The tests also verified that friction compensation is a critical and necessary component to improving tracking performance among predictive controllers. While some of the benchmark controllers were able to obtain peak accuracy without an explicit friction compensation model, it is evident from Figures 46 and 47 that friction compensation is necessary to obtain the best MPC performance. The best results are achieved using the admittance constraint, and the added friction terms have minimal downside on other performance metrics.

The MPCs also offered other benefits – for example, good bandwidth for low-stiffness systems. The admittance constrained system performed slightly better, providing a 3-4 Hz bandwidth for tracking subject to a 2000 - 4000 N/m stiffness bound, as observed in Figure 48, whereas impedance control suffers very low bandwidth (< 1 Hz) for stiffnesses under 3000 N/m.

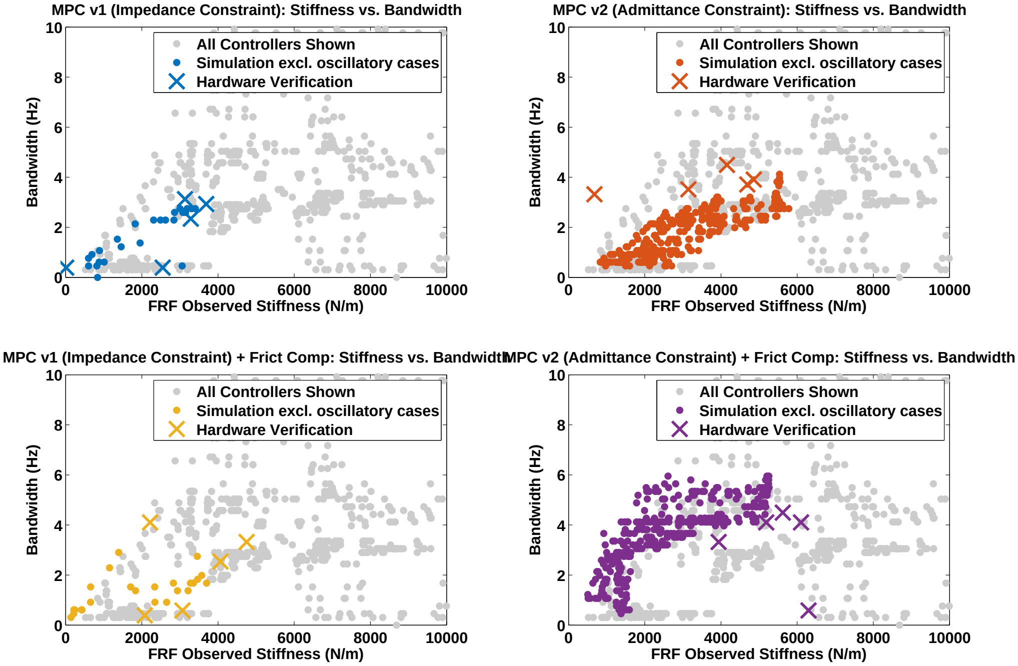

Figure 48: Breakdown of MPC variations: stiffness vs. bandwidth.

It is also useful to point out that interaction forces among MPCs are characteristically low: as illustrated in Figure 49, they are generally under 50 N, even with friction compensation.

Finally, a major asset of the MPC is that it converges to the minimum impedance needed to ensure optimal performance, as shown in Figures 50 and 51. In the figures, the four MPCs are compared to the impedance controller and tuned using a stiffness bound (for the impedance controller, the stiffness is a target). It is clear from the earlier plots that peak performance is achieved between 3000 and 5000 N/m. At that point, the MPC levels off, as further stiffness is unnecessary. However, in the impedance controller, the impedance continues to rise, as it is a target. In other words, the controllers satisfy the prescribed stiffness bounds and provide decreased stiffness for the same tracking performance where possible. An a-priori system understanding beyond the model is unnecessary – i.e., while it would certainly be possible to conduct this analysis and then define an impedance controller with gains that reflect the minimum impedance, it would be inconvenient and non-conducive

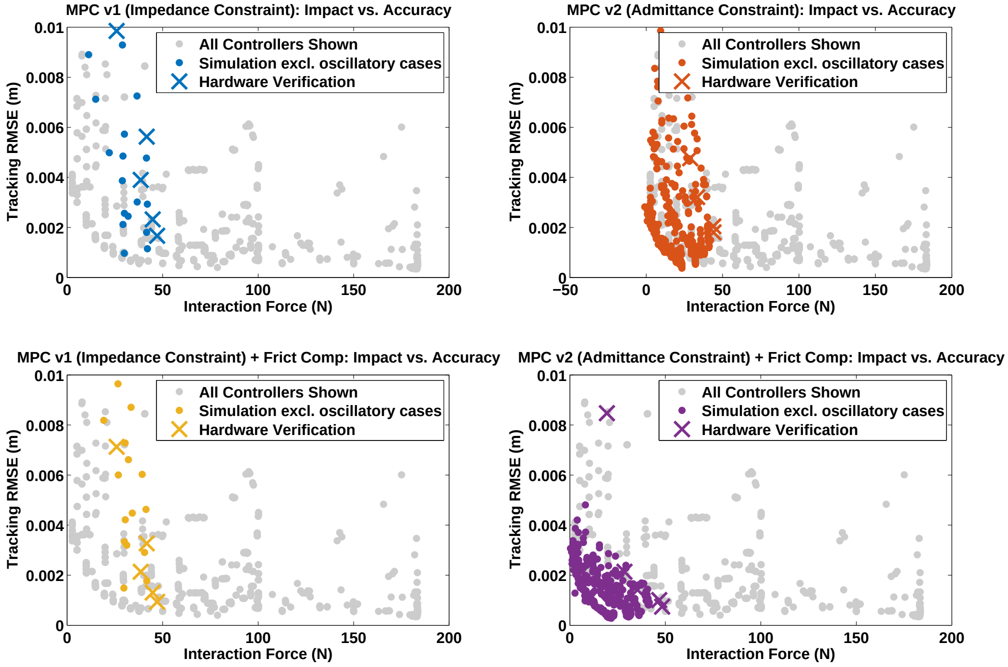

Figure 49: Breakdown of MPC variations: maximum interaction force vs. tracking accuracy.

.

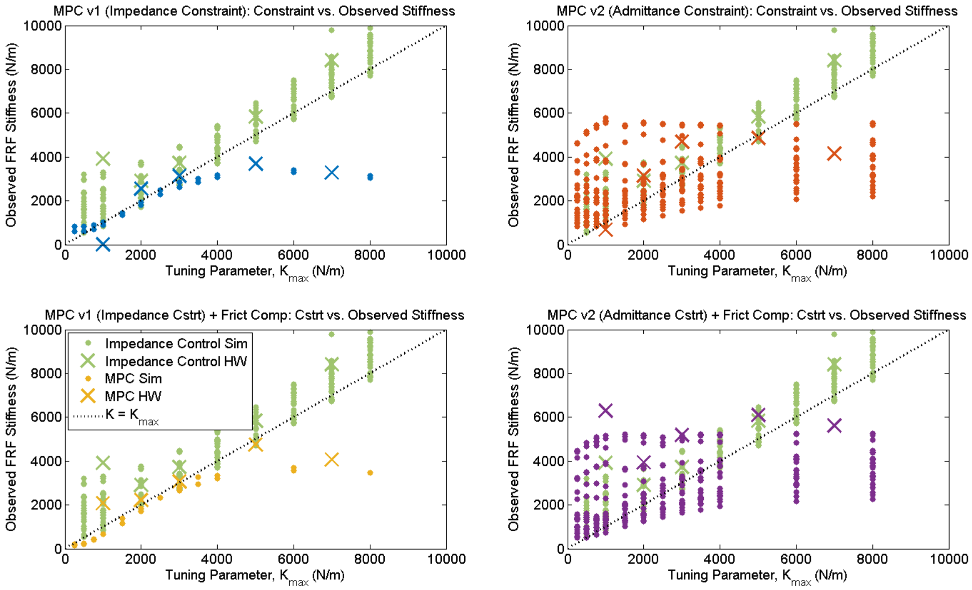

Figure 50: Impact of impedance constraint on MPC, and comparison to impedance controller with target stiffness. Stiffness judged by frequency response fit.

to systems with changing or more complex dynamics. The MPC eliminates this step by simply ensuring that a constraint that exceeds the critical value will converge to the critical impedance required for optimal tracking.

One limitation of the frequency response fit impedance is that, while it is an accurate assessment of the target property, it isn't really reflective of how the impedance constraints were defined. The impedance-constrained systems perform well under a compliance metric test like that employed in Figure 50, but the admittance-constrained systems are less satisfactory for low maximum stiffnesses. Instead, stiffness observed from perturbation is a much closer relationship to the admittance formulation (though it is less reflective of dynamic interactions). In fact, as illustrated in Figure 51, when measured using this secondary stiffness metric, the admittance-constrained MPCs appear to more strictly respect the provided stiffness bound. Impedance control generally closely approaches its targets with either bound.

Clearly, the constraint formulation has a significant impact on the observed results. One observed trend was that the stiffness term dominates the impedance constraint – especially

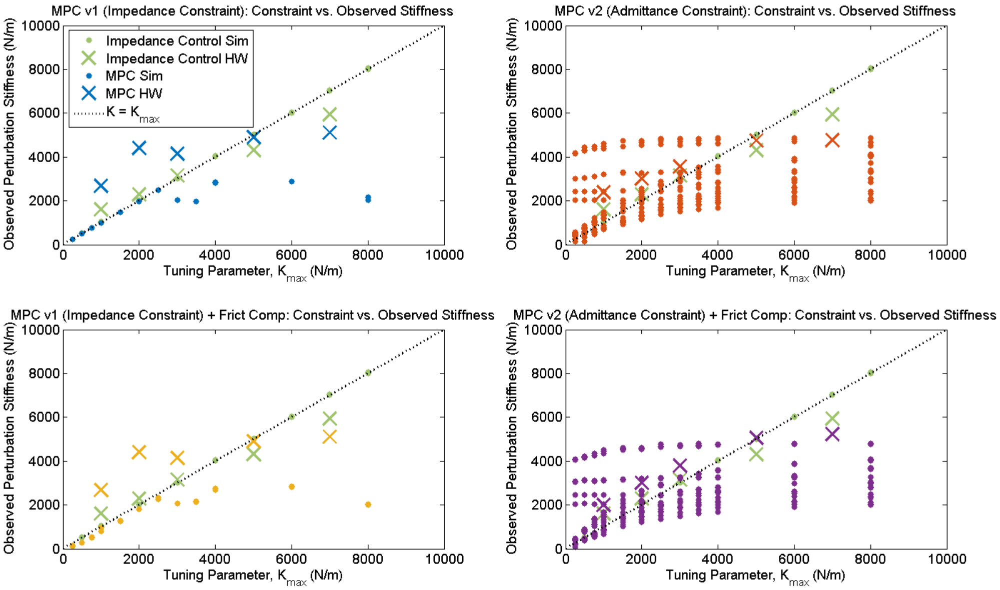

Figure 51: Impact of impedance constraint on MPC, and comparison to impedance controller with target stiffness. Stiffness judged by equivalent stiffness, found via perturbation.

within the operating bandwidth. As a result, it is difficult to set impedance targets that are more reflective of dynamic goals, such as a target damping parameter, without having them either negatively affect accuracy or stiffness at low frequencies. Part of this is due to the afore-mentioned fact that impedance is really a frequency-domain quantity, and the time-domain approximations used for constraint definition may replace actual target behavior with an implied internal trade-off of stiffness and damping. Thus, a particularly useful extension would be the creation of a custom solver that allowed nonlinear constraint definitions.

Overall, the MPC improves tracking for compliant, low-bandwidth systems relative to benchmark controllers, and provides a natural framework for compliant control by using impedance bounds rather than targets.

CHAPTER VI

COMPLIANT MPC FOR SYSTEMS WITH MULTIPLE ACTUATORS AND DEGREES OF FREEDOM

6.1 Overview

One of the aims of compliant control is practical implementation – ideally in a robotic system like the compact rescue robot (CRR). Realistically, this requires application to a system with multiple degrees of freedom (DoF) and non-trivial geometry and inertias. The additional complexity introduces several challenges, such as obtaining an acceptable system model, maintaining robustness to interaction between links, and avoiding computational load limits.

This chapter will discuss the implementation steps and qualitative performance of a two degree-of-freedom planar pneumatic arm controlled with MPC. First, the mechanical system will be described, labeled, and kinematics and dynamics will be derived. Next, an overview of the MPC design and alternatives for the multi-DoF cases is presented. Finally, the performance of the MPC-controlled pneumatic arm is discussed.

6.2 Planar Arm Geometry & Motion

The CRR has two 3-DoF arms. To avoid unnecessary complexity, only the bottom two joints, which constitute planar motion, will be assessed in this chapter.

6.2.1 Notation and Labeling

An illustration of the planar arm is shown in Figure 52, and then broken down into the top and bottom joint in Figures 53 and 54, respectively. The basic labeling scheme uses seven bodies and five body-fixed frames. The bodies are numbered, with each actuator split into two parts: part (a), the cylinder, and part (b), the piston. Since the pneumatic actuator cylinders and pistons operate along a common axis, they share a reference frame.

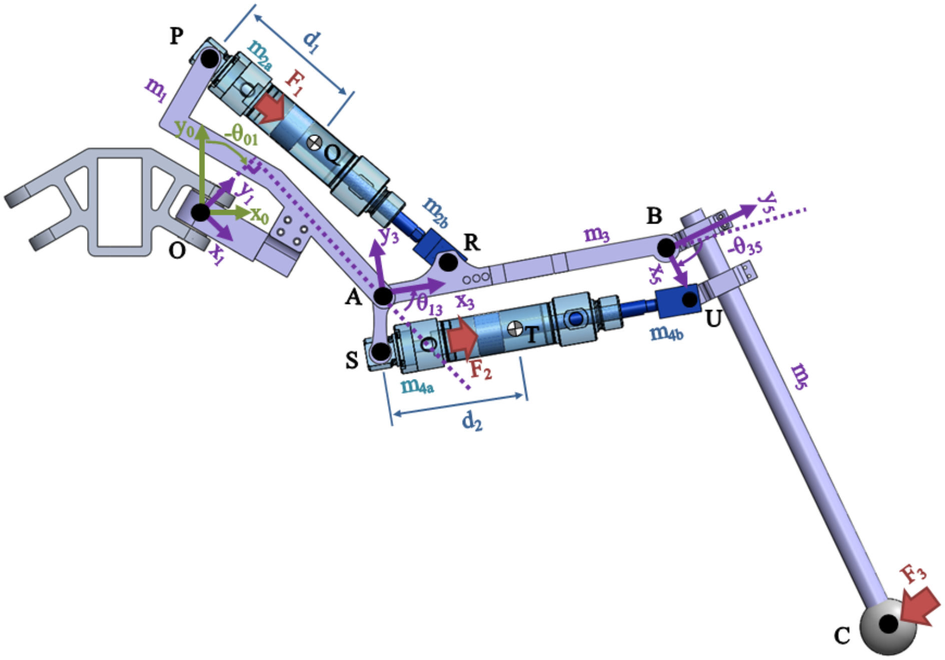

Figure 52: 2 DoF system overview.

Angles are denoted by Greek letters: θ is used to refer to coordinates, and corresponds to the link rotations, and ϕ represents the rotation of the pneumatic actuator. These angles are listed in the form θ_{ij}, referring to the angle from frame i to frame j. For angles used in derivations: α is used in derivations to refer to variable angles, and γ refers to constant angles with known values. Angles in derivations are specific to the local geometry and include a corresponding subscript.

Points are labeled with capital letters, and distances are specified with l_{IJ}, referring to the length from point I to point J. For clarity, since the distances l_{PQ} and l_{ST} vary, they are instead referred to by the variable distances d_1 and d_2. There are two input forces, F_1 and F_2, which act at points Q and T, and potential for an end effector disturbance force F_3, at point V.

6.2.2 Kinematics

In the CRR implementation, the goal is typically to place the end effector in a particular location in the X-Y plane. To do so, the forward and inverse kinematics of the leg are used

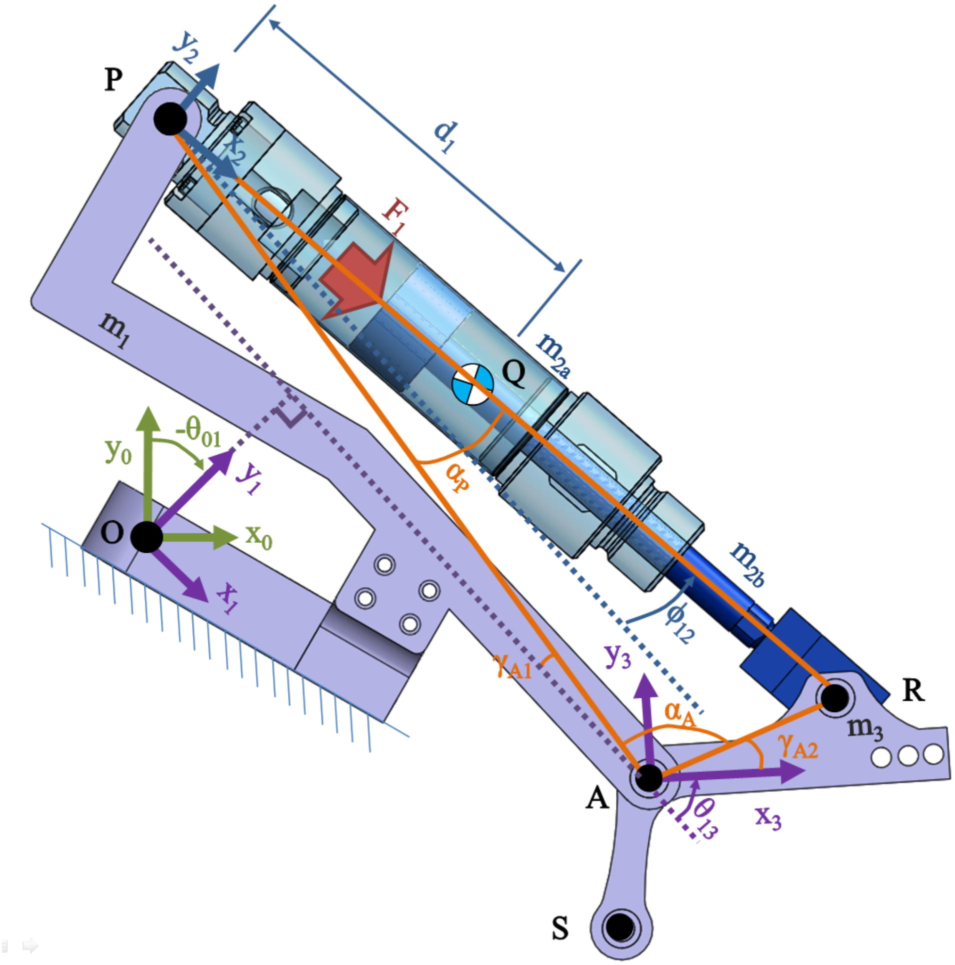

Figure 53: Geometric view to relate θ_{13} and d_1.

to relate joint angles to end effector positions and vice versa. These are shown in detail in the appendix in section D.1. However, they also require clearly established relationships between the linear motion of the prismatic actuators, and the angles of rotation of the joints.

In order to effectively use prismatic actuators, the actuators must pivot within the joint during piston motion, which leads to some geometric complexities. However, using trigonometric relations, the piston motion is easily related to angles of rotation between the links. The notation follows that used in Figures 53 and 54:

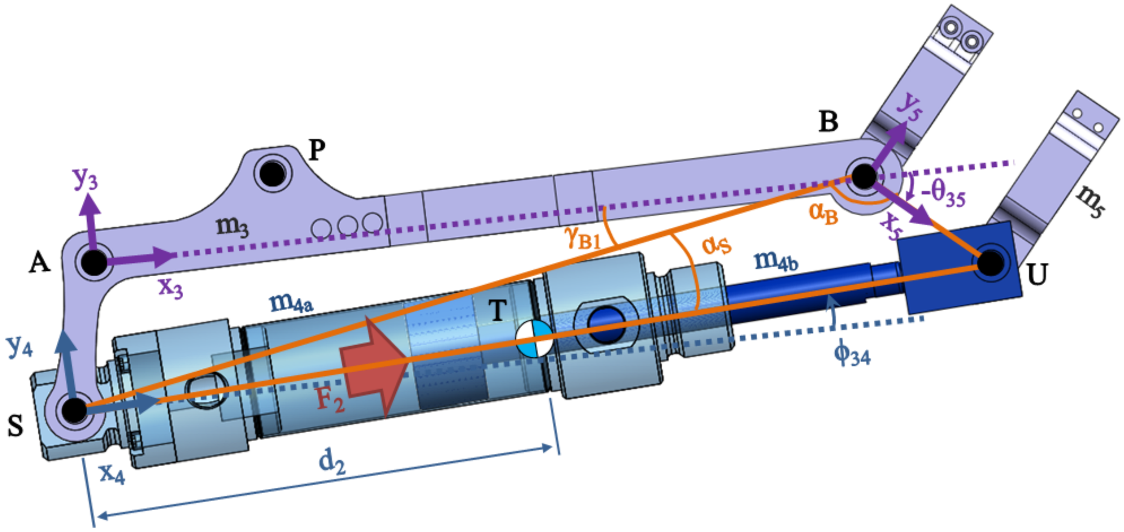

Figure 54: Geometric view of the middle link, relating θ_{35} and d_2.

$$d_1 = \sqrt{l_{PA}^2 + l_{AR}^2 + 2l_{PA}l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)} - l_{QR} \tag{97}$$

$$d_2 = \sqrt{l_{SB}^2 + l_{BU}^2 + 2l_{SB}l_{BU}\cos\left(\theta_{35} - \gamma_B\right)} - l_{TU} \tag{98}$$

where γ_{A1}, γ_{A2}, and γ_B are known angles, as described in the appendix. Conversely, joint angles are defined as functions of actuator position:

$$\theta_{13} = \pi - (\gamma_{A1} + \gamma_{A2}) - \cos^{-1}\left(\frac{l_{PA}^2 + l_{AR}^2 - (l_{QR} + d_1)^2}{2l_{PA}l_{AR}}\right) \tag{99}$$

$$\theta_{35} = -\pi + \gamma_B + \cos^{-1}\left(\frac{l_{SB}^2 + l_{BU}^2 - (d_2 + l_{TU})^2}{2l_{SB}l_{BU}}\right) \tag{100}$$

Arm dynamics were derived as well, as described in section D.3.

6.2.3 Dynamics Models

Planar arm dynamics were calculated using an analytical mechanics approach executed through MATLAB, as described in the Appendix in section D.3. The primary aim was to produce a set of dynamic equations that could be used to benchmark simplified model approaches. The results highlight part of the reasoning for limiting the multi-DoF analysis to two joints: because of the robot's complexity, a full dynamic model is already quite complex, even with only two actuators.

To check the validity of the derived arm dynamics, results were compared to a SimMechanics model that made use of SolidWorks body properties, using both time and frequency domain comparisons. These results were then compared to a simpler model based on a double pendulum approximation. The simple model treated each link as a fixed mass of set inertia (i.e. $\dot{\phi}_{12} = 0$ and $\dot{\phi}_{34} = 0$), measured position in terms of the angles, θ_{13} and θ_{35}, and used simple force-to-torque conversion equations:

$$\tau_{13} = -l_{AR} sin(\theta_{13}) \tag{101}$$

$$\tau_{35} = -l_{BU} sin(\theta_{35}) \tag{102}$$

$$\tag{103}$$

Comparisons to the detail model in the time and frequency domain showed that this model and simplified force-torque relations closely approximated system behavior,and would likely be sufficient for use in control.

6.3 MPC Implementation

As seen in chapter 5, the primary asset of the MPC-based control is that it enforces an impedance bound, such that accurate tracking is achieved at the minimum required impedance. Similarly, a lower impedance bound enforces the limit and achieves the tracking given the allowable impedance limit. To preserve these results in the multi-DoF implementation, an architecture must be used that enables this relationship to remain enforceable as time changes.

For example, one logical approach is to use a single MPC that contains all the state dynamics, linearized at each time instant. However, the end effector impedance constraint will be a distinctly nonlinear function of the states, so its linearization is unlikely to enforce any meaningful state relation, especially if the system geometry is rapidly changing. One alternative would be to change the optimization problem into a tracking problem with clear admittance constraints by using two low-level position controlled cylinders, with position provided in θ_{13} and θ_{35}. Then the optimization problem could be used as a reference trajectory generator, aimed to minimized tracking error while keeping errors below a time-varying but specified admittance relation. The use of position control at each actuator would enable the use of a state model that doesn't require explicit relations between position states and forces, provided the force controllers are robust to external disturbances.

Instead, the approach used here is to preserve as much of the 1-DoF architecture as possible and eliminate nonlinearities with mappings rather than linearizations wherever possible. Each joint is outfitted with a predictive controller that minimizes tracking error, subject to an impedance constraint. Instead of calculating dynamics linearly along the piston motion, they are calculated in angular position θ with an input torque, τ, and the conversions specified in equations (101) and (102) are used to produce force references. Since the joint is modeled as a rigid joint, the simplified, linearized actuator dynamics are dominated by an inertial term and a stiffness term due to gravity. Using the linearization of the simplified actuator dynamics as a guide, constant values may be selected to define each actuator dynamics as one with parameters $\ddot{\theta} = (K(g)\theta + \tau)/I$.

6.3.1 Joint-Level Implementation

The MPC configuration at each joint follows from a reconfiguration of the v1/impedance constraint version introduced in section 5.3.3:

$$min\ J = min \sum_{i=T_0}^{T_f} e_{act,i}^T Q e_{act,i} + \tau_{d,i}^T R \tau_{d,i} dt$$

$$\begin{cases} \dot{x} = Ax + Bu \\ x \leq b_{constr} \quad u \leq u_{constr} \\ e = [e_0\, e_I], \quad e = \theta_{ref} - \theta_1, \quad \dot{e}_I = e_0 \\ ||\tau_d|| \leq ||Z_{max} e||_{t=0} \\ \qquad = ||K_{max} e_{t=0} + B_{max} \dot{e}_{t=0} + M_{max} \ddot{e}_{t=0}|| \end{cases} \tag{104}$$

with state dynamics

$$\underbrace{\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\tau}_1 \\ \dot{\tau}_2 \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ K(g,x) & 0 & \frac{1}{I(x)} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \\ \tau_1 \\ \tau_2 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_n^2 \end{bmatrix}}_{B} \underbrace{\tau_d}_{u} + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{I}F_{frict} + f_{e1}(x_{jt2}) \\ f_{e2}(x_{jt2}) \\ 0 \end{bmatrix}}_{D} \tag{105}$$

Here, f_{e1} and f_{e2} are disturbance terms caused by the dynamics from the other joint, x_{jt2}, i.e. x_{13} for the bottom joint, and x_{35} for the top joint. The MPC model used was simplified by ignoring disturbance terms, with acceptable results. Nonetheless, this possible extension will be discussed in section 7.3.2.

In equation (104), θ_1 is the angular position, and θ_2 is the angular velocity. Since there are two joints, the implementation described in equation (104) is applied twice, once to (θ_{13}, τ_{13}), and once to (θ_{35}, τ_{35}). It would also be possible to linearize the simplified model dynamics (equations (221) and (222)) at each operating instant.

6.3.2 Performance and Impedance Constraint Conversions

Because the joint dynamics for the multi-DoF implementation have been converted to to angular dynamics, it is necessary to also convert performance and stiffness constraints. Maximum forces are easily converted using the relation introduced in equations (101) and (102):

$$(\tau_{13})_{max} = l_{AR}\sin(\theta_{13})F_{1,max} \tag{106}$$

$$(\tau_{35})_{max} = l_{BU}\sin(\theta_{35})F_{2,max} \tag{107}$$

Of course, since these equations are dependent on the angle, the limits are actually changing. In practice, it was found that setting too high a limit could be actively detrimental, the average (1/2 the maximum value) was used for the controller on θ_{35}, and the full value was used for θ_{13}.

The rates are slightly more complicated, since the angular position changes with time:

$$\left(\frac{d\tau_{13}}{dt}\right)_{max} = \left(l_{AR}\sin(\theta_{13})\frac{dF_1}{dt} + l_{AB}\dot{\theta}_{13}\cos(\theta_{13})F_1\right)_{max} \tag{108}$$

$$\left(\frac{d\tau_{35}}{dt}\right)_{max} = \left(l_{BU}\sin(\theta_{35})\frac{dF_1}{dt} + l_{BU}\dot{\theta}_{35}\cos(\theta_{35})F_2\right)_{max} \tag{109}$$

Again, it would be possible to update the constraint in real time, but given the poor results observed with such an approach (as noted in section 5.3.2), a constant value was instead used that ignored the angular-velocity dependent term and simply assumed the maximum value from the sinusoidal term:

$$\left(\frac{d\tau_{13}}{dt}\right)_{max} \equiv l_{AR}(\frac{dF_1}{dt})_{max} \tag{110}$$

$$\left(\frac{d\tau_{35}}{dt}\right)_{max} \equiv l_{BU}(\frac{dF_2}{dt})_{max} \tag{111}$$

The minimum bounds are simply the negative maximum values.

6.4 Performance

Tracking performance was compared with several tests, focusing on tracking accuracy, response to unexpected disturbance, interaction between the joints, and the effect of changing actuator stiffness properties.

6.4.1 Tracking

With the planar arm, most robotic tasks consist of tracking motions, e.g. a line in space, which requires simultaneous, continuous motion at differing amplitudes. As a sample, two sinusoidal motions were provided to the actuators so that they would span a greater range of inertias. Results are shown in Figure 55, and are plotted in degrees. As the figure shows, while tracking is not smooth, error minimization is quite good; for reference, the linear root-mean-square error values (error as measured by comparing piston position to position reference) can be compared to results on the significantly less complex 1-DoF model. From the hardware results in section 5.5, or from the raw data collection in the appendix in section A.4, the best tracking had RMS errors on the order of 1-2 mm, whereas the tracking seen in the figure on the multi-DoF model is similarly under 2 mm.

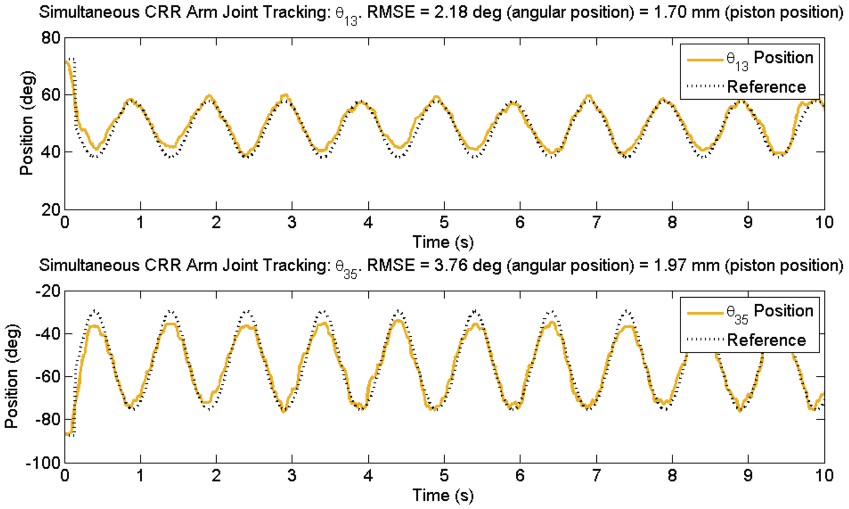

Figure 55: Simultaneous tracking θ_{13} and θ_{35} using MPC with high impedance constraints.

These results also point to MPC's inherent robustness to model error: despite the fact that the models used for control made significant simplifying assumptions including linear time-invariant dynamics, the control still functions fairly well. However, it also exposes a flaw – while the RMS error minimization is good, tracking is not as smooth as desired. A penalty on velocity error may be used to reduce this effect, but in general, this induces a trade-off that can be difficult to tune.

Next, the compliance of the controller will be tested by subjecting it to known but unexpected external loads.

6.4.2 Interaction & Disturbance Response

An important system quality is that it responds well to unexpected interaction: the system should be backdriveable without applying significant force, and once the external load is removed, it should return to tracking without significant overshoot or oscillation.

Obviously, the location of the disturbance matters as well. To test the value of the MPC to general safety, a spring scale was used to perturb the system by 9 N, and the associated disturbances were measured, as seen in Figures 56 and 57. First, the external load was applied to the middle link, by hooking onto m_3, as labeled in Figure 54.

The perturbations are annotated in Figure 56, and correspond to peak disturbances of

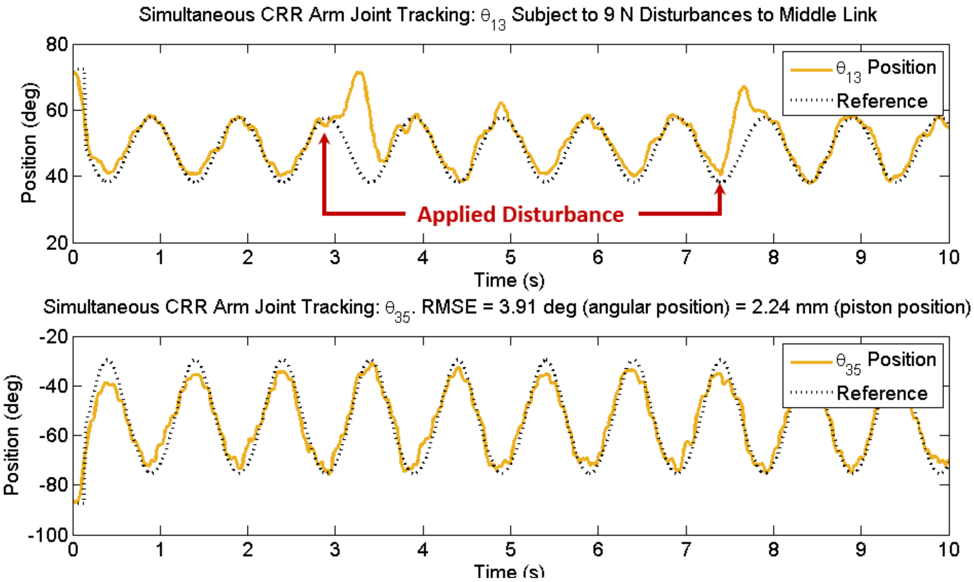

Figure 56: Simultaneous tracking θ_{13} and θ_{35} while subjecting middle link to 9 N disturbances (annotated).

20-30 degrees, or 30 - 50% of that joint's over 61-degree range. In other words, even with a stiffness bound capped only by the force limit (10 Nm, acting over a 5-50 mm moment arm), the system proved very backdriveable and with an impact force 1/5th that of the dangerous bound suggested in 2.2.4.

Figure 57 is a similar test, but the 9 N perturbation is instead applied to the bottom link, m_5. Interestingly, while the system is similarly compliant, most of the compliance is observed in the first actuator, θ_{13}, though this may be partly due to the system geometry.

One important question is whether it is possible to render the system defunct through impact. Figure 58 shows the system subject to large impacts to the bottom link (executed by repeatedly and rapidly hitting m_5 by hand). While the system responds well to disturbance and quickly returns to good tracking, the top link, which has significantly more inertia and therefore experiences greater joint stiffness due to gravity, occasionally overshoots on the return. Additionally, while no tests were observed in which an impact destabilized the system, it is evident that a large enough disturbance will make the controller on θ_{13} useless. As seen in the figure, disturbances that drive the top joint too close to the minimum moment arm exceed the controller's force capability by effectively placing the robot in a singular

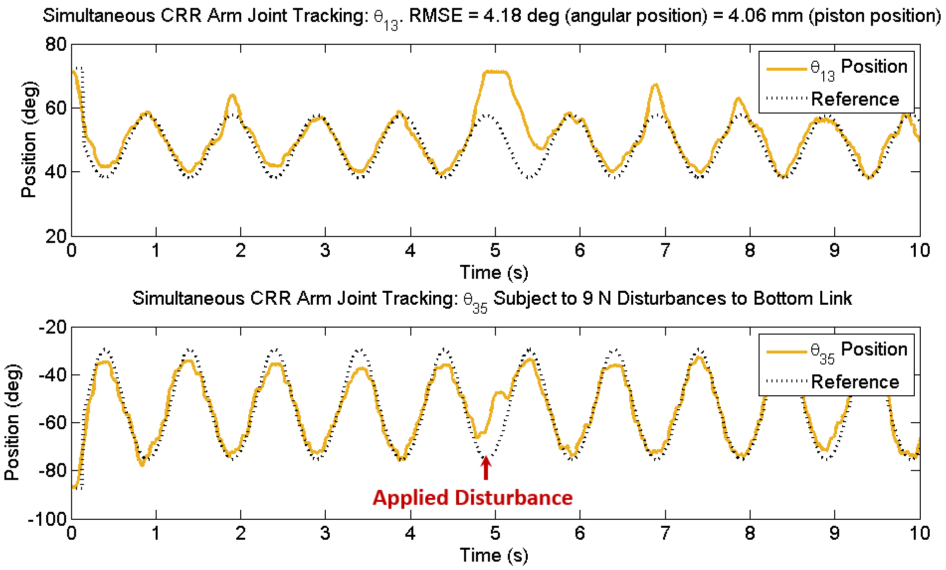

Figure 57: Simultaneous tracking θ_{13} and θ_{35} while subjecting bottom link to 9 N disturbances (annotated).

position. However, this generally requires a prolonged interaction – as seen in Figure 59, θ_{13} can be tracked over most of its 61-degree range (11 - 72 degrees) and subjected to disturbances, without entering into the singular position. Obviously, this problem could also be avoided through mechanical design changes.

Another interesting question is how MPC interacts with other controllers. For example, Figure 60 shows the performance of the planar arm, with θ_{13} controlled using MPC, and θ_{35} controlled using a PID controller that has been intentionally tuned to be comparatively stiff. As a result, almost all of the compliance is deflected to the MPC-controlled joint. The perturbations in the figure were achieved by pushing the robot (by hand) and were prolonged (the system returned to tracking as soon as it was released), which further emphasizes the result that the MPC will be backdriveable as necessary. A multi-link system could be smartly designed to apply compliant joints only where necessary, as in the DM^2 approach [101] to leverage this effect.

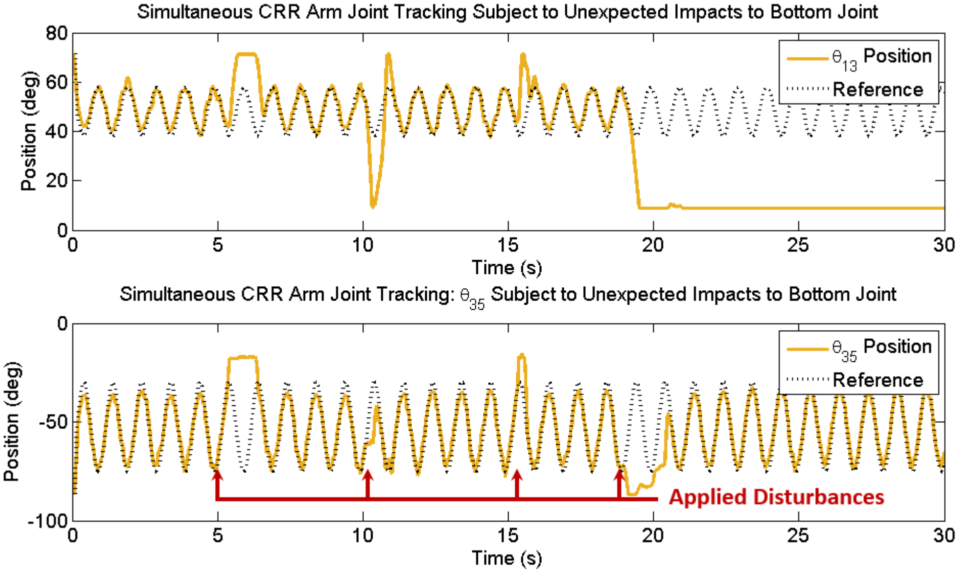

Figure 58: Simultaneous tracking θ_{13} and θ_{35} while subjecting bottom link to large impacts.

6.4.3 Effect and Performance of Impedance Constraints

The previous sections have shown that MPC carries over well to the multi-DoF system, is compliant to interaction, and safe for operation near a user. However, part of this is due to the fact that even the high impedance achieved with MPC is fairly compliant on the provided system, as the controllers above all use an unbounded stiffness and instead just apply a force limit. A further question is whether the use of impedance constraints translate to the multi-DoF case.

Figure 61 shows how joint tracking is affected as the stiffness bound on the MPC controlling θ_{35} is reduced. The annotations in the figure show the period over which a particular angular joint stiffness – 15, 10, and 5 Nm/rad – is enforced. It is clear that reducing the stiffness does gradually impact performance, especially at the peaks where friction has a more significant impact. Interestingly, the changing stiffness in θ_{35} control has a negligible effect on θ_{13} control, which continues without effect.

Similarly, impedance bounds may be used to affect θ_{13}, as seen in Figure 62. In the figure, the stiffness constraint on the top joint is reduced to 50 Nm/rad, which increases position RMS error. The second joint, θ_{35}, is first held constant (i.e., it's inertia is unchanged), and

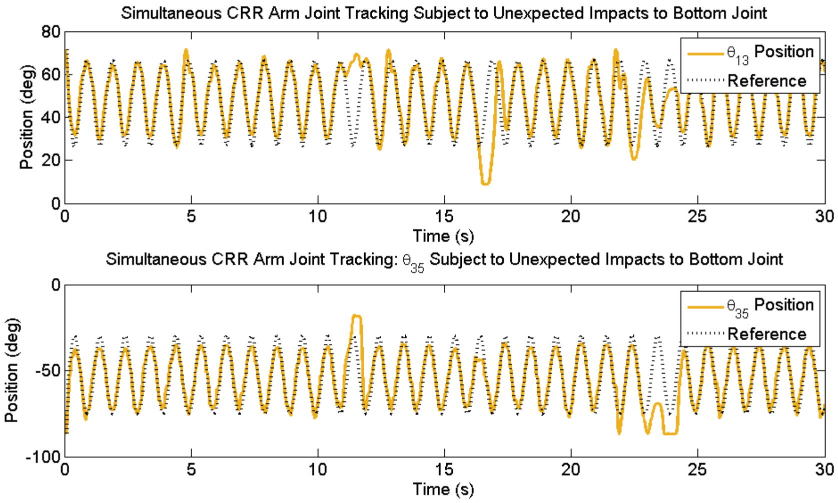

Figure 59: Simultaneous tracking θ_{13} and θ_{35}. Disturbance forces are applied to m_5 at T = 10, 15, and 22 seconds.

then suddenly moved to follow a sinusoidal pattern. This change has a negligible effect on control of the first joint, despite its decreased stiffness and corresponding increased tracking error.

6.4.4 Performance Observations

As shown in the last few sections, even a fairly straightforward multi-DoF implementation with multiple MPCs and simplified dynamic models is quite effective in achieving the goal of compliant control – accurate control and safe operating circumstances. Of course, there are clear concerns to be addressed.

First, the computational load is not insignificant. For this controller, at 100 Hz, the computational load could be measured using Task Execution Time, or TeT – a measure of how long it takes Real Time Workshop (RTW) / xPC Target to compute at each time step For a prototypical system operating a single MPC, an example average TeT was 0.0015 s with a max TeT of 0.003 , while with two MPCs, the average and maximum TeTs increased to 0.0043 s and 0.006 s, respectively. For the modest target used with this hardware, featuring 1 GB RAM and 1.4 GHz speed, this means that operation would likely be capped at 3 joints, given the current system configuration. Of course, the PC104 is a nearly 8

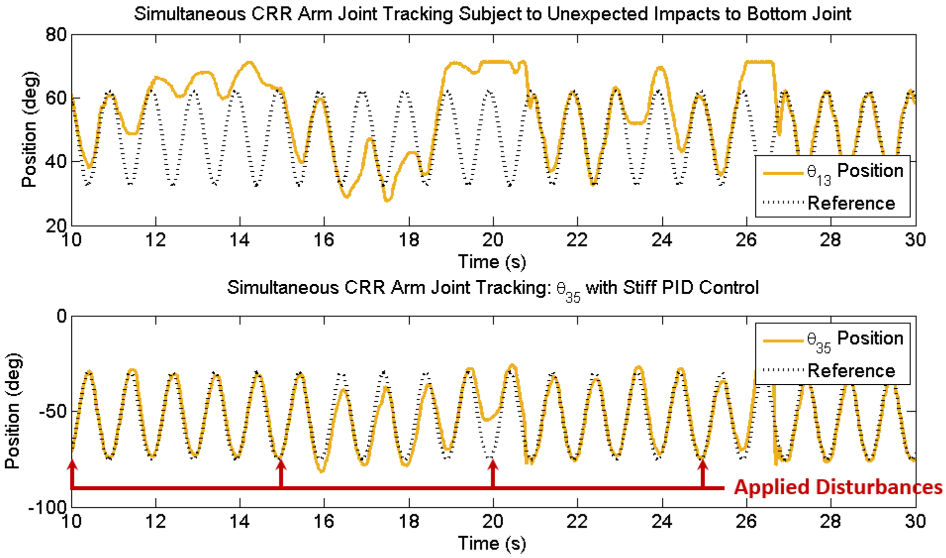

Figure 60: Tracking θ_{13} with MPC and θ_{35} with a stiff PID controller while subjecting bottom link unexpected disturbances.

year old computer, and the primary limitation on xPC Target / RTW execution speed is memory, so it could easily be upgraded to be more capable.

Computational concerns will also likely be alleviated as more fast solvers are produced; CVXgen is one of several similar solvers [29, 45] that are becoming ever more capable.

It was seen in the multi-DoF case that the biggest tracking errors happen when direction changes, i.e. when stiction is the most prevalent. Elsewhere, tracking proved to be fairly unaffected by friction, though it's possible that the non-smooth results are actually benefiting the MPC and act as a sort of dither signal. Results could be improved by adding friction compensation, and improving the model to more accurately reflect the changing inertial and gravitation dynamics, and the effects due to other joints.

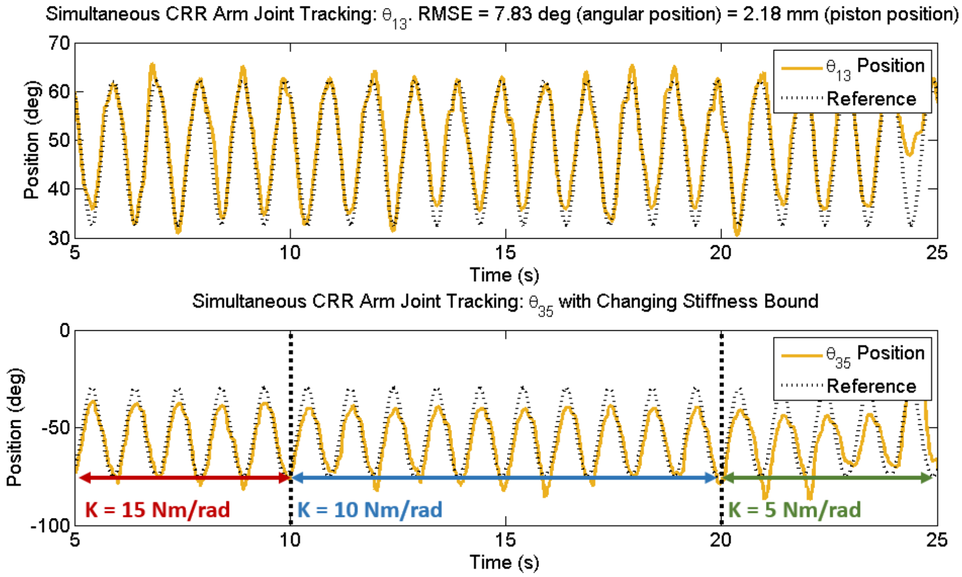

Figure 61: Simultaneous tracking θ_{13} and θ_{35} while changing the stiffness bound on the MPC controlling θ_{35}.

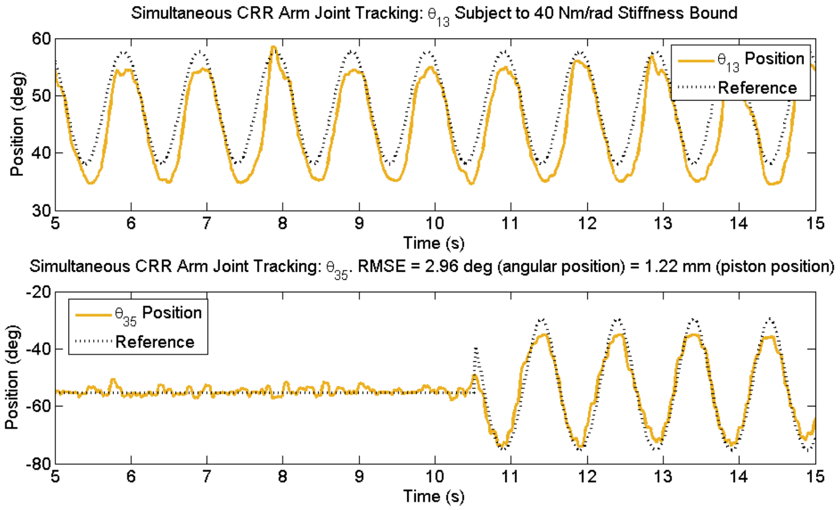

Figure 62: Effect of significant changes in the motion of θ_{35} while θ_{13} is tracking and subjected to a lower stiffness bound of 40 Nm/rad.

CHAPTER VII

CONCLUSIONS & FUTURE WORK

This thesis presents a new approach to compliant tracking, in which a constrained model predictive controller (MPC) enforces optimal tracking control while satisfying impedance bounds.

7.1 Summary of Contributions

The thesis has six primary contributions:

1. A complete analysis of the needs, standards, and design specifications (in the form of well-defined performance tests and associated metrics) for compliant pneumatic control, based on a literature review of applications and comparable tasks, and performance of a set of benchmark controllers that span the range of possible target performance characteristics.

2. A high-fidelity actuator model validated against several hardware platforms in the time and frequency domains.

3. Development and analysis of a novel predictive friction compensation scheme that improves tracking for slow systems, and comparison to standard friction compensation approaches.

4. Two approaches to MPC for compliant tracking that enforce impedance/admittance constraints in simulation and in hardware and meet the design targets laid out in contribution (1): they have bandwidth within the desired 2-5 Hz range, achieve tracking errors comparable to the best benchmark controllers at a given stiffness bound, and provided with an arbitrarily high bound, achieve the best accuracy possible with MPC at the lowest necessary impedance without any a-priori knowledge of what that impedance will be.

5. Extensions to the MPC architecture that provide theory-backed stability and feasibility guarantees while preserving compliance constraints.

6. Predictive control for a multi-DoF system consisting of a planar robotic arm, demonstrating that the benefits observed in the single-DoF case extend to the planar case, including compliant behavior, enforceable stiffness, and backdriveable joints for safe interaction.

Through an extensive literature review, it was shown that compliant tracking is a common aim among systems that aim to imitate natural motion, or that are operated on or around humans. Pneumatic actuators are well-suited for these applications: tracking bandwidths are generally low, up to 2-5 Hz, and inherent compliance is desirable. Additionally, in applications such as the Compact Rescue Robot, the inspiration for this work, a versatile actuator is necessitated that will be able to switch easily between compliant tracking in free space, and high-force control for walking and lifting tasks. The literature review also helped to establish quantitative performance goals by surveying related research, and established a precedent for a constrained optimal control approach via the *safe brachistochrone* problem: like the time-optimal variant, the goal of compliant tracking is to maintain safety while moving accurately; therefore, compliance should be asserted as a bound on a position-error-minimization problem.

To better establish baseline performance, several performance tests were designed to related qualitative control goals to quantitative design specifications. These tests were applied to a set of four benchmark controllers selected to span the accuracy-compliance trade-off in simulation and hardware to more accurately characterize the state-of-the-art on the trade-offs of accuracy, compliance, bandwidth, and response to unexpected disturbance. Tests were run on hardware and on a high-fidelity actuator model, developed in Simulink and validated against the single DoF hardware platform. Results showed that while all controllers are inevitably subject to a trade-off of compliance and accuracy (and bandwidth), there are clear asymptotes at for which tracking performance does not significantly improve.

Two MPC controllers were developed for the single DoF application – version 1 used an impedance constraint, while version 2 used an admittance constraint. The controllers were compared to benchmark performance and shown to both match the accuracy-stiffness tradeoff observed among benchmarks, and achieve tracking comparable to the best benchmark result at a given stiffness constraint. Furthermore, even at peak performance, the impedance observed by constrained MPC-controlled systems reached a steady-state value equivalent to the minimum impedance shown by the benchmark analysis to provide the best accuracy, without any a-priori knowledge of this value. By contrast, an impedance controller design to exceed this target impedance would obtain the same tracking, but with significantly less compliance.

In addition to the basic MPC formulation, a version using a dual-mode prediction formulation was designed that offered theory-backed stability and feasibility guarantees. However, tests in simulation and hardware showed that the addition of the terminal cost and constraint required for dual mode satisfaction provided little benefit to this system; instead, it simply increased computational load without significantly improving results. Furthermore, using the simulation model, extensive parameter variation showed the basic MPC to already be stable for a variety of reference and disturbance possibilities, due to stable low-level dynamics, constraints that ensured feasible reference inputs, and a sufficiently long prediction horizon. While many of these results are particular to this system, they do verify that a dual-mode prediction approach can be effectively used in accordance with compliance constraints, as long as other state/input constraints don't conflict. Furthermore, the process may be avoided through careful selection of key parameters such as the horizon length, or by ensuring that the low-level dynamics are stable and merely ensuring appropriate constraints for feasibility.

The MPC approach was also used to help identify limiting bounds in controller performance for compliant control. While the impedance and admittance constraint methods provided similar results in hardware, the admittance constraint marginally reduced the minimum compliance required for a given accuracy in simulation, suggesting that the benchmark controllers hadn't fully approached the capabilities of model-based feedback control. Additionally, using friction compensation, both MPCs were able to significantly improve tracking, even at low compliance metrics, with negligible effects on measured system impedance, demonstrating that friction remains a large obstacle in pneumatic control.

In addition to the additive friction compensation method used in the thesis, a novel predictive friction compensation method was introduced that sequenced a predictive estimator and controller so that the system could better prepare for sudden disturbances, such as stiction. The approach was shown to improve friction compensation for slow systems, both analytically and in simulation. Unfortunately, its value didn't extend to compliant control, presumably either due to conflicts with compliance constraints, or because a system already operating at optimal or near-optimal performance can't change behavior within the framework of MPC.

Finally, the results observed on the 1-DoF application were extended to a 2-DoF planar robotic arm. It was shown that, the MPC approach translated to a implementation on a multi-DoF system by using individual predictive controllers at each joint. Mappings were used in lieu of linearization where possible, and the controller proved to be effective despite significant assumptions, including use of a linear time-invariant model and no specific consideration for disturbances caused by other actuators. In fact, with maximum impedance limits, tracking errors under 2 mm were observed, comparable to the best single DoF RMSE, which was on the order of 1-2 mm. Additionally, the system was backdriveable without incurring significant force, stayed well under the 50 N interaction upper bound suggested in the literature review, and preserved stiffness constraints applied to the controller, though they were generally deemed unnecessary: the default observed impedance using MPC was already minimal, and ensured the desired target behavior of safe, compliant, and accurate tracking.

7.2 Applications & Limitations

While the predictive controller is better formulated to achieve low-error, low stiffness control than many state-of-the-art benchmarks, its main asset is the ability to treat impedance as an inequality constraint rather than a target (observed via force interaction). This result is clearly illustrated in Figure 50 and detailed in section 5.5. In applications such as orthoses and cooperative robots, in which humans and machines are in close proximity, the safety of their interaction is ensured by bounds on interaction forces, which are directly related to contact impedances. The MPC enables a designer to determine what that critical level is and use the predictive controller to ensure compliant control that is accurate without violating the safety bound. The format is in keeping with safety standards, which are frequently given as upper bounds on force or velocity [1,41]. Other applications include robots for pick-and-place manufacturing, or space and exploration machines, which require manipulators that can freely extend and contact the environment without risking harm or damage, i.e. machines that are subject to some impedance bound to guarantee mechanical robustness.

Of course, there are clear limitations to the application of compliant MPC. First, the system model must be well-defined and any loads must be easily sensed or negligible with respect to inherent machine dynamics and actuation forces. More importantly, while compliant control effectively limits output impedance, it is also an inherently compliant controller – meaning that it is ineffective for applications that require significant un-modeled force, e.g. for walking or lifting heavy loads. For example, in the true case of a rescue robot that both roams about and interacts with endangered victims, the compliance-constrained MPC would only be used during interaction, and instead would be replaced by more suitable controllers in the walking case, which requires sufficiently stiff outputs.

The scope of this work has been limited to pneumatic actuators and sets of single DoF applications, but the resultant predictive controller and constraints for compliant tracking are not limited to these sample cases. In fact, pneumatic systems are likely a particularly challenging application: pneumatic systems introduce nonlinear force dynamics that are inherently coupled to other system states, which makes any constraints that relate force to other states difficult to enforce. Instead, the application of compliance and impedance constraints to systems with inherent compliance should be more straightforward in actuators and systems that are with force dynamics that are better behaved. This presents a clear direction for future research.

7.3 Recommendations for Future Work

There are two clear paths for future work: first, the predictive controller capability could be improved for the pneumatic application, and second, as discussed in section 7.2, the breadth of MPC for compliant tracking could be explored by applying the controller & constraints to 'nicer' systems.

The application to pneumatic systems could be improved through better friction compensation and direct stiffness control via access to pressure states, while the overall predictive compensation scheme would benefit from improved solvers, multi-DoF realizations that are better suited to the strengths of predictive control, and expansion to actuators with more well-behaved internal dynamics.

7.3.1 Improvements to MPC for Pneumatic Applications

While predictive compensation was shown in section 5.4.1.3 to greatly improve friction compensation for slow systems (and especially pneumatic actuators) and produce tracking errors on the order of 1 mm, these results did not translate to compliant control. A likely reason is that the effects of the predictive compensator were only translated in the disturbance term, and that the added friction compensation effectively conflicted with impedance constraints. One proposed solution would be to attempt to offset the friction needs in the impedance constraint. Since the impedance and admittance constraints are frequently applied as a function of force, the estimated friction force over the prediction horizon could be added to the force incurred by the impedance relation, or otherwise combined with the impedance constraint. Since friction compensation is inherently canceled by system friction, the capability would have negligible effect on the output impedance. Of course, in contact situations, this approach would simply increase the force limit, necessitating some kind of disturbance observer.

Additionally, while MPC with compliance constraints was shown to be effective on pneumatic actuators, the challenge of nonlinear force dynamics was countered by controlling stiffness dynamics indirectly, via closed loop force control, rather than directly, using chamber pressure relations. This could be improved using a linear model that enabled direct access to the pressure states. While it is easy to linearize pneumatic dynamics based on the equations, most past researchers have avoided this approach and instead used either an inner force control loop or an aggressive position control method, such as sliding mode control. However, a model-based approach to full state-feedback pneumatic position control suggests that the inner loop may be unnecessary if appropriate modeling assumptions are made and the system is well-conditioned.

In section 3.4.3, it was shown that the mass flow dynamics could be easily modeled as a function of the chamber pressure and the input orifice area. Furthermore, this relationship is easily modeled with a polynomial surface defined by equation equation (19). The complete dynamics are then presented compactly as

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ P_a \\ P_b \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{P_a A_a - P_b A_b}{m} \\ \left(P_a A_a x_2 + \dot{m}(A(u), P_a) RT\right) \frac{1}{x_1 A_a} \\ \left(P_b A_b x_2 + \dot{m}(A(u), P_b) RT\right) \frac{1}{(L - x_1) A_a} \end{bmatrix} \tag{112}$$

where mass flow is approximated by local linear functions $\dot{m}(A(u), P_{a,b}) = K_1 A(u) + K_2 P_{a,b}$. K_1 and K_2 are constants that are easily obtained from equation (19).

While friction force is generally either omitted or found using a Stribeck-curve, it is also possible to approximate it, e.g. with a viscous friction model, $F_{frict} = bx_2$. A linearization can then be found by determining the Jacobian, J (using a Taylor Series approach):

$$\Delta\dot{X} = J\Delta X \quad \text{or} \quad \dot{X} = J\left(X - X|_{t=T_0}\right) + \dot{X}|_{t=T_0} \quad \text{where} \quad J =$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -b & A_a/M & -A_b/M & 0 \\ \frac{A_a\, P_a\, x_2 - R\, T\, \dot{m}_a(A_{eq}, P_a)}{A_a\, (L + ds_a + x_1)^2} & -\frac{P_a}{L + ds_a + x_1} & -\frac{A_a\, x_2 - R\, T\, \Phi_{2a}}{A_a\, (L + ds_a + x_1)} & 0 & \frac{R\, T\, \Phi_{3a}}{A_a\, (L + ds_a + x_1)} \\ \frac{A_b\, P_b\, x_2 + R\, T\, \dot{m}_b(A_{eq}, P_b)}{A_b\, (L + ds_b - x_1)^2} & \frac{P_b}{L + ds_b - x_1} & 0 & \frac{A_b\, x_2 + R\, T\, \Phi_{2b}}{A_b\, (L + ds_b - x_1)} & -\frac{R\, T\, \Phi_{3b}}{A_b\, (L + ds_b - x_1)} \end{bmatrix} \tag{113}$$

The functions Φ_{2a}, Φ_{2b}, Φ_{3a}, and Φ_{3b} result from the mass flow derivative, and are functions of chamber pressure and input. They are detailed in the appendix in section C.1.

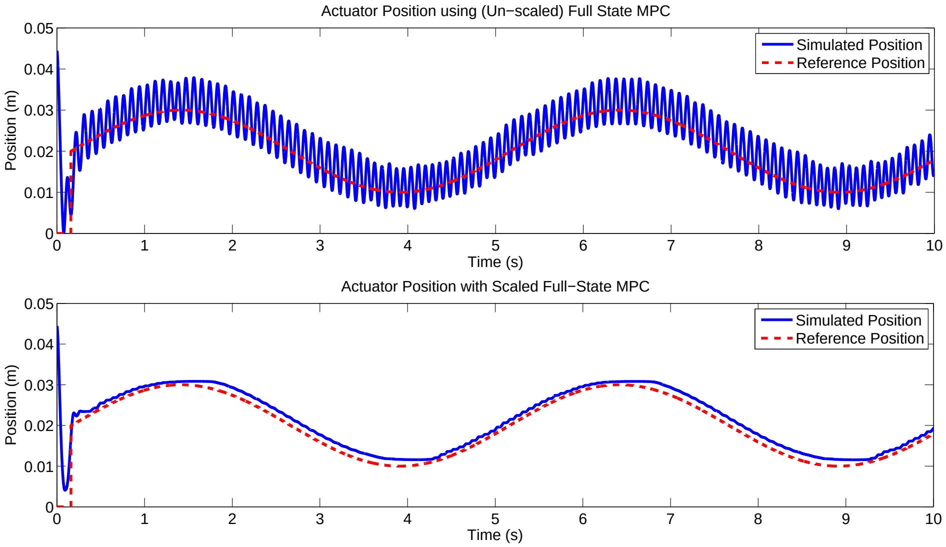

Figure 63: Pneumatic actuator position tracking using a model predictive controller equipped with a constant linearization of the full state dynamics. Top: direct linearization. Bottom: non-dimensionalized and scaled linearization.

However, this matrix is prone to singularities and poor conditioning. To avoid issues with condition number, a non-dimensionalization, detailed in the appendix in section C.1.1, may be used to define an appropriate change-of-variable. Results show that this non-dimensionalization is key, reducing condition numbers from values as high as 10^{15} to 300. Indeed, Figure 63 shows how a simulation of a model predictive controller (with simple force constraints – not impedance constraints) was markedly improved by using the well-conditioned linearization.

These good results suggest that a time-varying full-state model might be effective in MPC for compliant tracking. In both LTI and LTV cases, the full linearization would hopefully help relate compliance directly to the pressure states to more significantly improve the capability of the constrained compliant controller.

7.3.2 Changes to MPC Implementation, Formulation, and Scope

One of the clearest limitations on the MPC formulation is the choice of solver. CVXgen, while effective, imposes some rather severe limitations on the compliant control problem. If the optimization problem could instead be parsed for a nonlinear system, or with constraints that directly addressed impedance, the MPC could become much more powerful. With state-of-the-art solvers, CVXgen is an ideal choice for real-time implementation, and the most appropriate alternative would be an explicit MPC that computes solutions off-line and uses a lookup table to implement them in real-time.

Beyond solver limits, the particular problem formulation may be better defined for predictive control, especially when applied to a system with multiple degrees of freedom. In chapter 6, a planar robot was controlled using MPCs on each joint. The controllers used constant, simplified actuator models, but still recorded good performance. This approach could be improved upon in two ways: first, a higher-fidelity model with consideration of disturbances (as detailed in equation (105)) might be used to better relay system dynamics to the model-based controller, and second, the MPC could instead be used as a high-level controller, e.g. as a trajectory generation tool, subject to compliance constraints. The latter approach would be especially appropriate since MPC is excellent at enforcing constraints and solving for optimal trajectories, but the challenge of exact, actuator level tracking is better enforced by a low-level controller. This also reduces the stability problem to one of ensured feasibility.

Finally, it would be interesting to apply MPC for compliant tracking to systems that are better behaved than the single-valve/single-cylinder pneumatic actuator. Within the realm of pneumatic applications, a multi-input, multi-output (MIMO) system, such as a two-valve/single cylinder approach would allow simultaneous force and stiffness control [76], making it a type of variable stiffness actuator (VSA). Then the predictive controller could be used to more directly modulate the stiffness bounds without being heavily affected by the inherent coupling between actuator stiffness and piston position. Similarly, the predictive controller could be applied to other VSAs with simpler, more linear force dynamics, or to other actuators with better force dynamics, such as Series Elastic Actuators (SEAs).

The implementation of MPC with constraints on compliance is an effective strategy for creating control schemes that possess intrinsic safety, given an appropriate upper bound. The approach works well with pneumatic systems, and even improves the capability for general accurate, compliant control relative to common benchmark controllers. The controller has clear value to applications where machines may interact with humans and intrinsic safety is desired, or where a robot may move about an unknown environment and should be able to experience unexpected contact robustly and without damage. With further study, the approach could more directly affect system stiffness, and could applied to a broader set of compliant systems.

APPENDIX A

DATA COLLECTION & ANALYSIS

In chapters 4 and 5, the performance of a single degree of freedom platform was explored using numerous quantitative metrics. This section details the data collection process: methods and codes for non-obvious metrics, and an overview of the data collected.

The goal of the analysis was to represent individual controller performance via a concise set of metrics, so that over sufficient parameter variation, performance trends would be evident. Therefore, metrics included root-mean-square position and velocity tracking error, interaction force, response characteristics (overshoot, settling time, steady-state error), and frequency domain quantities, like closed loop bandwidth and system impedance.

The following sections detail the process as necessary. First, in section A.1, the spectral analysis approach is discussed. Next, specific codes for frequency-domain methods, data-fitting, and data-finding are detailed in section A.2. Section A.3 provides an overview of the hardware configuration connected to each platform (the platforms themselves are discussed in chapter 3), and possible concerns and differences. Finally, section A.4 provides a comprehensive overview of the observed data from simulation and hardware.

A.1 Spectral Analysis

In both hardware and simulation, it is necessary to use spectral analysis to examine the behavior of the system in the frequency domain. Fortunately, the available MATLAB tools are fairly powerful, provided results can be averaged– a straightforward process as long as the experiments are all performed with the same sampling frequency, `Ts = 1/dt` (or `dts` in the simulation case).

In general, spectral analysis looks at spectral density – the amount of power contained in a signal at a given frequency. A good high-level overview at `http://www.mathworks.com/help/signal/ug/spectral-analysis.html` that talks about the general process and value of various spectral analysis techniques.

The primary aim of the spectral analysis performed so far has been to find transfer functions for systems or subsystems. The transfer function is a frequency-domain representation of a system given and input, x, and an output, y. From the magnitude and phase plots, the goal is often to characterize system behavior, or to compare it between hardware and simulation, against an analytical estimate, or to make quantitative observations, such as the bandwidth.

The subsequent overview goes through the basic steps for obtaining a transfer function. Next, and overview of comparable MATLAB functions is given, as well as necessary procedures to improve their utility (primarily averaging).

A.1.1 Method for Obtaining a Transfer Function

The transfer function is actually the quotient of the cross power spectral density of x and y, P_{yx}, and the power spectral density of x, P_{xx}. Mathematically, the power spectral density is the discrete Fourier transform of the auto-correlation (i.e., it shows the frequency distribution of the *expected* power content in the signal). Similarly, the cross power spectral density is the DFT of the cross-correlation, so it measures the power shared by two signals at each frequency.

To find the transfer function, it is first necessary to remove the DC offset from the time-series data, which is done by subtracting the mean:

$$x(t) = x_{actual}(t) - mean(x_{actual}(t)) \tag{114}$$

Next, time series data is prepared for the DFT using windowing functions. More advanced algorithms, such as the Welch algorithm used by several MATLAB commands, break down the time content into multiple overlapping segments and apply windowing functions to each of them. While there are numerous windowing functions, they generally weight the content of the signal to improve accuracy of the DFT; for example, they typically provide more weight to the points in the center of the window to avoid the false recognition of jump discontinuities (the window is intended to discover periodic behavior, so if the values at the start and end are different, it would suggest a step, which has infinite frequency content). Common examples include Hanning and Hamming windows.

Subsequently, the DFT is applied and scaled by the length:

$$X(j\omega) = fft(x(t))/N \tag{115}$$

where N is the number of points in $X(t)$.

Next, the power spectral density (PSD) and cross power spectral density (CPSD) are found. This is done by multiplying the signal by its complex conjugate at each frequency value (which results in the squared magnitude of the complex value at that frequency). For each frequency:

$$P_{xx}(j\omega) = X(j\omega)X^*(j\omega) \tag{116}$$

$$P_{yx}(j\omega) = Y(j\omega)X^*(j\omega) \tag{117}$$

where the asterisk indicates a complex conjugate. Note that these calculations are performed at each frequency value (so if $X(j\omega)$ has 1000 data points, then $P_{xx}(j\omega)$ will also have 1000 data points).

At this point, the transfer function is found to be the ratio defined above:

$$H_{xy}(f) = \frac{P_{xy}(f)}{P_{xx}(f)} \tag{118}$$

If multiple experiments were performed, and the goal is just to find the transfer function, the the transfer function can be found as the sum of PSD and CPSD values at each frequency, i.e.

$$G_{xx}(f) = \Sigma_0^{N_{Tests}} P_{xx,N} \tag{119}$$

$$G_{yx}(f) = \Sigma_0^{N_{Tests}} P_{yx,N} \tag{120}$$

$$H_{xy}(f) = \frac{G_{xy}(f)}{G_{xx}(f)} \tag{121}$$

Of course, G_{xx} is not equal to P_{xx}. An alternative strategy is to average results at the end, once the transfer function is found, using the coherence as a weight.

The magnitude squared coherence of x with respect to u is a function with decimal values in the range (0,1) at each frequency that indicate how well x corresponds to y at that frequency. It is found using

$$C_{xy}(f) = \frac{|P_{xy}(f)|^2}{P_{xx}(f)P_{yy}(f)} \tag{122}$$

It is possible to further refine the data, e.g., through the use of averaging, which helps to smooth out and reduce the size of data.

A.1.2 Using the MATLAB tools for effective spectral analysis

While the basic approach provided in section A.1.1 is straightforward and largely acceptable for simulation, it isn't especially robust and benefits from advanced methods. Fortunately, MATLAB has a host of predefined functions and settings intended to improve results.

Most of the MATLAB tools use Welch's averaged periodogram method to find the PSD and CPSD, which essentially improves accuracy by splitting the signal into a number overlapping windowed segments and then averaging them to get the PSD estimate. It is possible to define windowing and overlap settings, though in general, the default parameters are sufficient.

The two primary tools for transfer function analysis are `tfestimate()` and `mscohere()`. The transfer function estimate function `[txy, F] = tfestimate()` provides the compex-valued transfer function given an input $x(t)$ and an output $y(t)$, at `nfft` frequencies from zero to $F_s/2$. F_s is the sampling frequency, and `nfft` is either specified by the user, or in the default case, found as the maximum of 256 or the next power of 2 greater than each section of $x(t)$ or $y(t)$. A standard input for `tfestimate()` looks like

```
>> [txy, F] = tfestimate(x, y,[],[],[],1/dt)
```

or

```
>> [txy, F] = tfestimate(x, y,[],[],nfft,1/dt)
```

if the number of points needs to be specified. The magnitude squared coherence takes the same inputs, so a corresponding coherence signal could be found using

```
>> [txy, F] = mscohere(x, y,[],[],nfft,1/dt).
```

Other useful tools include functions for the Power Spectral Density, `pwelch()`, and cross-power spectral density, `cpsd()`, both of which use Welch's method. `pwelch()` has one less input than the other signals, since it depends only on one signal.

One challenge with using MATLAB tools is that they only provide results for each experiment separately. However, results of multiple tests can easily be combined as long as the frequencies correspond; that is, `nfft` and `Fs` must match, so that the frequency range, `F`, is the same for all the experiments. Then the results can be averaged to find a better estimate. To get the best results, a weighted average should be used to ensure that better values are given more confidence and achieve better results with less, well-selected data. This can be achieved by using the coherence value as a weight and performing a weighted average:

```
>> weight = cxy.^2;
>> txy_avg = sum(weight.*txy,2)./sum(weight,2);
>> cxy_avg = sum(weight.*cxy,2)./sum(weight,2);
```

The scripts in this thesis made use of these functions for detail analysis.

A.2 MATLAB Codes for Data Analysis

This section includes three non-intuitive codes. `Plot_AvgFreqResp()` is used to obtain the bandwidth transfer function, and `Get_Bandwidth()` is subsequently applied to the transfer function to extract distinct bandwidth values by finding the +/- 3 dB crossover point. Additionally, `Plot_FRF_FitVariable()` is used on impedance data to fit a mass-spring damper model to frequency response tests. To obtain the best fit, the function uses and and index that falls inside a range of possible upper frequency bounds.

A.2.1 Plot_AvgFreqResp.m

```
function [PlotData, AvgFreqData] = Plot_AvgFreqResp(RunData, RunDetails,...
    Time_Preferences, InputFieldName, OutputFieldName, FRF_Title, ...
    FieldPrefix, PlotData_In)
%This file gets the frequency response data from
```

```
%RunData.OutputFieldName/RunData.InputFieldName. It averages over all the
%inputs.

%Get Data in and out
define_DataIn = ['dataIn = RunData(kk).' InputFieldName ';'];
define_DataOut = ['dataOut = RunData(kk).' OutputFieldName ';'];

%%%%%%%%%%%%%%% Get Key Data from the Given Inputs %%%%%%%%%%%%%%%%%%%%%%%%
%Get start and end times
t_start = Time_Preferences(1);
if length(Time_Preferences) > 1
    t_end = Time_Preferences(2);
else
    t_end = length(RunData(1).positions);
end

%Other initializations
dts = RunDetails(1).CV.dts;
PlotData = PlotData_In;

%%%%%% Get transfer function and coherence estimate from each trial %%%%%%%
for kk = 1:length(RunData)
    %Get Data in and out
    eval(define_DataIn); eval(define_DataOut);

    [txy_all(:,kk), F_all] = tfestimate(dataIn, dataOut,[],[],[],1/dts);
    [cxy_all(:,kk), Fc_all] = mscohere(dataIn, dataOut,[],[],[],1/dts);

    %Add a weighting function that is one at sampled frequencies
    %and zero otherwise
    F_cutoff(kk) = RunDetails(kk).CV.FreqF;
    i_cutoff = find(F_all > F_cutoff(kk),1);
    F_all_excitation(:,kk) = 0*F_all;
    F_all_excitation(1:(i_cutoff - 1),kk) = 1;
end
```

```
%%%%%%%%%%%%%%% Average results & get rid of useless data %%%%%%%%%%%%%%%%%
weight = (cxy_all.^2).*F_all_excitation;
txy_all_avg = sum(weight.*txy_all,2)./sum(weight,2);
cxy_all_avg = sum(weight.*cxy_all,2)./sum(weight,2);

%%%%%%%%%%%%%%% Put Results in nice form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Convert to magnitude and phase
txy_select_mag = mag2db(abs(txy_all_avg));
txy_select_phase = 180/pi*(angle(txy_all_avg));

%%%%%%%%%%%%%%% Plot Results %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RunIndex1 = num2str(RunDetails(1).ST.Run_Number);
ControlType = RunDetails(1).ST.Position_Controller;
if length(RunData) > 1
    RunIndex2 = num2str(RunDetails(length(RunData)).ST.Run_Number);
    TitleText = ['R' RunIndex1 ' - R' RunIndex2 ': ' ControlType ...
        ' Control. ' FRF_Title ' Frequency Response: Magnitude'];
else
    TitleText = ['R' RunIndex1 ': ' ControlType ' Control. ' FRF_Title ...
        ' Frequency Response: Magnitude'];
end

%Plot results with fit
fH = figure('Position',[350 50 1200 900]);
    ax(1) = subplot(3,1,1);
        semilogx(F_all, txy_select_mag,'g.');
        hold on;
        title(TitleText)
        xlabel('Frequency [Hz]')
        ylabel('Magnitude [dB]');
        xlim([.1, max(F_cutoff)]);
        grid on

    ax(2) = subplot(3,1,2);
```

```
        semilogx(F_all, txy_select_phase,'g.');
        hold on;
        title([FRF_Title ' Frequency Response: Phase'])
        xlabel('Frequency [Hz]')
        ylabel('Phase [deg]');
        grid on

    ax(3) = subplot(3,1,3);
        semilogx(Fc_all, cxy_all_avg,'g.');
        xlabel('Frequency [Hz]')

    title('Coherence');
    linkaxes(ax,'x');

%%%%%%%%%%%%%%% Save Outputs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
eval_lead_str = ['PlotData(Run_Index).' FieldPrefix '.'];

%Figure Handle
eval_struct(1).eval_str = [eval_lead_str 'FigHandle = fH;'];

%Indivual run data
eval_struct(2).eval_str = [eval_lead_str 'txy_run = txy_all(:,kk);'];
eval_struct(3).eval_str = [eval_lead_str 'cxy_run = cxy_all(:,kk);'];

%Averaged run data
eval_struct(4).eval_str = [eval_lead_str 'txy_avg = txy_all_avg;'];
eval_struct(5).eval_str = [eval_lead_str 'cxy_avg = cxy_all_avg;'];

%Frequency
eval_struct(6).eval_str = [eval_lead_str 'F_all = F_all;'];

for kk = 1:length(RunData)
    Run_Index = RunDetails(kk).ST.Run_Number;

    for jj = 1:length(eval_struct)
```

```
        eval(eval_struct(jj).eval_str);
    end
end

% To plot comparison data, since the PlotData output can be overwritten
% with other averages, it's necessary to output just the core frequency
% response data, and a run number to relate it to the run details
AvgFreqData.F_all = F_all;
AvgFreqData.txy_avg = txy_all_avg;
AvgFreqData.cxy_avg = cxy_all_avg;
AvgFreqData.Run_Index = RunDetails(1).ST.Run_Number;
%The Run_Index only really holds if all the runs being averaged are
%the same general controller

AvgFreqData.FRF_Title = FRF_Title;
AvgFreqData.F_cutoff = max(F_cutoff); %The biggest of all the sampled runs.

end
```

A.2.2 Get_Bandwidth.m

```
%% Function to get a number for bandwidth.
% Runs under assumption that output transfer function has been found.
% For that, use Plot_AvgFreqResp. For example, in
% MultiRun_Bandwidth_Fast_MPC_JPL.m, the function is called using
% PlotData = Plot_AvgFreqResp(RunData, RunDetails, 1, 'ref_x', ...
% 'positions(:,1)', 'CL Bandwidth', 'Bandwidth', PlotData);
% It can then be followed up by this function.

%Note that the input likely has to specify the first entry (PlotData(1)),
%since Bandwidth is local to the index when there are several indices.

function [PlotData] = Get_Bandwidth(PlotData)
```

```
txy_mag = abs(PlotData.Bandwidth.txy_avg);
txy_phase = 180/pi*(angle(PlotData.Bandwidth.txy_avg));
%Don't have a phase condition. But we could do that.
F_all = PlotData.Bandwidth.F_all;

idx_1 = find(txy_mag > db2mag(3), 1);
idx_2 = find(txy_mag < db2mag(-3), 1);

%Deal with cases where bounds are not crossed
if (isempty(idx_1) == 1), idx_1 = inf; end;
if (isempty(idx_2) == 1), idx_2 = inf; end;

BW_idx = min(min(idx_1, idx_2), length(F_all));
%If they're both infinite, we use the max frequency reached

PlotData.Bandwidth.BW_idx = BW_idx;
PlotData.Bandwidth.BW_Freq = F_all(BW_idx);



end
```

A.2.3 Plot_FRF_FitVariable.m

```
function [Plot_Data] = Plot_FRF_FitVariable(RunData, RunDetails, ...
    Time_Preferences, Frange, Plot_Data_In)
%this file gets the impedance transfer function from an averaged set of
%runs and then plots it against the expected values.



%%%%%%%%%%%%%%% Get Key Data from the Given Inputs %%%%%%%%%%%%%%%%%%%%%%%
%Get start and end times
t_start = Time_Preferences(1);
if length(Time_Preferences) > 1
    t_end = Time_Preferences(2);
else
```

```
    t_end = length(RunData(1).positions);
end

%Expected Curve parameters
if (strncmp(RunDetails(1).ST.Position_Controller, 'IMPEDANCE', 5) == 1)
    k_exp = RunDetails(1).CV.Z_k;
    b_exp = RunDetails(1).CV.Z_b;
    m_exp = RunDetails(1).CV.Z_m;
else
    k_exp = RunDetails(1).CV.K_Des;
    b_exp = RunDetails(1).CV.Z_Des;
    m_exp = RunDetails(1).CV.M_Des;
end
if m_exp == 0
    wn_exp = sqrt(k_exp/3);
    z_exp = b_exp/(2*3*wn_exp);
else
    wn_exp = sqrt(k_exp/m_exp);
    z_exp = b_exp/(2*m_exp*wn_exp);
end

%Other initializations
dts = RunDetails(1).CV.dts;
Plot_Data = Plot_Data_In;

%%%%%% Get transfer function and coherence estimate from each trial %%%%%%%
for kk = 1:length(RunData)
    %Get Data in and out
    dataIn = RunData(kk).positions(t_start:t_end,1) ...
        - RunDetails(kk).CV.Reference_Bias;
    dataOut = RunData(kk).F_Feel(t_start:t_end);

    [txy_all(:,kk), F_all] = tfestimate(dataIn, dataOut,[],[],[],1/dts);
    [cxy_all(:,kk), Fc_all] = mscohere(dataIn, dataOut,[],[],[],1/dts);
    %Add weighting function that's 1 at sampled frequencies and 0 otherwise
    F_cutoff(kk) = RunDetails(kk).CV.FreqF;
    i_cutoff = find(F_all > F_cutoff(kk),1);
    F_all_excitation(:,kk) = 0*F_all;
    F_all_excitation(1:(i_cutoff - 1),kk) = 1;
end

%%%%%%%%%%%%%%% Average results & get rid of useless data %%%%%%%%%%%%%%%%%
weight = (cxy_all.^2).*F_all_excitation; %cxy^2 works better than  cxy
txy_all_avg = sum(weight.*txy_all,2)./sum(weight,2);
cxy_all_avg = sum(weight.*cxy_all,2)./sum(weight,2);

%%%%%%%%%%%%%%% For Impedance Plots: Fit Curve to Results %%%%%%%%%%%%%%%%%
% Define a curve and include some pre-processing to get rid of problem
% cases like zero and NaN values.
f2ndorder = @(w,k_eq,wn,z)(20*log10(((1-(w/wn).^2).^2 ...
    + 4*z^2*(w/wn).^2).^(1/2).*k_eq));
ft=fittype('20*log10(ks^2.*((1-(x/wn).^2).^2 + 4*z^2*(x/wn).^2).^(1/2))');

%INSTEAD OF USING FULL FREQUENCY RANGE, ADJUST
best_rmse = inf; %initialize
for MaxRange = Frange(1):0.1:Frange(2)
    F_start_idx = find(F_all > 0.5, 1);
    F_end_idx = find(F_all > MaxRange,1);
    F_fit = F_all(F_start_idx:F_end_idx);
    txy_fit = txy_all_avg(F_start_idx:F_end_idx);

    nonzero_idx = all((abs(txy_fit)),2);
    nonNaN_idx = ~isnan(mag2db(abs(txy_fit)));
    idxValid = logical(nonzero_idx.*nonNaN_idx);

    %Turn warning off for this
    warning('off','curvefit:fit:noStartPoint')
        [fit_iteration, gof] = fit(2*pi*F_fit(idxValid), ...
            mag2db(abs(txy_fit(idxValid))), ft, 'Lower', [0 0 0], ...
            'StartPoint', [sqrt(k_exp) wn_exp z_exp]);
    warning('off','curvefit:fit:noStartPoint')

    %Use the goodness of fit stat to pick the range (within the acceptable
    %range) that works best
    if gof.rmse < best_rmse
        best_rmse = gof.rmse;
        fit_results = fit_iteration;
        best_fit_maxF = MaxRange;
        best_fit_idx = F_end_idx-(F_start_idx-1);
    end
end

%Now get the solutions from that
fit_solns=coeffvalues(fit_results);

%Define curves
k_fit = fit_solns(1)^2; wn_fit=abs(fit_solns(2)); z_fit=abs(fit_solns(3));
y_fit = f2ndorder(2*pi*F_fit, k_fit, wn_fit, z_fit);
y_exp = f2ndorder(2*pi*F_fit, k_exp, wn_exp, z_exp);

%%%%%%%%%%%%%%% Put Results in nice form %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Convert to magnitude and phase
txy_select_mag = mag2db(abs(txy_all_avg));
txy_select_phase = 180/pi*(angle(txy_all_avg));

%%%%%%%%%%%%%%% Plot Results %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m_fit = k_fit/wn_fit^2; b_fit = 2*z_fit*wn_fit*m_fit;
TitleText = ['Impedance F_{feel}/X_{err}. Z_{fit} = ' ...
    num2str(m_fit,'%.2f') 's^2 + ' num2str(b_fit,'%.2f') 's + ' ...
    num2str(k_fit,'%.2f'), '. Fitted over freq. range [' ...
    num2str(F_fit(1),'%.2f') '   ' num2str(best_fit_maxF) '] Hz'];
TitleText2 = ['Expected Z_{exp} = ' num2str(m_exp, '%.2f') 's^2 + ' ...
    num2str(b_exp,'%.2f') 's + ' num2str(k_exp,'%.2f')];
```

```
%Plot results with fit
fH = figure('Position',[250 50 1100 900]);
    ax(1) = subplot(3,1,1);
        semilogx(F_all, txy_select_mag,'g.');
        hold on;
        semilogx(F_fit, (y_fit),'k:');
        semilogx(best_fit_maxF, y_fit(best_fit_idx),'k.');
        semilogx(F_fit, (y_exp),'b--');
        l=legend('Actual', 'Fit', 'Freq. Range of Fit', ...
            'Expected Impedance');
        set(l,'Position',[.23 .03 .15 .05]);
        title(TitleText)
        xlabel('Frequency [Hz]')
        ylabel('Magnitude [dB]');
        xlim([F_all(1), max(F_cutoff)]);
        grid on

    ax(2) = subplot(3,1,2);
        semilogx(F_all, txy_select_phase,'g.');
        hold on;
        title(TitleText2)
        xlabel('Frequency [Hz]')
        ylabel('Phase [deg]');
        grid on

    ax(3) = subplot(3,1,3);
        semilogx(Fc_all, cxy_all_avg,'g.');
        xlabel('Frequency [Hz]')

    title('Coherence');
    linkaxes(ax,'x');

%%%%%%%%%%%%%%% Save Outputs %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RN1 = RunDetails(1).ST.Run_Number;
Plot_Data(RN1).Impedance.VarFit = [m_fit b_fit k_fit];
Plot_Data(RN1).Impedance.VarRangeLimit_Acceptable = [Frange(1) Frange(2)];
Plot_DataRN1).Impedance.VarRangeLimit_Final = best_fit_maxF;
Plot_Data(RN1).Impedance.ShortRangeIndex = [F_start_idx F_end_idx];

end
```

A.3 Hardware Deployment for Control

The basic hardware configuration has two key parts: a host and a target. The host runs MATLAB/Simulink and lets the user interface with the file during run-time, while the target runs xPC Target/Real-Time Workshop and executes the actual control code. The two are connected via a network; when the PC104 was used, a local network routed through a switch was used, while with the laptop & Quanser Q8 board configuration, a crossover cable was used to directly connect the two devices.

A.4 Overview of Data Collected

The analysis consisted of four tests, detailed in chapter 4:

1. **Slow Sine Test:** Produced RMS error as a function of the tuning parameter.

2. **System Impedance:** Produced a transfer function representation of the impedance. A fitting function was used to extrapolate stiffness and damping characteristics, again as a function of the tuning parameter.

3. **Force Disturbance/Release:** Produced an interaction force, a maximum disturbance perturbation, and post-release response characteristics. The hardware setup wasn't able to be configured to really support this test (because the shaker couldn't be disconnected, it inherently more than doubled the mass and added an inherent damping to the post-release behavior).

4. **Closed Loop Bandwidth Test:** Produced bandwidth (the +/- 3 dB cutoff value, in Hertz), as a function of the tuning parameter.

For each test, simulation and hardware variations were carried out on 12 controllers: the four benchmark controllers, the four MPC variations (v1 and v2, with and without friction compensation), and the four MPC variations with dual mode stability guarantees.

Simulations were used to explore the space of parameter variations, but only a subset of those were really comparable to hardware (based on run parameter settings). In the figures below, simulation results are indicated by dots, and hardware results are depicted by Xs. The bold (dark) dots represent simulations that are comparable to hardware, while the pastel-colored dots display the performance other parameter variations. In a few cases with the dual mode controllers, it was impossible to obtain the desired data in a reasonable amount of time due to CPU overload issues; in those cases, hardware tests were either limited or entirely excluded (indicated by a lack of corresponding X's in the figures below).

A.4.1 Slow Sine Tracking

The RMS error as a function of tuning parameters is depicted in Figures 65 - 66.

A.4.2 Impedance Fit

In the impedance fit, there was a discrepancy between the ideal version of the test in hardware and simulation. Ideally, the shaker would be subjected to very good position control, a position sweep would be provided, and the impedance would be obtained by measuring the matching interaction force measurement. Pneumatic nonlinearities are more prevalent at extreme positions (the differential pressures are inversely proportional to actuator chamber volume), so a set position is a meaningful parameter to vary. However, in practice, controlling a shaker to track perfectly in the presence of a large disturbance is a fairly difficult problem; instead, a voltage chirp may be provided to the shaker, which converts it to a force disturbance.

This approach is difficult to replicate in simulation, for several reasons. First, the shaker must be initially off, so that when the position controller is initialized, the shaker is not exerting forces on it (if the shaker and actuator are initially opposed, the stinger connecting the two may buckle, and the experiment must be aborted). Therefore, the shaker has to be turned on manually inside a set time-frame. Since the shaker's dynamics are only approximately known, the conversion of voltage to force, and the timing, are nearly

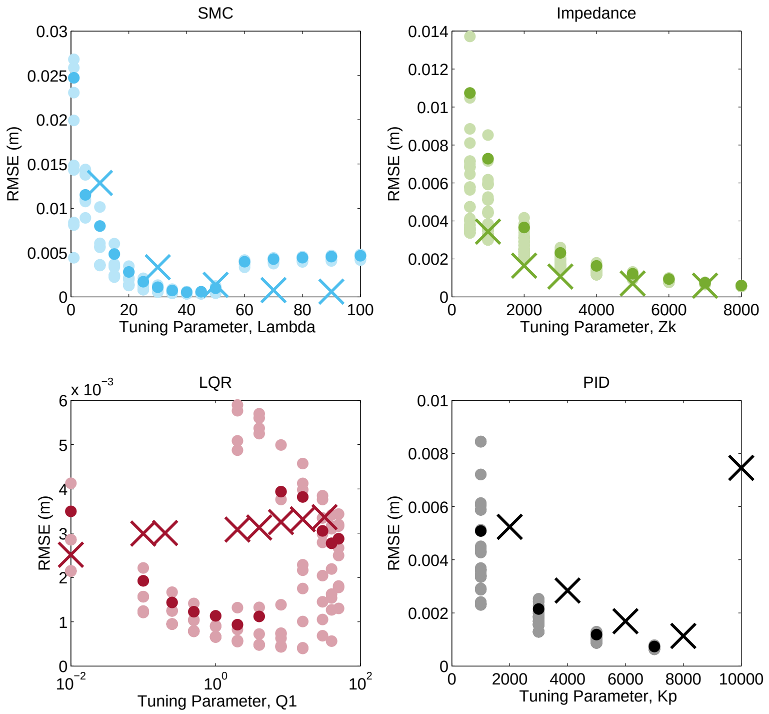

Figure 64: Benchmarks: tuning parameter vs. slow sine RMSE.

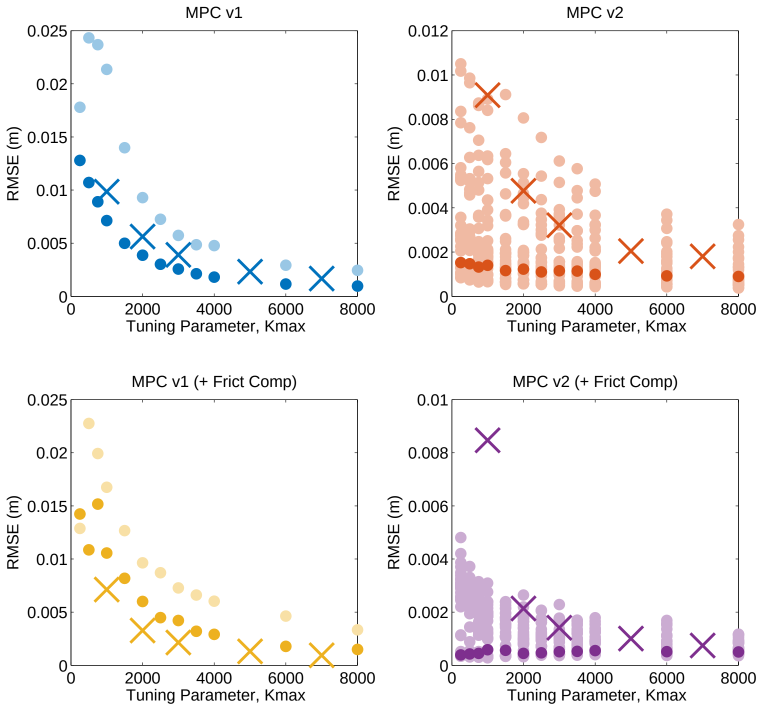

Figure 65: MPC: tuning parameter vs. slow sine RMSE.

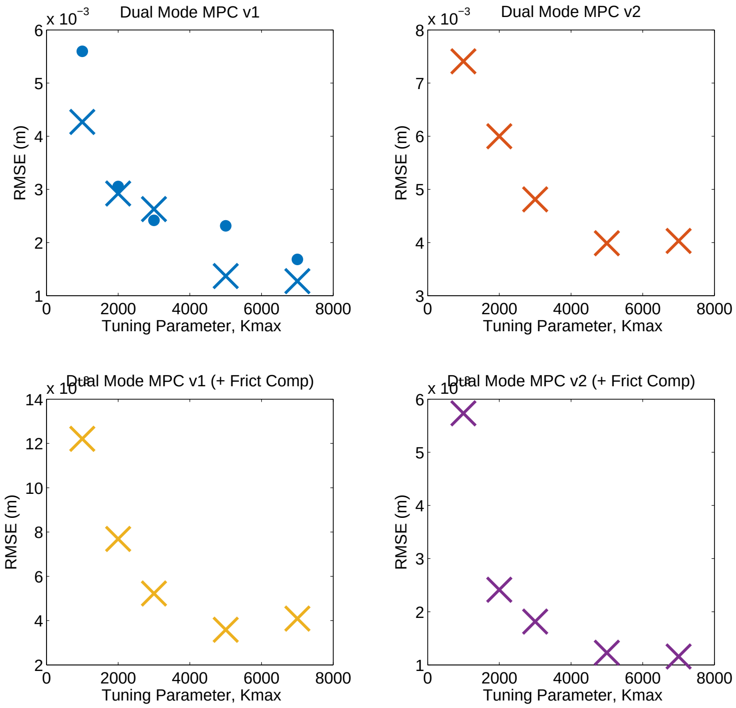

Figure 66: Dual mode MPC: tuning parameter vs. slow sine RMSE.

impossible to match in simulation.

Instead, the tests in simulation were primarily conducted using the position chirp, which is easily automated and straightforward to control, since the shaker is modeled as a perfect force source, and there are no limits on the gains of the corresponding PD controller. Additionally, however, a test similar to that tried in hardware was run: by assessing hardware measurements, a force disturbance was provided with bias and amplitudes that closely matched those observed in hardware. With this approach, it was possible to verify that the two methods do, in fact, produce very similar results.

In Figures 67 - 69, the bold circles depict simulations using the method matching the hardware tests (impedance obtained from a force chirp and measured position response), while the bold triangles illustrate the performance of the idealized test – impedance based on position chirp and measured interaction force.

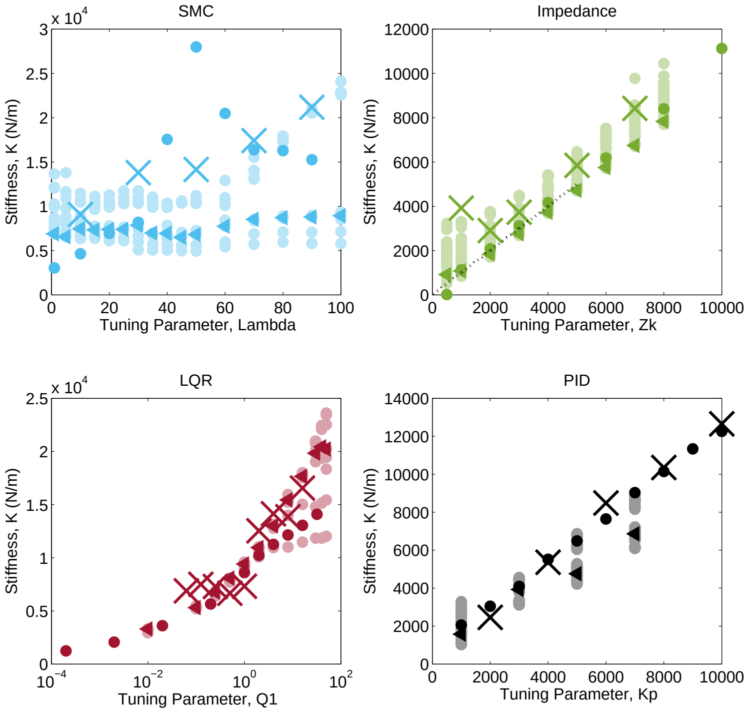

Figure 67: Benchmarks: tuning parameter vs. observed stiffness fit.

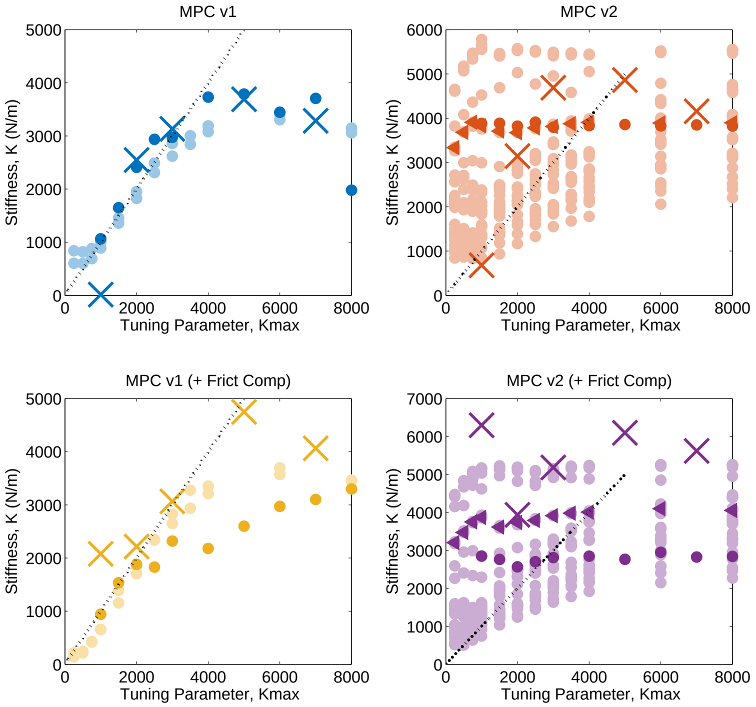

Figure 68: MPC: tuning parameter vs. observed stiffness fit.

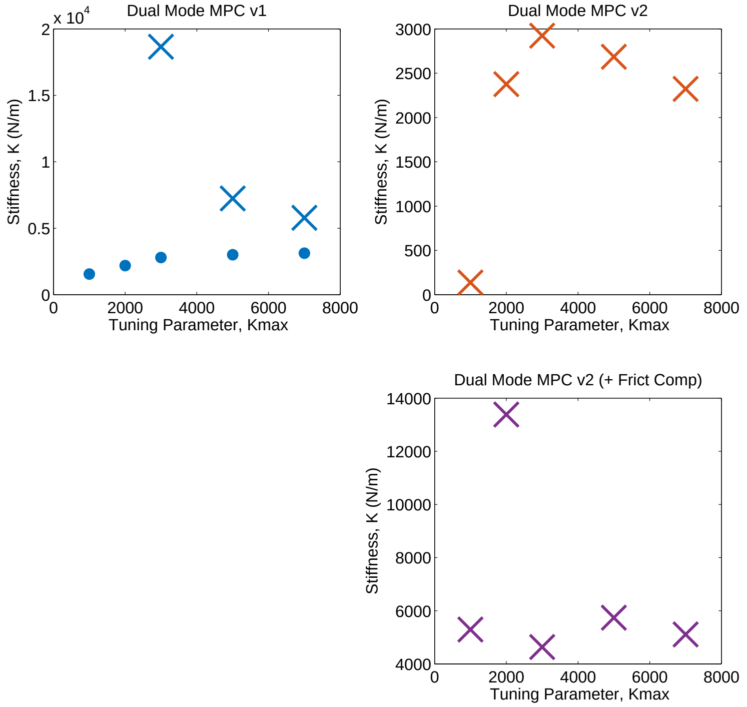

Figure 69: Dual mode MPC: tuning parameter vs. observed stiffness fit.

A.4.3 Force Perturbation/Release

Figures 70 - 72 illustrate the interaction force for a position-controlled disturbance to the actuator's intended position setpoint, applied by the shaker. In Figure 67, it is evident that the hardware results for SMC are much lower than the simulated ones. This was actually caused by a hardware limitation: even with the current limit at its maximum value, the shaker was unable to supply a sufficiently strong force to perturb the system by the desired amount of 0.01 inches for $\lambda > 10$. As a result, the forces are simply at their maximum possible value with the hardware setup.

In simulation, this is achieved by representing the shaker as a perfect force source and using an idealized PD controller to close the loop on position. As long as the shaker is significantly stronger than the actuator, this is a sufficient representation. In practice, a frequency response analysis was conducted to perform a system identification analysis on the electromagnetic shaker attached to an unpressurized cylinder. A Laser Doppler Vibrometer (LDV) was used to measure position, resulting in a system with a transfer function of $X/V = K/(s^2 + 43s)$, where X is shaker position, V is the voltage input to the amplifier, and K is dependent on the shaker amplifier scaling, and the denominator zero is due to damping of the unpressurized cylinder dynamics, verifying that the shaker itself – a linear motor – is a near-ideal force source. A discrete PD controller, designed for a fast response time and zero steady-state error, was used to control the shaker, though realistic input voltage limits and cylinder dynamics led to reduced tracking performance compared to simulation.

A.4.4 Closed Loop Bandwidth

The bandwidth as a function of tuning parameter is detailed in Figures 73 - 75.

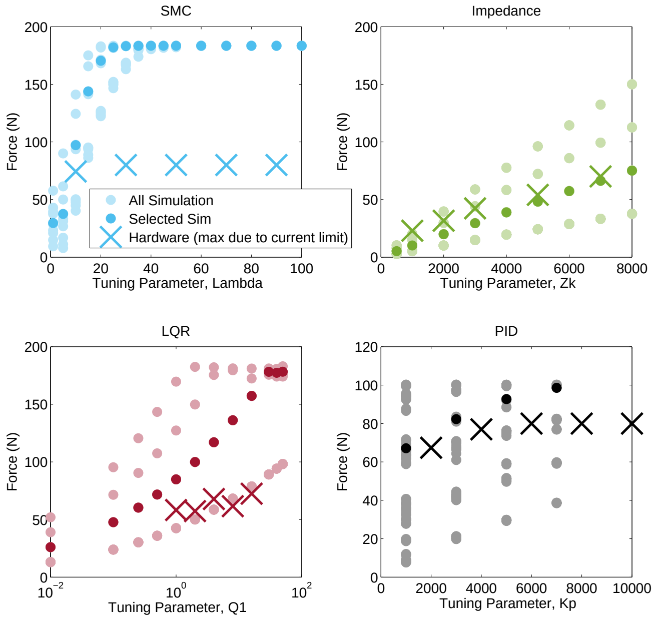

Figure 70: Benchmarks: tuning parameter vs. interaction force.

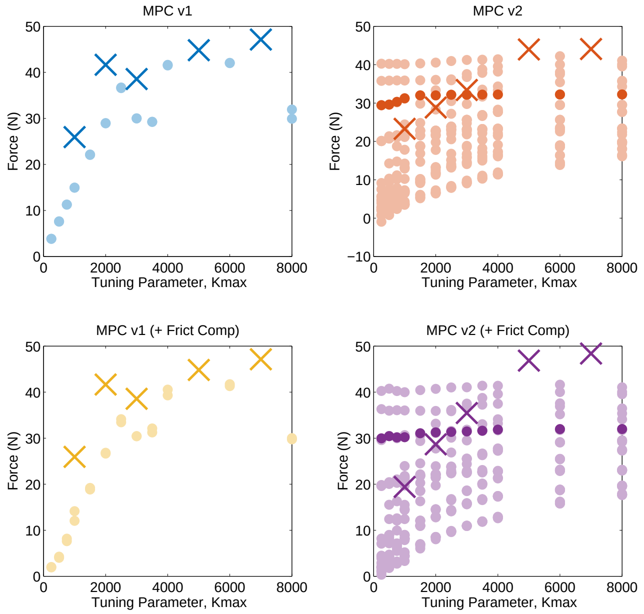

Figure 71: MPC: tuning parameter vs. interaction force.

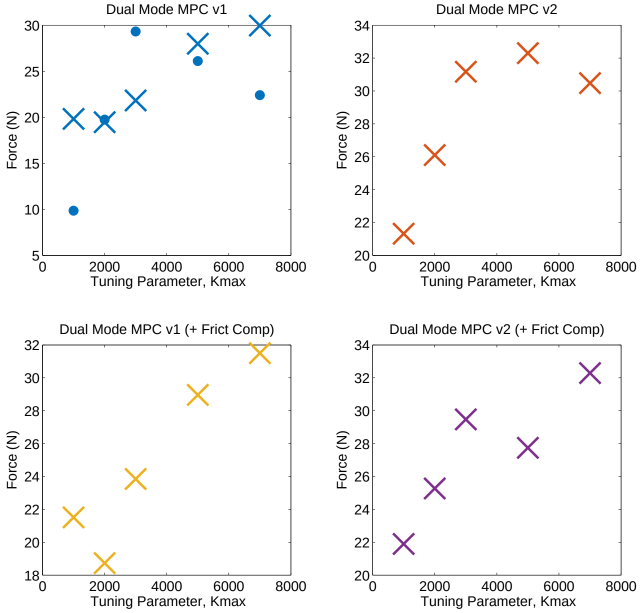

Figure 72: Dual mode MPC: tuning parameter vs. interaction force.

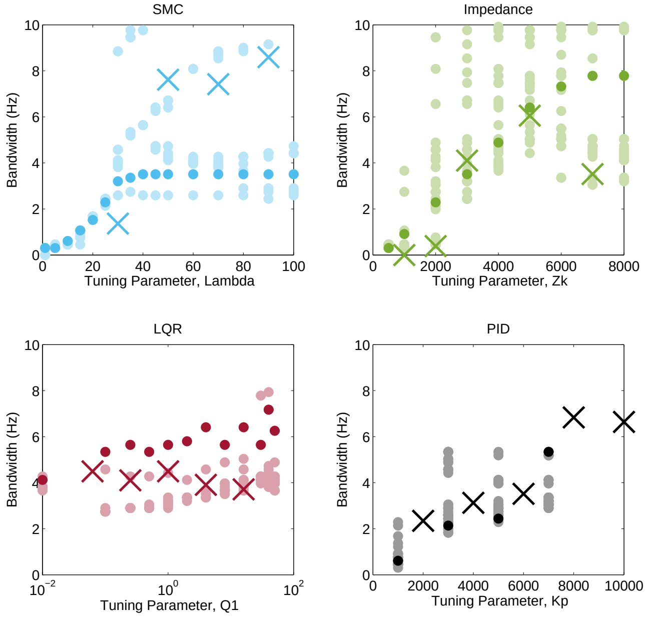

Figure 73: Dual mode MPC: tuning parameter vs. position tracking bandwidth.

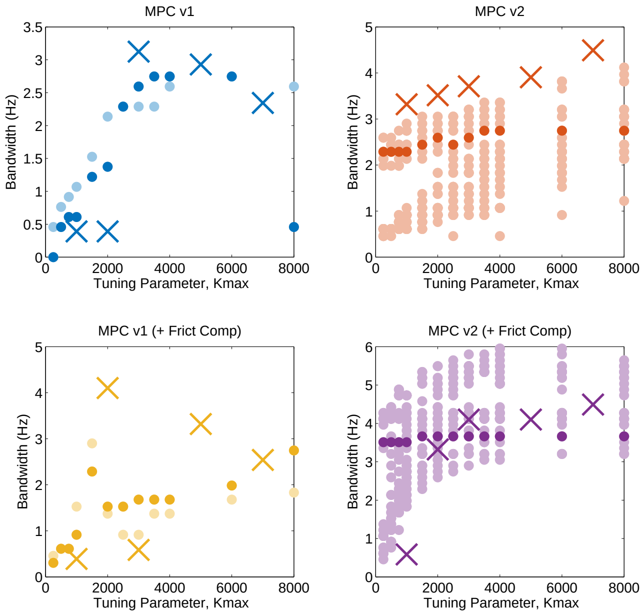

Figure 74: Dual mode MPC: tuning parameter vs. position tracking bandwidth.

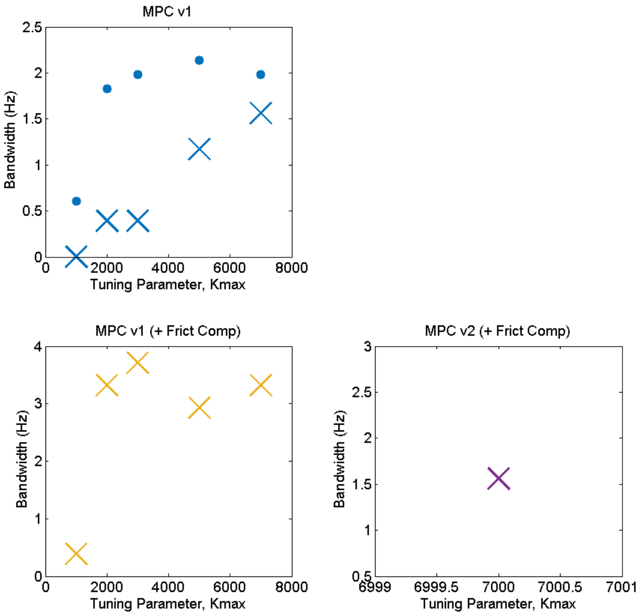

Figure 75: Dual mode MPC: tuning parameter vs. position tracking bandwidth.

A.4.5 Combination Plot Construction

The data in the previous sections was used to make combination plots. For example, if a plot is being designed to compare stiffness and bandwidth for a particular controller, it is first necessary to run simulations and hardware tests that determine these metrics individually. Figure 76 shows how, for a PID controller, stiffness and bandwidth are first assembled as functions of a dominant tuning parameter, K_p. The grey dots are all simulation, the black dots are simulation values with tuning parameters that match those of the hardware assessment, and the hardware checks are shown as X's.

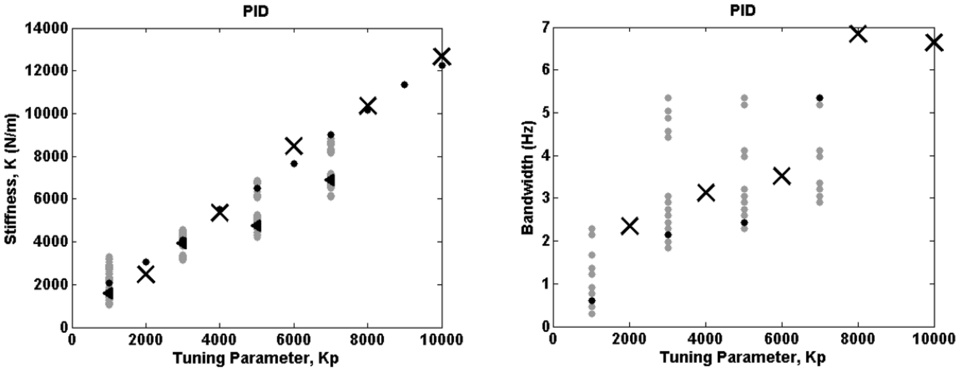

Figure 76: Example metrics for use in combination plots. Left: stiffness as a function K_p. Right: bandwidth as a function of K_p.

A combination plot is generated using parametric plotting and combining the two results, as seen in Figure 77. Typically, within the thesis, the data is then limited only to values that meet some normalized velocity error criterion, as discussed in section 4.3.2.

This style of plotting has a few noteworthy characteristics. First, because data is selected by metric using the dominant tuning parameter as an input, it follows clear trends in the individual plots, but those trends are not always as evident in the combination plot. Instead, results may be more scattered. It would be possible to conduct more extensive simulations to see the exact distribution of possible controller capabilities, but such a survey would require considerable further time and resources. Similarly, the hardware check locations were selected to be consistent, and don't necessarily correspond to exact simulation positions, as seen on the right in Figure 77. As a result, in the combination plot, there are X's that are not near any discernible simulation result. Again, this issue could be reduced using more

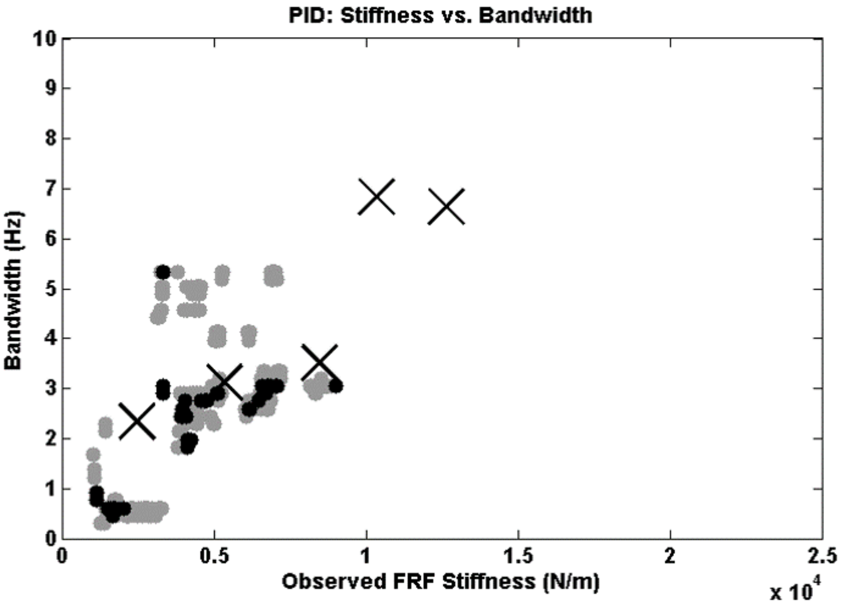

Figure 77: Example metrics for use in combination plots. Left: stiffness as a function K_p. Right: bandwidth as a function of K_p.

thorough hardware or simulation validation (more parameter variations), but that would require more time and resources. In either case, it is important to emphasize that the goal of the hardware verification is to identify and verify trends, which can clearly be done from the individual and combination plots.

APPENDIX B

MPC IMPLEMENTATION

B.1 Input Constraints

To obtain heuristics on achievable maximum and minim values for Δu, a model-based approach was used. First, a feedback-linearization approach (as discussed in section 4.2.2) was assumed for force control, as it explicitly formulated force and input dynamics in terms of valve inputs. In practice, a PID controller was shown to be slightly more capable and was used in stead, but that simply makes these constraints a conservative estimate of the overall capability.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{F} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{m} & 0 \\ 0 & -\left(\frac{P_aA_a}{x_a} + \frac{P_bA_b}{L-x_a}\right) & 0 & 0 \\ 0 & -\left(\frac{P_aA_a}{x_a} + \frac{P_bA_b}{L-x_a}\right) & \beta_0 & -\beta_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ F \\ u \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \left(\frac{RT}{x_a}c_{d,a}\psi_a + \frac{RT}{L-x_a}c_{d,b}\psi_b\right) \\ \left(\frac{RT}{x_a}c_{d,a}\psi_a + \frac{RT}{L-x_a}c_{d,b}\psi_b\right) \end{bmatrix} g_a \tag{123}$$

Using a simple approximation of the matrix exponential, $e^{Adt} = I + Adt$, the discretized system can be defined as

$$\begin{bmatrix} x_1 \\ x_2 \\ F \\ u \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & \frac{\Delta t}{m} & 0 \\ 0 & -\Delta t\left(\frac{P_aA_a}{x_a} + \frac{P_bA_b}{L-x_a}\right) & 1 & 0 \\ 0 & -\Delta t\left(\frac{P_aA_a}{x_a} + \frac{P_bA_b}{L-x_a}\right) & \beta_0\Delta t & 1-\beta_0\Delta t \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ F \\ u \end{bmatrix}_k + \Delta t \begin{bmatrix} 0 \\ 0 \\ \left(\frac{RT}{x_a}c_{d,a}\psi_a + \frac{RT}{L-x_a}c_{d,b}\psi_b\right) \\ \left(\frac{RT}{x_a}c_{d,a}\psi_a + \frac{RT}{L-x_a}c_{d,b}\psi_b\right) \end{bmatrix} g_{a,k} \tag{124}$$

Then limits on u_{k+1} can easily be found from the discretization:

$$\begin{aligned} u_{k+1} &= \beta_0 F_k \Delta t + (1 - \beta_0 \Delta t) u_k \\ &- \left(\frac{P_a A_a}{x_a} + \frac{P_b A_b}{L - x_a} \right) \Delta t x_{2,k} + \left(\frac{RT}{x_a} c_{d,a} \psi_a + \frac{RT}{L - x_a} c_{d,b} \psi_b \right) g_{a,k} \Delta t \end{aligned} \tag{125}$$

It is evident that the constraining values depend only on $g_{a,max}$ and $g_{a,min}$, and since the coefficient on orifice area (defined as A_3 below) is always positive,

$$\begin{aligned} [u_{k+1}]_{max} &= A_1 F_k + A_2 + A_3 g_{a,max} + A_4 u_k \\ [u_{k+1}]_{min} &= A_1 F_k + A_2 + A_3 g_{a,min} + A_4 u_k \\ A_1 &= (1 - \beta_0 \Delta t) \\ A_2 &= - \left(\frac{P_{a,k=0} A_a}{x_{a,k=0}} + \frac{P_{b,k=0} A_b}{L - x_{a,k=0}} \right) \Delta t x_{2,k=0} \\ A_3 &= \left(\frac{RT}{x_{a,k=0}} c_{d,a,k=0} \psi_{a,k=0} + \frac{RT}{L - x_{a,k=0}} c_{d,b,k=0} \psi_{b,k=0} \right) \Delta t \\ A_4 &= \beta_0 \Delta t \end{aligned} \tag{126}$$

The valve is characterized by a 125 Hz critical frequency, which – from the data sheet [34] – refers to “the 3 dB frequency at the maximum movement stroke of the piston spool”, which is similar to a cutoff frequency for the system bandwidth. This frequency can be used to impose rate limits on valve orifice area, g. If the valve can move its entire stroke at 125 Hz, then it can be estimate that the orifice area range, $g_{span} = g_{max} - g_{min}$, can be covered in at most $1/125 = 0.008$ s. Conservatively, then, the maximum amount that the orifice area can change by in a time step is $\Delta g_{max} = \frac{dt}{.008} g_{span}$.

Updated limits on u_{max} and u_{min} at each time step were found that includes effects from both the absolute extremes of g, and the rate limits:

$$g_{max} = min(g + \frac{dt}{.008} g_{span}, 1.05 \text{ x } 10^{-5}) \tag{127}$$

$$g_{min} = max(g - \frac{dt}{.008} g_{span}, -1.05 \text{ x } 10^{-5}) \tag{128}$$

For the feedback linearization case, these hover around Δu = 6 - 10 N, which – as noted earlier – is slightly more conservative for the force PID controller. Thus, a value of $||\Delta u||_{max} = 10$ was used in practice.

B.2 MATLAB Script to Find Terminal Constraint Horizons

```
%% Script to get terminal weight and constraint matrices
clear all; clc;
x0 = [0.05 1 100 1000 10 100]';

dt = 0.01;
m = 3; wf = 2*pi*45; zf = 0.5;
Ac = [0 1 0 0 0; 0 0 1/m 0 0; 0 0 0 1 0; 0 0 -wf^2 -2*zf*wf wf^2; ...
    0 0 0 0 0];
%Augment to include udot as input and xI and u as states:
%[pos vel f1 f2 xI u udot]
Ac = [0 1 0 0 0 0 0; 0 0 1/m 0 0 0 0; 0 0 0 1 0 0 0; ...
    0 0 -wf^2 -2*zf*wf 0 wf^2 0; 1 0 0 0 0 0 0; ...
    0 0 0 0 0 0 1; 0 0 0 0 0 0 0];

%Discretize
G1 = expm(Ac*dt);

%Rearrange again into form used in solver
SysModel.G = G1(1:6,1:6);
SysModel.H = G1(1:6,7);
SysModel.C = [1 0 0 0 0 0];
SysModel.D = 0;
SysModel.dt = dt;

SysCost.Q = diag([100 0 0 0 1 0]);
SysCost.R = 0;

%% Check for Positive Definiteness
SysVer.obsv = (obsv(SysModel.G, SysCost.Q^(1/2)));
if rank(SysVer.obsv) == size(SysVer.obsv,2)
    disp('(A,Q^{1/2} is an observable pair so J is positive definite!')
else
```

```
    disp('J is NOT positive definite');
end

%% Calculate Terminal Cost and infinite horizon gain matrix

[SysCost.Qbar,L,SysCost.Kbar,info] = dare(SysModel.G, SysModel.H , ...
    SysCost.Q, SysCost.R,'report');

%% Calculate Terminal Input Constraints
clearvars -except SysModel SysCost x0;

%In this case, input is really dU
Nmax = 10; %Total number of points to check
umax = 10; umin = -umax; nc = 2;

[s, params, w] = pred_parameters(SysModel, SysCost, umax, nc);
K = -SysCost.Kbar;
Phi = SysModel.G + SysModel.H*K;

%Populate b0 and Bx matrices up to some fixed point
Bx = zeros(Nmax*params.nu, params.nx);
b0 = zeros(Nmax*params.nu, params.nu);
for i = 1:Nmax
    Bx((i-1)*params.nu+1:i*params.nu,:) = K*Phi^(i-1);
    b0_max((i-1)*params.nu+1:i*params.nu) = umax;
    b0_min((i-1)*params.nu+1:i*params.nu) = umin;
end

%Solve linear programs to see how much constraint checking is really nec.
options = optimset('display','off'); zbnd = x0;
fval = 1e3; i = 1;
while i < Nmax && abs(fval) > umax
    fT = K*Phi^(i); %One step ahead
    Bx_mat = Bx(1:i*params.nu,:);
    b0_max_vec = b0_max(1:i*params.nu);
    b0_min_vec = b0_min(1:i*params.nu);
    [x, fval] = linprog(-fT, [Bx_mat; -Bx_mat], ...
        [b0_max_vec; -b0_min_vec],[],[],-zbnd, zbnd, [], options);
    disp(['Case for i = ' num2str(i) ': ' num2str(x') ...
        ' and fval =  ' num2str(fval)]);
    i = i + 1;
end
i = i-1; %Get rid of that last addition.

%Return verdict
if abs(fval) < umax
    disp(['Constraint checking horizon set at Nc = ' num2str(i-1)]);
else
    disp(['Need a longer constraint checking horizon. Nc = ' ...
        num2str(i-1) ' is insufficient.']);
    disp(['Present umax = ' num2str(abs(fval))]);
end

%% Calculate Terminal State Constraint: v1 States
clearvars -except SysModel SysCost x0 params; disp(' ');

Nmax = 10; %Total number of points to check
xmax = [.044 1e10 100 1e10 1e10 10]'; xmin = -xmax; xmin(1) = 0;
x_select(1,:) = [0 0 1 0 0 0]; state_select(1).Name = 'Force';%Force
x_select(2,:) = [0 0 0 0 0 1]; state_select(2).Name = 'Ref Rate';%Ref Rate
x_select(3,:) = [1 0 0 0 0 0]; state_select(3).Name = 'Position';%Ref Rate

K = -SysCost.Kbar;
Phi = SysModel.G + SysModel.H*K;
params.ncx = size(x_select,1);

%Populate b0 and Bx matrices up to some fixed point
Bx_x = zeros(Nmax*params.ncx, params.nx);
b0_x_max = zeros(Nmax*params.ncx, 1);
b0_x_min = zeros(Nmax*params.ncx, 1);
```

```
for i = 1:Nmax
    Bx_x((i-1)*params.ncx+1:i*params.ncx,:) = x_select*Phi^(i-1);
    b0_x_max((i-1)*params.ncx+1:i*params.ncx) = x_select*xmax;
    b0_x_min((i-1)*params.ncx+1:i*params.ncx) = x_select*xmin;
end

%Solve linear programs to see how much constraint checking is really nec.
options = optimset('display','off'); zbnd = x0;
fval_v1 = 1e3*ones(params.ncx,1); i = 1;
while i < Nmax && sum(abs(fval_v1) > x_select*xmax)
    fT = Phi^(i); %Just need to select the correct states
    Bx_mat = Bx_x(1:i*params.ncx,:);
    b0_max_vec = b0_x_max(1:i*params.ncx);
    b0_min_vec = b0_x_min(1:i*params.ncx);
    for j = 1:size(x_select,1)
        [x, fval_v1(j)] = linprog(-x_select(j,:)*fT, ...
            [Bx_mat; -Bx_mat], [b0_max_vec; -b0_min_vec],...
            [],[],-zbnd, zbnd, [], options);
        disp([state_select(j).Name ': Case for i = ' num2str(i) ': ' ...
            num2str(x') ' and fval =  ' num2str(fval_v1')]);
    end
    i = i + 1;
end
i = i-1; %Get rid of that last addition.

%Return verdict
if abs(fval_v1) < x_select*xmax
    disp(['V1 X: Constraint checking horizon set at Nc = ' num2str(i-1)]);
else
    disp(['V1 X: Need a longer constraint checking horizon. Nc = ' ...
        num2str(i-1) ' is insufficient.']);
    disp(['Present xmax = ' num2str(abs(fval_v1'))]);
end

%% Calculate Terminal State Constraint: v2 States
```

```matlab
clearvars -except SysModel SysCost x0 params; disp(' ');

K = -SysCost.Kbar;
Zk = 2000; Zb = 100;

Nmax = 10; %Total number of points to check
xmax = [.044 1e10 100 1e10 1e10 10]'; xmin = -xmax; xmin(1) = 0;
x_select(1,:) = [Zk Zb -Zk 0 0 0] - K; state_select(1).Name = 'Force';
x_select(2,:) = [0 0 0 0 0 1]; state_select(2).Name = 'Ref Rate';
x_select(3,:) = [1 0 0 0 0 0]; state_select(3).Name = 'Position';

K = -SysCost.Kbar;
Phi = SysModel.G + SysModel.H*K;
params.ncx = size(x_select,1);

%Populate b0 and Bx matrices up to some fixed point
Bx_x = zeros(Nmax*params.ncx, params.nx);
b0_x_max = zeros(Nmax*params.ncx, 1);
b0_x_min = zeros(Nmax*params.ncx, 1);
for i = 1:Nmax
    Bx_x((i-1)*params.ncx+1:i*params.ncx,:) = x_select*Phi^(i-1);
    b0_x_max((i-1)*params.ncx+1:i*params.ncx) = x_select*xmax;
    b0_x_min((i-1)*params.ncx+1:i*params.ncx) = x_select*xmin;
end

%Solve linear programs to see how much constraint checking is really nec.
options = optimset('display','off'); zbnd = x0;
fval_v1 = 1e3*ones(params.ncx,1); i = 1;
while i < Nmax && sum(abs(fval_v1) > x_select*xmax)
    fT = Phi^(i); %Just need to select the correct states
    Bx_mat = Bx_x(1:i*params.ncx,:);
    b0_max_vec = b0_x_max(1:i*params.ncx);
    b0_min_vec = b0_x_min(1:i*params.ncx);
    for j = 1:size(x_select,1)
        [x, fval_v1(j)] = linprog(-x_select(j,:)*fT, [Bx_mat; -Bx_mat], ...
            [b0_max_vec; -b0_min_vec],[],[],-zbnd, zbnd, [], options);
        disp([state_select(j).Name ': Case for i = ' num2str(i) ': ' ...
            num2str(x') ' and fval =  ' num2str(fval_v1')]);
    end
    i = i + 1;
end
i = i-1; %Get rid of that last addition.

%Return verdict
if abs(fval_v1) < x_select*xmax
    disp(['V2 X: Constraint checking horizon set at Nc = ' num2str(i-1)]);
else
    disp(['V2 X: Need a longer constraint checking horizon. Nc = ' ...
        num2str(i-1) ' is insufficient.']);
    disp(['Present xmax = ' num2str(abs(fval_v1'))]);
end
```

B.3 MPC Implementation – Target Hardware & CPU Overload Concerns

One limiting concern was the issue of CPU overloads, which may happen seemingly at random. The effect is that, after starting a trial at a seemingly well-selected sample time, the target will crash and return a CPU overload error, displaying the associated Task Execution Time (TeT) that caused the crash. Typically, acceptable TeTs average 1/3 - 1/2 the sample time (since periodic spikes are anticipated). A crash and CPU overload occurs when the TeT jumps unexpectedly. For example, if a block is written that has an acceptable range of inputs that produce numeric outputs, but it is suddenly given an input that produces a NaN (Not-a-Number) output, the subsequent effect can be significant enough to cause a TeT spike and crash the program. However, these kinds of issues are detectable & reproducible. Other sources include hardware latencies and target features, such as power management, which must be altered in the BIOS.

In the case with the dual-mode controller and CVXgen, the dual-mode implementation caused CPU overloads at changing and seemingly random times during execution. The overloads were never observed with the simpler problem, so it is presumed that they would result from the more constrained optimization. Changes to the memory usage (outputs saved, data logged, etc.) were tried, but were unsuccessful at mitigating these limitations. Instead, the main limitation is likely the hardware, which was limited to 2 GB of memory and 2 Ghz, as well as the older A/D and D/A boards, which have been in use for several years. Due to time constraints, this hypothesis could not be more thoroughly tested. However, a user encountering similar issues might refer to Mathworks's support communities (e.g., 1, 2) and official documentation, which includes strategies for detection and work-arounds (such as continuing to run after a CPU overload), which might prove to be more revealing.

APPENDIX C

1-DOF MODELING & LINEARIZATION

This section covers modeling of the pneumatic actuator in greater detail than is covered in the main portion of the thesis.

C.1 Linearized Model Using Polynomial Mass Flow

This approach uses the polynomial approximation of mass flow introduced in section 3.4.3. An especially nice result occurs when the input is treated as A_{eq}, using the assumption that $A_{eq,a} = -A_{eq,b}$, using polynomial models of $\dot{m} = f(P_{(a,b)}, A_{eq})$ for mass flow. This system is summarized as:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ P_a \\ P_b \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{(P_aA_a - P_bA_b) - F_{frict}}{M} \\ \frac{\dot{m}_aRT - P_aA_ax_2}{V_a} \\ \frac{\dot{m}_bRT - P_bA_bx_2}{V_b} \end{bmatrix} \tag{129}$$

The volume terms in the denominator are clearly functions of position, x_1, but depend on the origin. Assuming the position is measured from the center position, the volume terms are written as:

$$V_a = L + ds_a + x_1 \quad \text{and} \quad V_b = L + ds_b - x_1 \tag{130}$$

where $ds_{(a,b)}$ is the dead space in the ends of the chamber, and L is the stroke length. Further, mass flow is defined using equation (19), such that

$$\dot{m}_a(A_{eq}, P_a) = \dot{m}(A_{eq}, P_a) \quad \text{and} \quad \dot{m}_b(A_{eq}, P_b) = \dot{m}(-A_{eq}, P_b) \tag{131}$$

Finally, while friction force is generally either omitted or found using a Streibeck curve, it is possible to approximate it, e.g. with a viscous friction model, $F_{frict} = bx_2$. A linearization can then be found by determining the Jacobian (using a Taylor Series approach):

$$\Delta\dot{X} = J\Delta X \quad \text{or} \quad \dot{X} = J\left(X - X|_{t=T_0}\right) + \dot{X}|_{t=T_0} \tag{132}$$

J is the Jacobian, which is found to be:

$$J =$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -b & \frac{A_a}{M} & -\frac{A_b}{M} & 0 \\ \frac{A_a\,P_a\,x_2 - R\,T\,\dot{m}_a(A_{eq},P_a)}{A_a\,(L+ds_a+x_1)^2} & -\frac{P_a}{L+ds_a+x_1} & -\frac{A_a\,x_2 - R\,T\,\Phi_{2a}}{A_a\,(L+ds_a+x_1)} & 0 & \frac{R\,T\,\Phi_{3a}}{A_a\,(L+ds_a+x_1)} \\ \frac{A_b\,P_b\,x_2 + R\,T\,\dot{m}_b(A_{eq},P_b)}{A_b\,(L+ds_b-x_1)^2} & \frac{P_b}{L+ds_b-x_1} & 0 & \frac{A_b\,x_2 + R\,T\,\Phi_{2b}}{A_b\,(L+ds_b-x_1)} & -\frac{R\,T\,\Phi_{3b}}{A_b\,(L+ds_b-x_1)} \end{bmatrix} \tag{133}$$

Alternatively, the Jacobian can be found via symbolic differentiation (in MATLAB), which produces this similar-looking matrix:

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & -b & \frac{A_a}{M} & -\frac{A_b}{M} & 0 \\ \frac{A_a\,P_a\,x_2 - R\,T\,\Phi_{1a}}{A_a\,(L+ds_a+x_1)^2} & -\frac{P_a}{L+ds_a+x_1} & -\frac{A_a\,x_2 - R\,T\,\Phi_{2a}}{A_a\,(L+ds_a+x_1)} & 0 & \frac{R\,T\,\Phi_{3a}}{A_a\,(L+ds_a+x_1)} \\ \frac{A_b\,P_b\,x_2 + R\,T\,\Phi_{1b}}{A_b\,(L+ds_b-x_1)^2} & \frac{P_b}{L+ds_b-x_1} & 0 & \frac{A_b\,x_2 + R\,T\,\Phi_{2b}}{A_b\,(L+ds_b-x_1)} & -\frac{R\,T\,\Phi_{3b}}{A_b\,(L+ds_b-x_1)} \end{bmatrix} \tag{134}$$

where

$$\begin{aligned} \Phi_{1a} = \;& p_{41}\,{A_{eq}}^4\,P_a + p_{40}\,{A_{eq}}^4 + p_{32}\,{A_{eq}}^3\,{P_a}^2 + p_{31}\,{A_{eq}}^3\,P_a + p_{30}\,{A_{eq}}^3 \\ & + p_{23}\,{A_{eq}}^2\,{P_a}^3 + p_{22}\,{A_{eq}}^2\,{P_a}^2 + p_{21}\,{A_{eq}}^2\,P_a + p_{20}\,{A_{eq}}^2 \\ & + p_{14}\,A_{eq}\,{P_a}^4 + p_{13}\,A_{eq}\,{P_a}^3 + p_{12}\,A_{eq}\,{P_a}^2 + p_{11}\,A_{eq}\,P_a + p_{10}\,A_{eq} \\ & + p_{05}\,{P_a}^5 + p_{04}\,{P_a}^4 + p_{03}\,{P_a}^3 + p_{02}\,{P_a}^2 + p_{01}\,P_a + p_{00} \end{aligned} \tag{135}$$

$$\begin{aligned} \Phi_{2a} = \;& p_{41}\,{A_{eq}}^4 + 2\,p_{32}\,{A_{eq}}^3\,P_a + p_{31}\,{A_{eq}}^3 \\ & + 3\,p_{23}\,{A_{eq}}^2\,{P_a}^2 + 2\,p_{22}\,{A_{eq}}^2\,P_a + p_{21}\,{A_{eq}}^2 \\ & + 4\,p_{14}\,A_{eq}\,{P_a}^3 + 3\,p_{13}\,A_{eq}\,{P_a}^2 + 2\,p_{12}\,A_{eq}\,P_a + p_{11}\,A_{eq} \\ & + 5\,p_{05}\,{P_a}^4 + 4\,p_{04}\,{P_a}^3 + 3\,p_{03}\,{P_a}^2 + 2\,p_{02}\,P_a + p_{01} \end{aligned} \tag{136}$$

$$\begin{aligned}\Phi_{3a} = \;& 4\,p_{41}\,{A_{eq}}^3\,P_a + 4\,p_{40}\,{A_{eq}}^3 + 3\,p_{32}\,{A_{eq}}^2\,{P_a}^2 + 3\,p_{31}\,{A_{eq}}^2\,P_a + 3\,p_{30}\,{A_{eq}}^2 \\ & + 2\,p_{23}\,A_{eq}\,{P_a}^3 + 2\,p_{22}\,A_{eq}\,{P_a}^2 + 2\,p_{21}\,A_{eq}\,P_a + 2\,p_{20}\,A_{eq} \\ & + p_{14}\,{P_a}^4 + p_{13}\,{P_a}^3 + p_{12}\,{P_a}^2 + p_{11}\,P_a + p_{10} \qquad (137)\end{aligned}$$

$$\begin{aligned}\Phi_{1b} = \;& p_{41}\,{A_{eq}}^4\,P_b + p_{40}\,{A_{eq}}^4 - 1\,p_{32}\,{A_{eq}}^3\,{P_b}^2 - 1\,p_{31}\,{A_{eq}}^3\,P_b - 1\,p_{30}\,{A_{eq}}^3 \\ & + p_{23}\,{A_{eq}}^2\,{P_b}^3 + p_{22}\,{A_{eq}}^2\,{P_b}^2 + p_{21}\,{A_{eq}}^2\,P_b + p_{20}\,{A_{eq}}^2 \\ & - 1\,p_{14}\,A_{eq}\,{P_b}^4 - 1\,p_{13}\,A_{eq}\,{P_b}^3 - 1\,p_{12}\,A_{eq}\,{P_b}^2 - 1\,p_{11}\,A_{eq}\,P_b - 1\,p_{10}\,A_{eq} \\ & + p_{05}\,{P_b}^5 + p_{04}\,{P_b}^4 + p_{03}\,{P_b}^3 + p_{02}\,{P_b}^2 + p_{01}\,P_b + p_{00} \qquad (138)\end{aligned}$$

$$\begin{aligned}\Phi_{2b} = \;& p_{41}\,{A_{eq}}^4 - 2\,p_{32}\,{A_{eq}}^3\,P_b - 1\,p_{31}\,{A_{eq}}^3 \\ & + 3\,p_{23}\,{A_{eq}}^2\,{P_b}^2 + 2\,p_{22}\,{A_{eq}}^2\,P_b + p_{21}\,{A_{eq}}^2 \\ & - 4\,p_{14}\,A_{eq}\,{P_b}^3 - 3\,p_{13}\,A_{eq}\,{P_b}^2 - 2\,p_{12}\,A_{eq}\,P_b - 1\,p_{11}\,A_{eq} \\ & + 5\,p_{05}\,{P_b}^4 + 4\,p_{04}\,{P_b}^3 + 3\,p_{03}\,{P_b}^2 + 2\,p_{02}\,P_b + p_{01} \qquad (139)\end{aligned}$$

$$\begin{aligned}\Phi_{3b} = \;& -4\,p_{41}\,{A_{eq}}^3\,P_b - 4\,p_{40}\,{A_{eq}}^3 + 3\,p_{32}\,{A_{eq}}^2\,{P_b}^2 + 3\,p_{31}\,{A_{eq}}^2\,P_b + 3\,p_{30}\,{A_{eq}}^2 \\ & - 2\,p_{23}\,A_{eq}\,{P_b}^3 - 2\,p_{22}\,A_{eq}\,{P_b}^2 - 2\,p_{21}\,A_{eq}\,P_b - 2\,p_{20}\,A_{eq} \\ & + p_{14}\,{P_b}^4 + p_{13}\,{P_b}^3 + p_{12}\,{P_b}^2 + p_{11}\,P_b + p_{10} \qquad (140)\end{aligned}$$

C.1.1 Non-Dimensionalization & Scaling

The Jacobian derived using Taylor Series methods is general, but tends to lead to poorly conditioned numerical matrices, due largely to differences in units. Of course, a scaling could be applied, but given the number of interacting variables, the scaling will vary depending on the equilibrium point.

Instead, a non-dimensionalization is used to remove units from consideration and reduce the number of scaling factors to a minimal quantity. A simple scaling can then be used to obtain linearized models that are numerically sound.

To non-dimensionalize the equations associated with the equation $\dot{\vec{x}} = J_0\vec{x}$, with J_0 defined as above, each state and time must be replaced by a non-dimensional state, enforced by scaling factors ϕ, α, β, ϵ_a, ϵ_b, and ω: $t = \phi\tilde{t}$, $x_1 = \alpha\tilde{x}_1$, $x_2 = \beta\tilde{x}_2$, $\tilde{P}_a = \epsilon_a P_a$, $\tilde{P}_b = \epsilon_b P_b$, and $\tilde{u} = \omega u$, where the non-dimensionalized state is indicated by the tilde. Since the scaling is initially set to affect space and time, the derivatives are of course affected as well, e.g., $\frac{d}{dt}(x_1) = \frac{d}{d\phi\tilde{t}}(\alpha x_1) = \frac{\alpha}{\phi}\frac{d}{d\tilde{t}}\tilde{x}_1$, such that:

$$\frac{d}{d\tilde{t}}\begin{bmatrix}\tilde{x}_1\\ \tilde{x}_2\\ \tilde{P}_a\\ \tilde{P}_b\\ \tilde{u}\end{bmatrix} = \begin{bmatrix}\phi/\alpha & 0 & 0 & 0\\ 0 & \phi/\beta & 0 & 0\\ 0 & 0 & \phi/\epsilon_a & 0\\ 0 & 0 & 0 & \phi/\epsilon_b\end{bmatrix} J_0 \begin{bmatrix}\alpha\tilde{x}_1\\ \beta\tilde{x}_2\\ \epsilon_a\tilde{P}_a\\ \epsilon_b\tilde{P}_b\\ \omega\tilde{u}\end{bmatrix} \tag{141}$$

The scaling parameters are then chosen to result in non-dimensionalized states. In the absence of any desired time scaling, $\phi = 1$. It is clear from equation (133) that the terms pre-multiplying pressure are the most complex and are generally scaled by chamber length (or volume, by extension). Therefore, a logical step is to choose ϵ_a and ϵ_b to normalize one of the pressure terms, so we set $J_{3,1} = 1$ and $J_{4,1} = 1$:

$$\epsilon_a = \frac{\alpha\,\phi\,(A_a\,P_{a,0}\,x_{2,0} - R\,T_{air}\,\dot{m}_{a,0})}{A_a\,(L_0 + ds_a + x_{1,0})^2} \tag{142}$$

$$\epsilon_b = \frac{\alpha\,\phi\,(A_b\,P_{b,0}\,x_{2,0} + R\,T_{air}\,\dot{m}_{b,0})}{A_b\,(L_0 + ds_b - x_{1,0})^2} \tag{143}$$

Then β and ω may be selected as well:

$$\beta = \left(\frac{\kappa P_0}{L_0}\right)_{Norm}\epsilon_b \tag{144}$$

$$\omega = A_b L_0/(RT_{air}\Phi_{3a})\epsilon_b/\phi; \tag{145}$$

where P_0 is chosen to be the average pressure, and L_0 is the stroke, κ is a scaling term that can be used to better condition the matrix. α is cancelled out, and in the absence of time scaling, $\phi = 1$, resulting in a non-dimensionalized Jacobian.

A convenient, broadly effective choice for κ is the value that minimizes the difference in orders of magnitude between terms that are products and quotients of the scaling κ.

In principle, this is just a change-of-variable or diagonal scaling, and the latter can be performed a number of ways (e.g., in MATLAB, using `balance()`). However, the non-dimensionalization and scaling is an approach that is physically meaningful and that simplifies the scaling to a single factor κ. The term can be chosen to minimize the difference in orders of magnitude between terms that are products and quotients of the scaling term – a method that is fairly effective. So if the smallest term that is a product of κ has magnitude a_1 and the largest term divided by *kappa* has magnitude a_2, then the scaling term should be chosen to be $\kappa = \sqrt{a_2/a_1}$ or a close integer value, if nicer constants are desired. Since the majority of remaining terms are scaled by κ, this approach is effective at creating well-conditioned linearized matrices that retain physical meaning without requiring broadly different scalings for any unique equilibrium state. A sweep of variable ranges in MATLAB demonstrated that using a range of possible variables showed that the non-dimensionalized system with the κ scaling based on bringing orders of magnitude together consistently reduced the condition number by several orders of magnitude, as desired.

APPENDIX D

CRR TWO LINK KINEMATICS & DYNAMICS

The details of the kinematics and dynamics derivation are reviewed here. The actual calculation was produced using a MATLAB script, provided in section D.3.1.

The model was derived using an analytical mechanics approach. It was also designed in Simulink to determine simulated results.

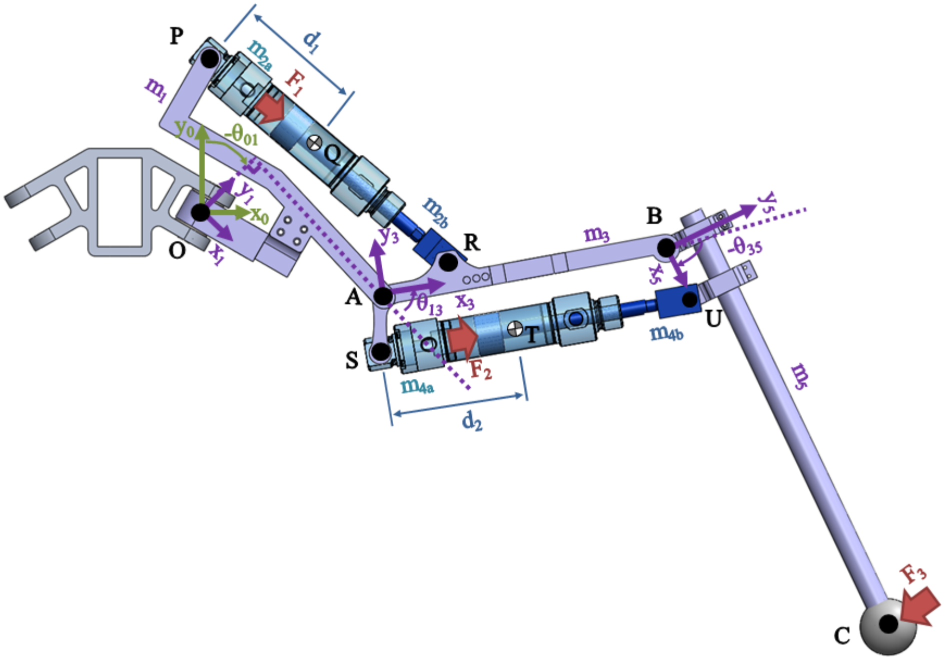

Figure 78: 2 DoF system overview.

Figure 78 shows the basic system layout and labeling scheme. There are seven bodies and five body-fixed reference frames. The bodies are numbered, with each actuator split into two parts: part (a), the cylinder, and part (b), the piston. Since the pneumatic actuator cylinders and pistons operate along a common axis, they share a reference frame.

Angles are denominated with Greek letters: θ is used to refer to coordinates, and corresponds to the link rotations, and ϕ represents the rotation of the pneumatic actuator.

These angles are listed in the form θ_{ij}, referring to the angle from frame i to frame j. For angles used in derivations: α is used in derivations to refer to variable angles, and γ refers to constant angles with known values. Angles in derivations are specific to the local geometry and include a corresponding subscript.

Points are labeled with capital letters, and distances are specified with l_{IJ}, referring to the length from point I to point J. For clarity, since the distances l_{PQ} and l_{ST} vary, they are instead referred to by the variable distances d_1 and d_2. There are three forces, F_1, F_2, and F_3, which act at points Q, T, and V.

D.1 Coordinate Geometry

Two coordinate schemes could be applied. Either $q_1 = \theta_{13}$ and $q_2 = \theta_{35}$, or $q_1 = d_1$ and $q_2 = d_2$. Either way, it is necessary to determine the geometrical relations that connect the two.

- **ϕ_{12} and θ_{13} can be written as functions of d_1:** The law of cosines can be used to relate these two variables. First, a triangle is drawn connecting the points P, A, and R, such that $l_{PR} = d_1 + l_{QR}$. Then the angle α_A can be found using the law of cosines:

  $$\alpha_A = \cos^{-1}\left(\frac{l_{PA}^2 + l_{AR}^2 - l_{PR}^2}{2l_{PA}l_{AR}}\right) \tag{146}$$

  Finally, θ_A just the result of a difference:

  $$\begin{aligned}\theta_{13} &= \pi - (\gamma_{A1} + \gamma_{A2}) - \alpha_A \\ &= \pi - (\gamma_{A1} + \gamma_{A2}) - \cos^{-1}\left(\frac{l_{PA}^2 + l_{AR}^2 - (l_{QR} + d_1)^2}{2l_{PA}l_{AR}}\right)\end{aligned} \tag{147}$$

  where the γ_{A1} and γ_{A2} are known constants.

  Similarly, ϕ_{12} is found using the Law of Cosines to solve for α_P, and recognizing that

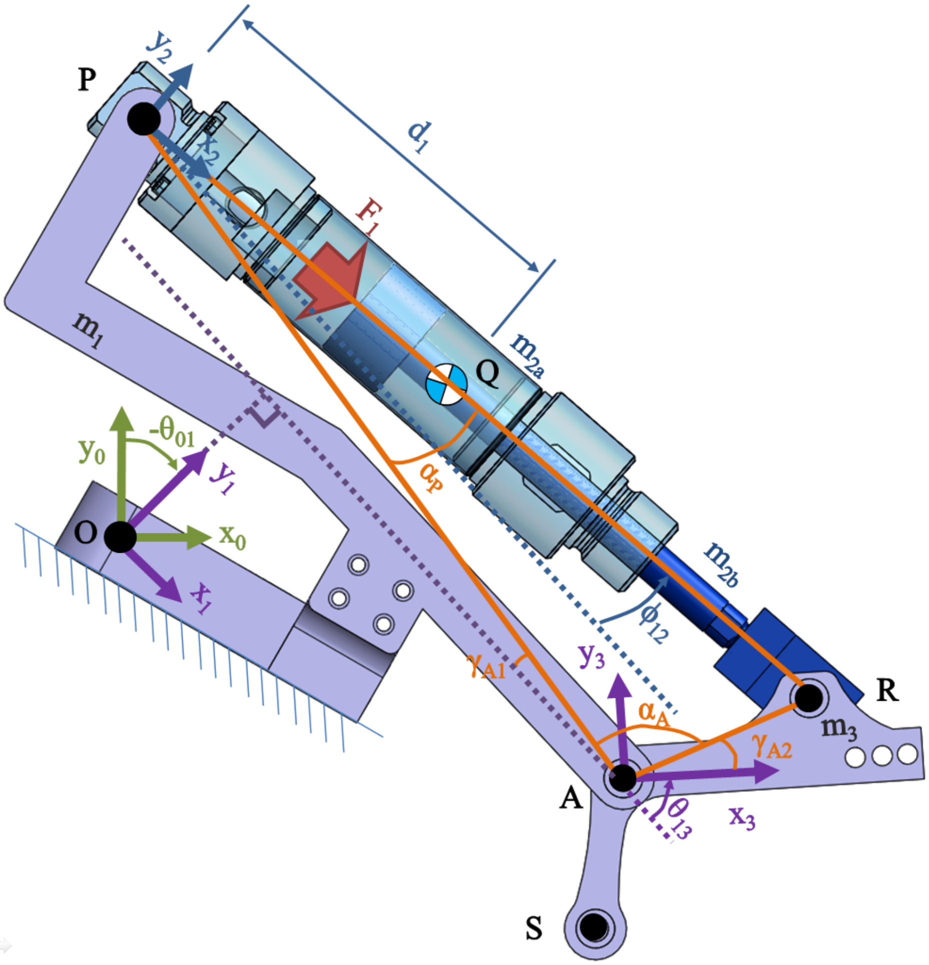

Figure 79: Geometric view to relate θ_{13} and d_1.

$\alpha_P = \phi_{12} + \gamma_{A1}$:

$$\begin{aligned}
\phi_{12} &= \alpha_P - \gamma_{A1} \\
&= \cos^{-1}\left(\frac{l_{PA}^2 + l_{PR}^2 - l_{AR}^2}{2l_{PA}l_{PR}}\right) - \gamma_{A1} \\
&= \cos^{-1}\left(\frac{l_{PA}^2 + (d_1 + l_{QR})^2 - l_{AR}^2}{2l_{PA}(d_1 + l_{QR})}\right) - \gamma_{A1} \qquad (148)
\end{aligned}$$

- **d_1 and ϕ_{12} from θ_{13}:**

$$\alpha_A = \pi - \theta_{13} - (\gamma_{A1} + \gamma_{A2}) \qquad (149)$$

and

$$
\begin{aligned}
d_1 &= \sqrt{l_{PA}^2 + l_{AR}^2 - 2l_{PA}l_{AR}\cos\alpha_A} - l_{QR} \\
&= \sqrt{l_{PA}^2 + l_{AR}^2 + 2l_{PA}l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)} - l_{QR} \qquad (150)
\end{aligned}
$$

Note that $\cos(\pi + x) = -\cos(x)$ and $\cos -y = \cos x$, so I've made that arbitrary conversion throughout my cosine simplifications. To get ϕ_P as a function of θ_{13}, l_{PR} will be substituted into equation (148), using the form seen in equation (150):

$$
\begin{aligned}
\phi_{12} &= \cos^{-1}\left(\frac{l_{PA}^2 + \left(\sqrt{l_{PA}^2 + l_{AR}^2 - 2l_{PA}l_{AR}\cos\alpha_A}\right)^2 - l_{AR}^2}{2l_{PA}\sqrt{l_{PA}^2 + l_{AR}^2 - 2l_{PA}l_{AR}\cos\alpha_A}}\right) - \gamma_{A1} \\
&= \cos^{-1}\left(\frac{2l_{PA}^2 + 2l_{PA}l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)}{2l_{PA}\sqrt{l_{PA}^2 + l_{AR}^2 + 2l_{PA}l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)}}\right) - \gamma_{A1} \\
&= \cos^{-1}\left(\frac{l_{PA} + l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)}{\sqrt{l_{PA}^2 + l_{AR}^2 + 2l_{PA}l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)}}\right) - \gamma_{A1} \qquad (151)
\end{aligned}
$$

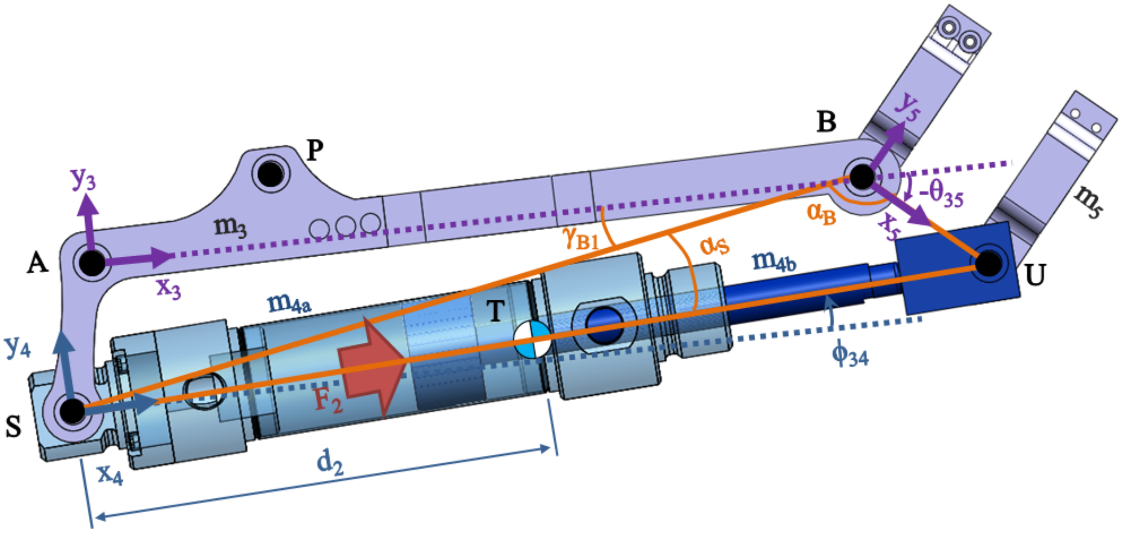

Figure 80: Geometric view of the middle link, relating θ_{35} and d_2.

- **θ_{35} and ϕ_{34} from d_2:** These relations are found in a similar manner to θ_A and d_1. A triangle is drawn connecting the points S, B, and U, with $l_{SU} = d_2 + l_{TU}$. Then the angle α_B can be found using the law of cosines:

$$\alpha_B = \cos^{-1}\left(\frac{l_{SB}^2 + l_{BU}^2 - l_{SU}^2}{2l_{SB}l_{BU}}\right) \tag{152}$$

Finally, θ_{35} is found using a difference calculation. The format differs slightly from equation (147) because of the sign convention used for θ_{13}.

$$\begin{aligned}\theta_{35} &= -(\pi - \gamma_B) - \alpha_B \\ &= -\pi + \gamma_B + \cos^{-1}\left(\frac{l_{SB}^2 + l_{BU}^2 - l_{SU}^2}{2l_{SB}l_{BU}}\right) \\ &= -\pi + \gamma_B + \cos^{-1}\left(\frac{l_{SB}^2 + l_{BU}^2 - (d_2 + l_{TU})^2}{2l_{SB}l_{BU}}\right)\end{aligned} \tag{153}$$

where γ_B is a known constant angle.

ϕ_{34} is also found using the Law of Cosines, via the relationship $\phi_{34} + \alpha_S = \gamma_B$:

$$\begin{aligned}\phi_{34} = \gamma_B - \alpha_S &= \gamma_B - \cos^{-1}\left(\frac{l_{SB}^2 + l_{SU}^2 - l_{BU}^2}{2l_{SB}l_{SU}}\right) \\ &= \gamma_B - \cos^{-1}\left(\frac{l_{SB}^2 + (d_2 + l_{TU})^2 - l_{BU}^2}{2l_{SB}(d_2 + l_{TU})}\right)\end{aligned} \tag{154}$$

- **d_2 and ϕ_{34} from θ_{35}:**

$$\alpha_B = \pi + \theta_{35} - \gamma_B \tag{155}$$

and

$$\begin{aligned}d_2 &= \sqrt{l_{SB}^2 + l_{BU}^2 - 2l_{SB}l_{BU}\cos\alpha_B} - l_{TU} \\ &= \sqrt{l_{SB}^2 + l_{BU}^2 + 2l_{SB}l_{BU}\cos(\theta_{35} - \gamma_B)} - l_{TU}\end{aligned} \tag{156}$$

ϕ_{34} is found by substituting $l_{A'Q}$ from (156) into equation (154):

$$\begin{aligned}\phi_{34} &= \gamma_B - \cos^{-1}\left(\frac{l_{SB}^2 + (l_{SU})^2 - l_{BU}^2}{2l_{SB}l_{SU}}\right) \\ &= \gamma_B - \cos^{-1}\left(\frac{l_{SB}^2 + l_{SB}^2 + l_{BU}^2 + 2l_{SB}l_{BU}\cos(\theta_{35} - \gamma_B) - l_{BU}^2}{2l_{SB}\sqrt{l_{SB}^2 + l_{BU}^2 + 2l_{SB}l_{BU}\cos(\theta_{35} - \gamma_B)}}\right) \\ &= \gamma_B - \cos^{-1}\left(\frac{l_{SB} + l_{BU}\cos(\theta_{35} - \gamma_B)}{\sqrt{l_{SB}^2 + l_{BU}^2 + l_{SB}l_{BU}\cos(\theta_{35} - \gamma_B)}}\right)\end{aligned} \tag{157}$$

Equations (158) to (165) summarize critical coordinate geometry relations.

$$\theta_{13} = \pi - (\gamma_{A1} + \gamma_{A2}) - \cos^{-1}\left(\frac{l_{PA}^2 + l_{AR}^2 - (l_{QR} + d_1)^2}{2l_{PA}l_{AR}}\right) \tag{158}$$

$$\theta_{35} = -\pi + \gamma_B + \cos^{-1}\left(\frac{l_{SB}^2 + l_{BU}^2 - (d_2 + l_{TU})^2}{2l_{SB}l_{BU}}\right) \tag{159}$$

$$d_1 = \sqrt{l_{PA}^2 + l_{AR}^2 + 2l_{PA}l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)} - l_{QR} \tag{160}$$

$$d_2 = \sqrt{l_{SB}^2 + l_{BU}^2 + 2l_{SB}l_{BU}\cos\left(\theta_{35} - \gamma_B\right)} - l_{TU} \tag{161}$$

$$\phi_{12} = \cos^{-1}\left(\frac{l_{PA}^2 + (d_1 + l_{QR})^2 - l_{AR}^2}{2l_{PA}(d_1 + l_{QR})}\right) - \gamma_{A1} \tag{162}$$

$$\phi_{12} = \cos^{-1}\left(\frac{l_{PA} + l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)}{\sqrt{l_{PA}^2 + l_{AR}^2 + 2l_{PA}l_{AR}\cos\left(\theta_{13} + (\gamma_{A1} + \gamma_{A2})\right)}}\right) - \gamma_{A1} \tag{163}$$

$$\phi_{34} = \gamma_B - \cos^{-1}\left(\frac{l_{SB}^2 + (d_2 + l_{TU})^2 - l_{BU}^2}{2l_{SB}(d_2 + l_{TU})}\right) \tag{164}$$

$$\phi_{34} = \gamma_B - \cos^{-1}\left(\frac{l_{SB} + l_{BU}\cos\left(\theta_{35} - \gamma_B\right)}{\sqrt{l_{SB}^2 + l_{BU}^2 + l_{SB}l_{BU}\cos\left(\theta_{35} - \gamma_B\right)}}\right) \tag{165}$$

From these, velocity relationships are obtained by differentiating with respect to time (done in MATLAB).

$$\dot{\theta}_{13} = -\frac{\dot{d}_1\ (l_{QR} + d_1)}{l_{AR}\,l_{PA}\sqrt{1 - \frac{\left({l_{AR}}^2 - {\left(l_{QR} + d_1\right)}^2 + {l_{PA}}^2\right)^2}{4\,{l_{AR}}^2\,{l_{PA}}^2}}} \tag{166}$$

$$\dot{\theta}_{35} = \frac{\dot{d}_2\ (l_{TU} + d_2)}{l_{BU}\,l_{SB}\sqrt{1 - \frac{\left({l_{BU}}^2 - (l_{TU} + d_2)^2 + {l_{SB}}^2\right)^2}{4\,{l_{BU}}^2\,{l_{SB}}^2}}} \tag{167}$$

$$\dot{d}_1 = -\frac{l_{AR}\,l_{PA}\,\dot{\theta}_{13}\ \sin(\gamma_{A1} + \gamma_{A2} + \theta_{13})}{\sqrt{{l_{AR}}^2 + 2\ \cos(\gamma_{A1} + \gamma_{A2} + \theta_{13})\ l_{AR}\,l_{PA} + {l_{PA}}^2}} \tag{168}$$

$$\dot{d}_2 = -\frac{l_{BU}\,l_{SB}\,\dot{\theta}_{35}\ \sin(\theta_{35} - \gamma_B)}{\sqrt{{l_{BU}}^2 + 2\ \cos(\theta_{35} - \gamma_B)\ l_{BU}\,l_{SB} + {l_{SB}}^2}} \tag{169}$$

$$\tag{170}$$

$$\dot{\phi}_{12} = -\frac{\frac{\dot{d}_1}{l_{PA}} - \frac{\dot{d}_1\left(\left(l_{QR}+d_1\right)^2 - {l_{AR}}^2 + {l_{PA}}^2\right)}{2\,l_{PA}\left(l_{QR}+d_1\right)^2}}{\sqrt{1 - \frac{\left(\left(l_{QR}+d_1\right)^2 - {l_{AR}}^2 + {l_{PA}}^2\right)^2}{4\,{l_{PA}}^2\left(l_{QR}+d_1\right)^2}}} \tag{171}$$

$$\dot{\phi}_{12} = \frac{\dot{\theta}_{13}\,\sin(\Gamma_{sum})\,{l_{AR}}^3 + l_{PA}\,\dot{\theta}_{13}\,\cos(\Gamma_{sum})\,\sin(\Gamma_{sum})\,{l_{AR}}^2}{\sqrt{-\frac{{l_{AR}}^2\cos(\Gamma_{sum})^2 - {l_{AR}}^2}{{l_{AR}}^2 + 2\,\cos(\Gamma_{sum})\,l_{AR}\,l_{PA} + {l_{PA}}^2}}\left({l_{AR}}^2 + 2\,\cos(\Gamma_{sum})\,l_{AR}\,l_{PA} + {l_{PA}}^2\right)^{\frac{3}{2}}}$$

$$\text{where} \quad \Gamma_{sum} = \gamma_{A1} + \gamma_{A2} + \theta_{13} \tag{172}$$

$$\dot{\phi}_{34} = \frac{\frac{\dot{d}_2}{l_{SB}} - \frac{\dot{d}_2\left(\left(l_{TU}+d_2\right)^2 - {l_{BU}}^2 + {l_{SB}}^2\right)}{2\,l_{SB}\left(l_{TU}+d_2\right)^2}}{\sqrt{1 - \frac{\left(\left(l_{TU}+d_2\right)^2 - {l_{BU}}^2 + {l_{SB}}^2\right)^2}{4\,{l_{SB}}^2\left(l_{TU}+d_2\right)^2}}} \tag{173}$$

$$\dot{\phi}_{34} = -\frac{\dot{\theta}_{35}\,\sin(\theta_{35}-\gamma_B)\,{l_{BU}}^3 + l_{SB}\,\dot{\theta}_{35}\,\cos(\theta_{35}-\gamma_B)\,\sin(\theta_{35}-\gamma_B)\,{l_{BU}}^2}{\sqrt{\frac{{l_{BU}}^2 - {l_{BU}}^2\cos(\theta_{35}-\gamma_B)^2}{{l_{BU}}^2 + 2\,\cos(\theta_{35}-\gamma_B)\,l_{BU}\,l_{SB} + {l_{SB}}^2}}\left({l_{BU}}^2 + 2\,\cos(\theta_{35}-\gamma_B)\,l_{BU}\,l_{SB} + {l_{SB}}^2\right)^{\frac{3}{2}}} \tag{174}$$

D.2 Body Properties

Mass and inertial properties of the moving components were determined using SolidWorks:

$$\begin{aligned} &m_1 = 0.8570 \text{ kg} \quad && m_{2a} = 0.3250 \text{ kg} \quad && m_{2b} = 0.0822 \text{ kg} \\ &m_3 = 0.6780 \text{ kg} && m_{4a} = 0.3250 \text{ kg} && m_{4b} = 0.0987 \text{ kg} \\ &m_5 = 0.2415 \text{ kg} \end{aligned} \tag{175}$$

Since the two degree of freedom system is planar, only the $I_z z$ inertial terms are necessary:

$$\begin{aligned} &(I_{2a})_P = 2.6\cdot 10^{-3} \text{ kg m}^2 \quad && (I_{2b})_Q = 2.7\cdot 10^{-4} \text{ kg m}^2 \\ &(I_3)_{G_3} = 2.0\cdot 10^{-3} \text{ kg m}^2 \\ &(I_{4a})_{G_{4a}} = 5.3\cdot 10^{-3} \text{ kg m}^2 && (I_{4b})_Q = 3.3\cdot 10^{-4} \text{ kg m}^2 \\ &(I_5)_{G_5} = 2.7\cdot 10^{-3} \text{ kg m}^2 \end{aligned} \tag{176}$$

D.3 Lagrange's Method

Since there are a number of connecting points and some complicated geometry, and since reaction forces are not of interested, an analytical mechanics approach will be used to obtain the equations of motion. These will be found using the generalized coordinates suggested above, by solving Lagrange's equation:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_j}\right) - \frac{\partial L}{\partial q_j} = Q_j^{(a)} + \sum_{i=1}^{J} a_{ij}\lambda_i \quad j = 1, 2, ...N \tag{177}$$

where $L = T - V$ is the Lagrangian. For the two-DoF arm, there are no constraints (geometric constraints are inherently enforced), so $a_{ij} = 0 \; \forall j$ and the remaining quantities are as follows:

- **Kinetic Energy terms, T:** There are 7 kinetic energy terms, corresponding to the seven moving bodies (2 cylinders, 2 pistons, two links, and the end-point mass):

$$T_{2a} = \frac{1}{2}(I_{2a})_P \omega_2^2 \tag{178}$$

$$T_{2b} = \frac{1}{2}(I_{2b})_{G_{2b}} \omega_2^2 + \frac{1}{2} m_{2b} v_{G_{2b}}^2 \tag{179}$$

$$T_3 = \frac{1}{2}(I_3)_{G_3} \omega_3^2 + \frac{1}{2} m_3 v_{G_3}^2 \tag{180}$$

$$T_{4a} = \frac{1}{2}(I_{4a})_{G_{4a}} \omega_4^2 + \frac{1}{2} m_{4a} v_{G_{4a}}^2 \tag{181}$$

$$T_{4b} = \frac{1}{2}(I_{4b})_{G_{4b}} \omega_4^2 + \frac{1}{2} m_{4b} v_{G_{4b}}^2 \tag{182}$$

$$T_5 = \frac{1}{2}(I_5)_{G_5} \omega_5^2 + \frac{1}{2} m_5 v_{G_5}^2 \tag{183}$$

with angular velocities

$$\vec{\omega}_2 = \dot{\phi}_{12}\hat{k} \tag{184}$$

$$\vec{\omega}_3 = \dot{\theta}_{13}\hat{k} \tag{185}$$

$$\vec{\omega}_4 = (\dot{\theta}_{13} + \dot{\phi}_{34})\hat{k} \tag{186}$$

$$\vec{\omega}_5 = (\dot{\theta}_{13} + \dot{\theta}_{35})\hat{k} \tag{187}$$

and velocities:

$$\vec{v}_{G_{2b}} = \vec{v}_Q = \omega_2 \times \vec{r}_{PG_{2b}} + \vec{v}_{rel} = \dot{\phi}_{12}\hat{k} \times d_1\hat{i}_2 + \dot{d}_1\hat{i}_2 = \dot{d}_1\hat{i}_2 + d_1\dot{\phi}_{12}\hat{j}_2 \tag{188}$$

$$\vec{v}_A = 0 \tag{189}$$

$$\begin{aligned}\vec{v}_{G_3} &= \vec{v}_A + \vec{\omega}_3 \times \vec{r}_{AG_3} \\ &= \dot{\theta}_{13}\hat{k} \times \Big((l_{AG_3})_{x_3}\hat{i}_3 + (l_{AG_3})_{y_3}\hat{j}_3\Big) \\ &= -(l_{AG_3})_{y_3}\dot{\theta}_{13}\hat{i}_3 + (l_{AG_3})_{x_3}\dot{\theta}_{13}\hat{j}_3\end{aligned} \tag{190}$$

$$\begin{aligned}\vec{v}_S &= \vec{v}_A + \vec{\omega}_3 \times \vec{r}_{AS} \\ &= \dot{\theta}_{13}\hat{k} \times ((l_{AS})_{x_3}\hat{i}_3 + (l_{AS})_{y_3}\hat{j}_3) \\ &= -(l_{AS})_{y_3}\dot{\theta}_{13}\hat{i}_3 + (l_{AS})_{x_3}\dot{\theta}_{13}\hat{j}_3\end{aligned} \tag{191}$$

$$\begin{aligned}\vec{v}_{G_{4a}} &= \vec{v}_S + \vec{\omega}_4 \times \vec{r}_{SG_{4a}} \\ &= -(l_{AS})_{y_3}\dot{\theta}_{13}(\cos\phi_{34}\hat{i}_4 - \sin\phi_{34}\hat{j}_4) + (l_{AS})_{x_3}\dot{\theta}_{13}(-\sin\phi_{34}\hat{i}_4 + \cos\phi_{34}\hat{j}_4) \\ &\quad + (\dot{\theta}_{13} + \dot{\phi}_{34})\hat{k} \times l_{SG_{4a}}\hat{i}_4 \\ &= \Big(-(l_{AS})_{y_3}\dot{\theta}_{13}\cos\phi_{34} + (l_{AS})_{x_3}\dot{\theta}_{13}\sin\phi_{34}\Big)\,\hat{i}_4 \\ &\quad + \Big((l_{AS})_{y_3}\dot{\theta}_{13}\sin\phi_{34} + (l_{AS})_{x_3}\dot{\theta}_{13}\cos\phi_{34} + l_{SG_{4a}}(\dot{\theta}_{13} + \dot{\phi}_{34})\Big)\,\hat{j}_4\end{aligned} \tag{192}$$

$$\begin{aligned}\vec{v}_{G_{4b}} = \vec{v}_T &= \vec{v}_S + \vec{\omega}_4 \times \vec{r}_{SG_{4b}} + \vec{v}_{rel} \\ &= \vec{v}_S + (\dot{\theta}_{13} + \dot{\phi}_{34})\hat{k} \times d_2\hat{i}_4 + \dot{d}_2\hat{i}_4 \\ &= \vec{v}_S + \dot{d}_2\hat{i}_4 + d_2(\dot{\theta}_{13} + \dot{\phi}_{34})\hat{j}_4 \\ &= \Big(-(l_{AS})_{y_3}\dot{\theta}_{13}\cos\phi_{34} + (l_{AS})_{x_3}\dot{\theta}_{13}\sin\phi_{34} + \dot{d}_2\Big)\,\hat{i}_4 \\ &\quad + \Big((l_{AS})_{y_3}\dot{\theta}_{13}\sin\phi_{34} + (l_{AS})_{x_3}\dot{\theta}_{13}\cos\phi_{34} + d_2(\dot{\theta}_{13} + \dot{\phi}_{34})\Big)\,\hat{j}_4\end{aligned} \tag{193}$$

$$\vec{v}_B = \vec{v}_A + \vec{\omega}_3 \times \vec{r}_{AB} = \dot{\theta}_{13}\hat{k} \times l_{AB}\hat{i}_3 = l_{AB}\dot{\theta}_{13}\hat{j}_3 \tag{194}$$

$$\begin{aligned}\vec{v}_{G_5} &= \vec{v}_B + \vec{\omega}_5 \times \vec{r}_{BG_5} \\ &= l_{AB}\dot{\theta}_{13}\hat{j}_3 + (\dot{\theta}_{13} + \dot{\theta}_{35})\hat{k} \times \Big((l_{BG_5})_{x5}\hat{i}_5 + (l_{BG_5})_{y5}\hat{j}_5\Big) \\ &= l_{AB}\dot{\theta}_{13}(\sin\theta_{35}\hat{i}_5 + \cos\theta_{35}\hat{j}_5) - (l_{BG_5})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\hat{i}_5 + (l_{BG_5})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\hat{j}_5 \\ &= \Big(l_{AB}\dot{\theta}_{13}\sin\theta_{35} - (l_{BG_5})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\Big)\,\hat{i}_5 \\ &\quad + \Big(l_{AB}\dot{\theta}_{13}\cos\theta_{35} + (l_{BG_5})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\Big)\,\hat{j}_5\end{aligned} \tag{195}$$

Additionally, the velocity of the foot (end effector), v_C is found so that it can be used to determine generalized forces if a force in some arbitrary direction, observed in the inertial frame, is applied to the end effector.

$$\begin{aligned}
v_C &= l_{AB}\dot{\theta}_{13}\hat{j}_3 - (l_{BC})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\hat{i}_5 + (l_{BC})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\hat{j}_5 \\
&= l_{AB}\dot{\theta}_{13}\Big(-\sin(\theta_{01} + \theta_{13})\hat{i}_0 + \cos(\theta_{01} + \theta_{13})\hat{j}_0\Big) \\
&\quad - (l_{BC})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\Big(\cos(\theta_{01} + \theta_{13} + \theta_{35})\hat{i}_0 + \sin(\theta_{01} + \theta_{13} + \theta_{35})\hat{j}_0\Big) \\
&\quad + (l_{BC})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\Big(-\sin(\theta_{01} + \theta_{13} + \theta_{35})\hat{i}_0 + \cos(\theta_{01} + \theta_{13} + \theta_{35})\hat{j}_0\Big) \\
&= -\Big(l_{AB}\dot{\theta}_{13}\sin(\theta_{01} + \theta_{13}) + (l_{BC})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\cos(\theta_{01} + \theta_{13} + \theta_{35}) \\
&\qquad + (l_{BC})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\sin(\theta_{01} + \theta_{13} + \theta_{35})\Big)\hat{i}_0 \\
&\quad + \Big(l_{AB}\dot{\theta}_{13}\cos(\theta_{01} + \theta_{13}) - (l_{BC})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\sin(\theta_{01} + \theta_{13} + \theta_{35}) \\
&\qquad + (l_{BC})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\cos(\theta_{01} + \theta_{13} + \theta_{35})\Big)\hat{j}_0 \qquad (196)
\end{aligned}$$

- **Potential Energy terms, V:** There are no springs, so potential energy terms only come from gravity. Note that the true robot is constrained to a lower bound for θ_{13} of approximately 20°, but that will be ignored. Further, since the heights of m_{2a} and m_{2b} don't change significantly, and since most of the load is carried by point P, the effects of gravity on those two bodies will be assumed negligible. While ϕ_{34} doesn't contribute nearly as significantly as θ_{13} to the gravitational potential of m_{4a} and m_{4b}, so those terms will be left out as well.

  The gravitational terms are identified in the following equation. The height of point A is used as the datum, since point A is fixed. The subscripts of the angle notation indicate start and end frame, e.g. $\theta_{03} = \theta_{01} + \theta_{13} = -\theta_{30}$ and indicates the angle that Frame 0 must be rotated by to get to frame 3 (so to view coordinates defined in frame 3 in frame 0 notation, they must be rotated by $\theta_{30} = -\theta_{03}$. Further, frames 2 and 4 require the inclusion of ϕ terms, e.g. $\theta_{02} = \theta_{01} + \phi_{12}$.

$$\begin{aligned}
V_{2a} &= m_{2a}g\left(\vec{r}_{PG_{2a}} \cdot \hat{j}_0\right) \\
&= m_{2a}g\left(l_{PG_{2a}}\hat{i}_2 \cdot \hat{j}_0\right) \\
&= m_{2a}gl_{PG_{2a}}\left(\cos(\theta_{20})\hat{i}_0 - \sin(\theta_{20})\hat{j}_0\right) \cdot \hat{j}_0 \\
&= m_{2a}gl_{PG_{2a}}\sin(\theta_{02}) \qquad (197) \\
V_{4b} &= m_{2b}g\left(\vec{r}_{PG_{2b}} \cdot \hat{j}_0\right) \\
&= m_{2b}g\left(l_{PG_{2b}}\hat{i}_2 \cdot \hat{j}_0\right) \\
&= m_{2b}gl_{PG_{2b}}\left(\cos(\theta_{20})\hat{i}_0 - \sin(\theta_{20})\hat{j}_0\right) \cdot \hat{j}_0 \\
&= m_{2b}gl_{PG_{2b}}\sin(\theta_{02}) \qquad (198)
\end{aligned}$$

$$\begin{aligned}
V_{4a} &= m_{4a}g\left(\vec{r}_{AG_{4a}} \cdot \hat{j}_0\right) = m_{4a}g\left(l_{AS_x}\hat{i}_3 + l_{AS_y}\hat{j}_3 + l_{SG_{4a}}\hat{i}_4\right) \cdot \hat{j}_0 \\
&= m_{4a}g\Big(l_{AS_x}(\cos(\theta_{30})\hat{i}_0 - \sin(\theta_{30})\hat{j}_0) + l_{AS_y}(\sin(\theta_{30})\hat{i}_0 + \cos(\theta_{30})\hat{j}_0) \\
&\quad + l_{SG_{4a}}(\cos(\theta_{40})\hat{i}_0 - \sin(\theta_{40})\hat{j}_0)\Big) \cdot \hat{j}_0 \\
&= m_{4a}g\left(l_{AS_x}\sin(\theta_{03}) + l_{AS_y}\cos(\theta_{03}) + l_{SG_{4a}}\sin(\theta_{04})\right) \qquad (199) \\
V_{4b} &= m_{4b}g\left(\vec{r}_{AG_{4b}} \cdot \hat{j}_0\right) \\
&= m_{4b}g\left(l_{AS_x}\sin(\theta_{03}) + l_{AS_y}\cos(\theta_{03}) + d_2\sin(\theta_{04})\right) \qquad (200)
\end{aligned}$$

$$\begin{aligned}
V_3 &= m_3 g\left(\vec{r}_{AG_3}\cdot\hat{j}_0\right) = m_3 g\left((l_{AG_3})_x\hat{i}_3 + (l_{AG_3})_y\hat{j}_3\right)\cdot\hat{j}_0\\
&= m_3 g\left((l_{AG_3})_x(\cos(\theta_{30})\hat{i}_0 - \sin(\theta_{30})\hat{j}_0) + (l_{AG_3})_y(\sin(\theta_{30})\hat{i}_0 + \cos(\theta_{30})\hat{j}_0)\right)\cdot\hat{j}_0
\end{aligned}$$

$$= m_3 g\,((l_{AG_3})_x\sin(\theta_{03}) + (l_{AG_3})_y\cos(\theta_{03})) \tag{201}$$

$$\begin{aligned}
V_5 &= m_5 g\left(\vec{r}_{AG_5}\cdot\hat{j}_0\right)\\
&= m_5 g\left(l_{AB}\hat{i}_3 + (l_{BG_5})_x\hat{i}_5 + (l_{BG_5})_y\hat{j}_5\right)\cdot\hat{j}_0\\
&= m_5 g\Big(l_{AB}(\cos(\theta_{30})\hat{i}_0 - \sin(\theta_{30})\hat{j}_0)\\
&\quad + (l_{BG_5})_x(\cos(\theta_{50})\hat{i}_0 - \sin(\theta_{50})\hat{j}_0) + (l_{BG_5})_y(\sin(\theta_{50})\hat{i}_0 + \cos(\theta_{50})\hat{j}_0)\Big)\cdot\hat{j}_0\\
&= m_5 g\,[-l_{AB}\sin(-\theta_{03})\ - (l_{BG_5})_x\sin(-\theta_{05}) + (l_{BG_5})_y\cos(-\theta_{05})]
\end{aligned}$$

$$= m_5 g\,[l_{AB}\sin(\theta_{03})\ + (l_{BG_5})_x\sin(\theta_{05}) + (l_{BG_5})_y\cos(\theta_{05})] \tag{202}$$

- **Generalized Forces, Q:** To get generalized forces, it is necessary to calculate the virtual work performed by those forces, which requires finding virtual displacements at the point of action. These can be found using $v_{G_{2b}} = v_Q$, $v_{G_{4b}} = v_T$, and v_{G_C}.

  The virtual displacements may be found by replacing velocity variables with virtual δ terms. This was done in MATLAB. Generalized forces were found by first finding the *virtual work* – the dot product of virtual displacements with the corresponding force vector at its point of action – and then finding the coefficients of each generalized coordinate (so Q_i corresponds to the coefficient of δq_i that results from the virtual work δW).

D.3.1 MATLAB derivation

```
%% Overview
% This script computes the equations of motion using Lagrangian Dynamics

% Methods for time derivatives:
% (a) construct expressions with time-free variables
% (b) Substitute Time Dependent Variables
% (c) Differentiate
% (d) Substitute Time Free Variables
% (e) Move to next step

%General rule: At the start and end of each section, everything should be
%written in time-free variables

%% Prepare workspace
clear all;
close all;
clc;

%% Define variables
%Note that lowercase 'p' is used to denote 'prime'
disp('Defining Variables');
% Time
syms t real

% Constants
syms gamma_A1 gamma_A2 gamma_B theta_01 real
syms l_PA l_AR l_QR real
syms l_AS l_AB l_SB l_BU l_ST l_TU real
syms l_BV l_BC real
syms l_AG3 l_BG5
syms l_AG3x l_AG3y l_BG5x l_BG5y l_BCx l_BCy l_ASx l_ASy real
syms l_PG2a l_SG4a real

syms I0_1 I0_2a IG_2b IG_3 IG_4a IG_4b IG_5 real
syms m_1 m_2a m_2b m_3 m_4a m_4b m_5 real

% Forces
syms g F_1 F_2 F_3_x F_3_y real

%Virtual work and related variables
```

```
syms DEL_d_1 DEL_d_2 DEL_theta_13 DEL_theta_35 real
syms DEL_W_d Q_1_d Q_2_d real
syms DEL_W_th Q_1_th Q_2_th real

%Just the symbols
syms theta_01
syms theta_13 theta_13_DOT th_13_DDOT
syms theta_35 theta_35_DOT th_35_DDOT
syms d_1 d_1_DOT d_1_DDOT
syms d_2 d_2_DOT d_2_DDOT
%No phi symbols -- these are just working variables

%Time dependent functions
syms theta_13_t(t) theta_13_DOT_t(t) th_13_DDOT_t(t)
syms theta_35_t(t) theta_35_DOT_t(t) th_35_DDOT_t(t)
syms d_1_t(t) d_1_DOT_t(t) d_1_DDOT_t(t)
syms d_2_t(t) d_2_DOT_t(t) d_2_DDOT_t(t)

%Single Derivative expressions
dt13dt = diff(theta_13_t(t),t); dt13DOTdt = diff(theta_13_DOT_t(t),t);
dt35dt = diff(theta_35_t(t),t); dt35DOTdt = diff(theta_35_DOT_t(t),t);
dd1dt = diff(d_1_t(t), t); dd1DOTdt = diff(d_1_DOT_t(t),t);
dd2dt = diff(d_2_t(t), t); dd2DOTdt = diff(d_2_DOT_t(t),t);
% dp12dt = diff(phi_12, t);
% dp34dt = diff(phi_34, t);

%Double Derivative expressions. These shouldn't exist if we always convert
%back after each segment (we only take one derivative at a time).
% d_dt13dt = diff(dt13dt,t);
% d_dt35dt = diff(dt35dt,t);
% d_dd1dt = diff(dd1dt,t);
% d_dd2dt = diff(dd2dt,t);

%For converting to time-dependent form:
before_ddt_noT_vars = [theta_13 theta_13_DOT th_13_DDOT ...
```

```
    theta_35 theta_35_DOT th_35_DDOT...
    d_1 d_1_DOT d_1_DDOT...
    d_2 d_2_DOT d_2_DDOT];
before_ddt_T_vars = [theta_13_t theta_13_DOT_t th_13_DDOT_t ...
    theta_35_t theta_35_DOT_t th_35_DDOT_t...
    d_1_t d_1_DOT_t d_1_DDOT_t...
    d_2_t d_2_DOT_t d_2_DDOT_t];

%For converting to time-independent form
after_ddt_T_vars = [theta_13_t theta_13_DOT_t dt13dt dt13DOTdt ...
    theta_35_t theta_35_DOT_t dt35dt dt35DOTdt...
    d_1_t d_1_DOT_t dd1dt dd1DOTdt ...
    d_2_t d_2_DOT_t dd2dt dd2DOTdt];
after_ddt_noT_vars = [theta_13 theta_13_DOT theta_13_DOT th_13_DDOT...
    theta_35 theta_35_DOT theta_35_DOT th_35_DDOT...
    d_1 d_1_DOT d_1_DOT d_1_DDOT...
    d_2 d_2_DOT d_2_DOT d_2_DDOT];

%Angle Expressions
theta_03 = theta_01 + theta_13;
theta_05 = theta_01 + theta_13 + theta_35;



%% Define working Position variables
% These are the relationships needed to derive dynamics with just one set
% of generalized coordinates when things are often written as functions of
% each other.
disp('Defining theta/d/phi equivalences..');

%'_fd' indictates function of d1, d1;
%'_fth' indicates function of 'theta_A', 'theta_B'
theta_13_fd = pi - (gamma_A1 + gamma_A2) - acos((l_PA^2 ...
    + l_AR^2 - (d_1 + l_QR)^2)/(2*l_PA*l_AR));
theta_35_fd = -pi + gamma_B + acos((l_SB^2 + l_BU^2 ...
    - (d_2 + l_TU)^2)/(2*l_SB*l_BU));
```

```
d_1_fth = sqrt(l_PA^2 + l_AR^2 + 2*l_PA*l_AR*cos(theta_13 ...
    + (gamma_A1 + gamma_A2))) - l_QR;
d_2_fth = sqrt(l_SB^2 + l_BU^2 + 2*l_SB*l_BU*cos(theta_35 ...
    - gamma_B)) - l_TU;

phi_12_fd = acos((l_PA^2 + (d_1 + l_QR)^2 - l_AR^2)/(2*l_PA*(d_1 ...
    + l_QR))) - gamma_A1;
phi_12_fth = acos((l_PA + l_AR*cos(theta_13 + (gamma_A1 ...
    + gamma_A2)))/sqrt(l_PA^2 + l_AR^2 + 2*l_PA*l_AR*cos(theta_13 ...
    + (gamma_A1 + gamma_A2))));

phi_34_fd = gamma_B - acos((l_SB^2 + (d_2 + l_TU)^2 ...
    - l_BU^2)/(2*l_SB*(d_2 + l_TU)));
phi_34_fth = gamma_B - acos((l_SB + l_BU*cos(theta_35 ...
    - gamma_B))/sqrt(l_SB^2 + l_BU^2 ...
    + 2*l_SB*l_BU*cos(theta_35 - gamma_B)));

%% Find working Velocity variables
% Velocity equivalences are also convenient to have
disp('Defining theta-dot/phi-dot/d-dot equivalences..');

%For converting to time-independent form
% RECALL:
% Methods for time derivatives:
% (a) construct expressions with time-free variables
% (b) Substitute Time Dependent Variables
% (c) Differentiate
% (d) Substitute Time Free Variables
% (e) Move to next step

% (b) Substitute Time Dependent Variables
theta_13_fd_t = subs(theta_13_fd, before_ddt_noT_vars, before_ddt_T_vars);
theta_35_fd_t = subs(theta_35_fd, before_ddt_noT_vars, before_ddt_T_vars);
```

```
d_1_fth_t = subs(d_1_fth, before_ddt_noT_vars, before_ddt_T_vars);
d_2_fth_t = subs(d_2_fth, before_ddt_noT_vars, before_ddt_T_vars);

phi_12_fd_t = subs(phi_12_fd, before_ddt_noT_vars, before_ddt_T_vars);
phi_12_fth_t = subs(phi_12_fth, before_ddt_noT_vars, before_ddt_T_vars);

phi_34_fd_t = subs(phi_34_fd, before_ddt_noT_vars, before_ddt_T_vars);
phi_34_fth_t = subs(phi_34_fth, before_ddt_noT_vars, before_ddt_T_vars);

% (c) Differentiate
theta_13_DOT_fd_t = diff(theta_13_fd_t, t);
theta_35_DOT_fd_t = diff(theta_35_fd_t, t);

d_1_DOT_fth_t = diff(d_1_fth_t, t);
d_2_DOT_fth_t = diff(d_2_fth_t, t);

phi_12_DOT_fd_t = diff(phi_12_fd_t, t);
phi_12_DOT_fth_t = diff(phi_12_fth_t, t);

phi_34_DOT_fd_t = diff(phi_34_fd_t, t);
phi_34_DOT_fth_t = diff(phi_34_fth_t, t);

% (d) Substitute Time Free Variables
theta_13_DOT_fd = subs(theta_13_DOT_fd_t, after_ddt_T_vars, ...
    after_ddt_noT_vars);
theta_35_DOT_fd = subs(theta_35_DOT_fd_t, after_ddt_T_vars, ...
    after_ddt_noT_vars);

d_1_DOT_fth = subs(d_1_DOT_fth_t, after_ddt_T_vars, after_ddt_noT_vars);
d_2_DOT_fth = subs(d_2_DOT_fth_t, after_ddt_T_vars, after_ddt_noT_vars);

phi_12_DOT_fd = subs(phi_12_DOT_fd_t, after_ddt_T_vars, ...
    after_ddt_noT_vars);
phi_12_DOT_fth = subs(phi_12_DOT_fth_t, after_ddt_T_vars, ...
    after_ddt_noT_vars);
```

```
phi_34_DOT_fd = subs(phi_34_DOT_fd_t, after_ddt_T_vars, ...
    after_ddt_noT_vars);
phi_34_DOT_fth = subs(phi_34_DOT_fth_t, after_ddt_T_vars, ...
    after_ddt_noT_vars);

%% Write out velocity terms and take derivatives
disp('Defining key velocity terms...');


v_Q_d = [d_1_DOT, d_1*phi_12_DOT_fd];
v_T_d = [-l_ASy*theta_13_DOT_fd*cos(phi_34_fd) ...
    + l_ASx*theta_13_DOT_fd*sin(phi_34_fd) + d_2_DOT, ...
    l_ASy*theta_13_DOT_fd*sin(phi_34_fd) ...
    + l_ASx*theta_13_DOT_fd*cos(phi_34_fd) ...
    + d_2*(theta_13_DOT_fd + phi_34_DOT_fd)];
v_C_d = [-(l_AB*theta_13_DOT_fd*sin(theta_01 + theta_13_fd) ...
    + l_BCy*(theta_13_DOT_fd + theta_35_DOT_fd)*cos(theta_01 ...
    + theta_13_fd + theta_35_fd) ...
    + l_BCx*(theta_13_DOT_fd + theta_35_DOT_fd)*sin(theta_01 ...
    + theta_13_fd + theta_35_fd)), ...
    l_AB*theta_13_DOT_fd*cos(theta_01 + theta_13_fd) ...
    - l_BCy*(theta_13_DOT_fd + theta_35_DOT_fd)*sin(theta_01 ...
    + theta_13_fd + theta_35_fd) + l_BCx*(theta_13_DOT_fd ...
    + theta_35_DOT_fd)*cos(theta_01 + theta_13_fd + theta_35_fd)];

v_Q_th = [d_1_DOT_fth d_1_fth*phi_12_DOT_fth];
v_T_th = [-l_ASy*theta_13_DOT*cos(phi_34_fth) ...
    + l_ASx*theta_13_DOT*sin(phi_34_fth) + d_2_DOT_fth ...
    l_ASy*theta_13_DOT*sin(phi_34_fth) ...
    + l_ASx*theta_13_DOT*cos(phi_34_fth) + d_2_fth*(theta_13_DOT ...
    + phi_34_DOT_fth)];
v_C_th = [-(l_AB*theta_13_DOT*sin(theta_01 + theta_13) ...
    + l_BCy*(theta_13_DOT + theta_35_DOT)*cos(theta_01 ...
    + theta_13 + theta_35) + l_BCx*(theta_13_DOT ...
    + theta_35_DOT)*sin(theta_01 + theta_13 + theta_35)) ...
    l_AB*theta_13_DOT*cos(theta_01 + theta_13) ...
    - l_BCy*(theta_13_DOT + theta_35_DOT)*sin(theta_01 ...
    + theta_13 + theta_35) + l_BCx*(theta_13_DOT ...
    + theta_35_DOT)*cos(theta_01 + theta_13 + theta_35)];

%% Find virtual displacements
disp('Defining virtual displacements from velocity equations...');

%Keep matrix format intact for later construction of gen. forces
dr_Q_d_mat = subs(v_Q_d, [d_1_DOT d_2_DOT], [DEL_d_1 DEL_d_2]);
dr_T_d_mat = subs(v_T_d, [d_1_DOT d_2_DOT], [DEL_d_1 DEL_d_2]);
dr_C_d_mat = subs(v_C_d, [d_1_DOT d_2_DOT], [DEL_d_1 DEL_d_2]);

dr_Q_th_mat = subs(v_Q_th, [theta_13_DOT theta_35_DOT], ...
    [DEL_theta_13 DEL_theta_35]);
dr_T_th_mat = subs(v_T_th, [theta_13_DOT theta_35_DOT], ...
    [DEL_theta_13 DEL_theta_35]);
dr_C_th_mat = subs(v_C_th, [theta_13_DOT theta_35_DOT], ...
    [DEL_theta_13 DEL_theta_35]);

%% Create Virtual Work and Get General Forces
disp('Extracting generalized forces from virtual work...');
DEL_W_d = dr_Q_d_mat*[F_1; 0] + dr_T_d_mat*[F_2; 0] ...
    + dr_C_d_mat*[F_3_x; F_3_y];
DEL_W_th = dr_Q_th_mat*[F_1; 0] + dr_T_th_mat*[F_2; 0] ...
    + dr_C_th_mat*[F_3_x; F_3_y];

[coeffs_d, operators_d] = coeffs(DEL_W_d, [DEL_d_1 DEL_d_2]);
Q_1_d = coeffs_d(1);
Q_2_d = coeffs_d(2);

[coeffs_th, operators_th] = coeffs(DEL_W_th, [DEL_theta_13 DEL_theta_35]);
Q_1_th = coeffs_th(1);
Q_2_th = coeffs_th(2);
```

```
Q_sum_d = Q_1_d + Q_2_d;
Q_sum_th = Q_1_th + Q_2_th;

%% Add kinetic energy terms
disp('Defining velocity terms (T)...');
%I could sub (e.g., v_G2b = v_Q, but this seems more straightforward)

%In terms of d
v_G2b_d = [d_1_DOT, ...
    d_1*phi_12_DOT_fd];
v_A_d = [0 0];
v_G3_d = [-l_AG3y*theta_13_DOT_fd, ...
    l_AG3x*theta_13_DOT_fd];
v_S_d = [-l_ASy*theta_13_DOT_fd, ...
    l_ASx*theta_13_DOT_fd];
v_G4a_d = [-l_ASy*theta_13_DOT_fd*cos(phi_34_fd) ...
    + l_ASx*theta_13_DOT_fd*sin(phi_34_fd), ...
    l_ASy*theta_13_DOT_fd*sin(phi_34_fd) ...
    + l_ASx*theta_13_DOT_fd*cos(phi_34_fd) ...
    + l_SG4a*(theta_13_DOT_fd + phi_34_DOT_fd)];
v_G4b_d = [-l_ASy*theta_13_DOT_fd*cos(phi_34_fd) ...
    + l_ASx*theta_13_DOT_fd*sin(phi_34_fd) + d_2_DOT ...
    l_ASy*theta_13_DOT_fd*sin(phi_34_fd) ...
    + l_ASx*theta_13_DOT_fd*cos(phi_34_fd) ...
    + d_2*(theta_13_DOT_fd + phi_34_DOT_fd)];
v_B_d = [0, ...
    l_AB*theta_13_DOT_fd];
v_G5_d = [(l_AB*theta_13_DOT_fd*sin(theta_35_fd) ...
    - l_BG5y*(theta_13_DOT_fd + theta_35_DOT_fd)), ...
    l_AB*theta_13_DOT_fd*cos(theta_35_fd) + l_BG5x*(theta_13_DOT_fd ...
    + theta_35_DOT_fd)];

%In terms of theta
v_G2b_th = [d_1_DOT_fth ...
    d_1_fth*phi_12_DOT_fth];
```

```
v_A_th = [0 0];
v_G3_th = [0, ...
    l_AG3*theta_13_DOT];
v_S_th = [-l_ASy*theta_13_DOT, ...
    l_ASx*theta_13_DOT];
v_G4a_th = [-l_ASy*theta_13_DOT*cos(phi_34_fth) ...
    + l_ASx*theta_13_DOT*sin(phi_34_fth), ...
    l_ASy*theta_13_DOT*sin(phi_34_fth) ...
    + l_ASx*theta_13_DOT*cos(phi_34_fth) ...
    + l_SG4a*(theta_13_DOT + phi_34_DOT_fth)];
v_G4b_th = [-l_ASy*theta_13_DOT*cos(phi_34_fth) ...
    + l_ASx*theta_13_DOT*sin(phi_34_fth) + d_2_DOT_fth ...
    l_ASy*theta_13_DOT*sin(phi_34_fth) ...
    + l_ASx*theta_13_DOT*cos(phi_34_fth) ...
    + d_2_fth*(theta_13_DOT + phi_34_DOT_fth)];
v_B_th = [0, ...
    l_AB*theta_13_DOT];
v_G5_th = [(l_AB*theta_13_DOT*sin(theta_35) ...
    - l_BG5y*(theta_13_DOT + theta_35_DOT)), ...
    l_AB*theta_13_DOT*cos(theta_35)+ l_BG5x*(theta_13_DOT + theta_35_DOT)];

omega_d(2) = phi_12_DOT_fd;
omega_d(3) = theta_13_DOT_fd;
omega_d(4) = theta_13_DOT_fd + phi_34_DOT_fd;
omega_d(5) = theta_13_DOT_fd + theta_35_DOT_fd;

omega_th(2) = phi_12_DOT_fth;
omega_th(3) = theta_13_DOT;
omega_th(4) = theta_13_DOT + phi_34_DOT_fth;
omega_th(5) = theta_13_DOT + theta_35_DOT;

% Since dot product commands lead to issues with the conj() operator, the
% 'real' declaration wasn't working, I just wrote out the dot product.
% Previously I thought I couldn't select by entry, so I would post-multiply
% by a 2 x 1 selection vector (but I think that that's unnecessary).
```

```
v2_d(2,2) = v_G2b_d(1)^2 + v_G2b_d(2)^2;
v2_d(3,1) = v_G3_d(1)^2 + v_G3_d(2)^2;
v2_d(4,1) = v_G4a_d(1)^2 + v_G4a_d(2)^2;
v2_d(4,2) = v_G4b_d(1)^2 + v_G4b_d(2)^2;
v2_d(5,1) = v_G5_d(1)^2 + v_G5_d(2)^2;

v2_th(2,2) = v_G2b_th(1)^2 + v_G2b_th(2)^2;
v2_th(3,1) = v_G3_th(1)^2 + v_G3_th(2)^2;
v2_th(4,1) = v_G4a_th(1)^2 + v_G4a_th(2)^2;
v2_th(4,2) = v_G4b_th(1)^2 + v_G4b_th(2)^2;
v2_th(5,1) = v_G5_th(1)^2 + v_G5_th(2)^2;

%% Compute Lagrange's equation
disp('Defining kinetic energy terms (T) & potential energy terms (V)...');

%Do this systematically, using matrices that correspond to body ID
I_sym_mat = [I0_1 0; ...
    I0_2a IG_2b; ...
    IG_3 0;...
    IG_4a IG_4b; ...
    IG_5 0];

m_sym_mat = [m_1 0; ...
    m_2a m_2b; ...
    m_3 0;...
    m_4a m_4b; ...
    m_5 0];

%Kinetic Energy Terms
T_i = 2; T_j = 1;
    T_d(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_d(T_i)^2;
    T_th(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_th(T_i)^2;
T_i = 2; T_j = 2;
    T_d(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_d(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_d(T_i, T_j);
    T_th(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_th(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_th(T_i, T_j);
T_i = 3; T_j = 1;
    T_d(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_d(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_d(T_i, T_j);
    T_th(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_th(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_th(T_i, T_j);
T_i = 4; T_j = 1;
    T_d(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_d(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_d(T_i, T_j);
    T_th(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_th(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_th(T_i, T_j);
T_i = 4; T_j = 2;
    T_d(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_d(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_d(T_i, T_j);
    T_th(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_th(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_th(T_i, T_j);
T_i = 5; T_j = 1;
    T_d(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_d(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_d(T_i, T_j);
    T_th(T_i,T_j) = 1/2*I_sym_mat(T_i,T_j)*omega_th(T_i)^2 ...
        + 1/2*m_sym_mat(T_i,T_j)*v2_th(T_i, T_j);

T_sum_d = sum(sum(T_d));
T_sum_th = sum(sum(T_th));

%Potential Energy Terms
V_th(2,1) = m_sym_mat(2,1)*g...
    *(l_PG2a*sin(theta_01 + phi_12_fth));
V_th(2,2) = m_sym_mat(2,2)*g...
    *(d_1_fth*sin(theta_01 + phi_12_fth));
V_th(3,1) = m_sym_mat(3,1)*g...
    *(l_AG3x*sin(theta_01 + theta_13) + l_AG3y*cos(theta_01 + theta_13));
V_th(4,1) = m_sym_mat(4,1)*g...
    *(l_ASx*sin(theta_01 + theta_13) + l_ASy*cos(theta_01 + theta_13) ...
        + l_SG4a*sin(theta_01 + theta_13 + phi_34_fth));
V_th(4,2) = m_sym_mat(4,2)*g...
    *(l_ASx*sin(theta_01 + theta_13) + l_ASy*cos(theta_01 + theta_13) ...
        + d_2_fth*sin(theta_01 + theta_13 + phi_34_fth));
V_th(5,1) = m_sym_mat(5,1)*g...
    *(l_AB*sin(theta_01 + theta_13) + l_BG5x*sin(theta_01 + theta_13 ...
        + theta_35) + l_BG5y*cos(theta_01 + theta_13 + theta_35));

V_d(2,1) = subs(V_th(2,1), [theta_13 theta_35], [theta_13_fd theta_35_fd]);
V_d(2,2) = subs(V_th(2,2), [theta_13 theta_35], [theta_13_fd theta_35_fd]);
V_d(3,1) = subs(V_th(3,1), [theta_13 theta_35], [theta_13_fd theta_35_fd]);
V_d(4,1) = subs(V_th(4,1), [theta_13 theta_35], [theta_13_fd theta_35_fd]);
V_d(4,2) = subs(V_th(4,2), [theta_13 theta_35], [theta_13_fd theta_35_fd]);
V_d(5,1) = subs(V_th(5,1), [theta_13 theta_35], [theta_13_fd theta_35_fd]);

V_sum_d = sum(sum(V_d));
V_sum_th = sum(sum(V_th));

% Assemble Lagrangian
L_d = T_sum_d - V_sum_d;
L_th = T_sum_th - V_sum_th;

%% Find State Derivatives
disp('Finding dL/dq and dL/dqDOT...');

%Position State Derivatives
dL_th_dt13 = diff(L_th, theta_13);
dL_th_dt35 = diff(L_th, theta_35);
dL_d_dd1 = diff(L_d, d_1);
dL_d_dd2 = diff(L_d, d_2);

%Velocity State Derivatives
dL_th_dt13DOT = diff(L_th, theta_13_DOT);
dL_th_dt35DOT = diff(L_th, theta_35_DOT);
dL_d_dd1DOT = diff(L_d, d_1_DOT);
```

```
dL_d_dd2DOT = diff(L_d, d_2_DOT);

%% Find Time Derivatives
disp('Finding d(dL/dqDOT)/dt...');

%For converting to time-independent form
% RECALL:
% Methods for time derivatives:
% (a) construct expressions with time-free variables
% (b) Substitute Time Dependent Variables
% (c) Differentiate
% (d) Substitute Time Free Variables
% (e) Move to next step

% (b) Substitute Time Dependent Variables
dL_th_dt13DOT_t = subs(dL_th_dt13DOT, before_ddt_noT_vars, ...
        before_ddt_T_vars);
dL_th_dt35DOT_t = subs(dL_th_dt35DOT, before_ddt_noT_vars, ...
        before_ddt_T_vars);
dL_d_dd1DOT_t = subs(dL_d_dd1DOT, before_ddt_noT_vars, before_ddt_T_vars);
dL_d_dd2DOT_t = subs(dL_d_dd2DOT, before_ddt_noT_vars, before_ddt_T_vars);

% (c) Differentiate
d_dL_th_dt13DOT_dT_t = diff(dL_th_dt13DOT_t, t);
d_dL_th_dt35DOT_dT_t = diff(dL_th_dt35DOT_t, t);
d_dL_d_dd1DOT_dT_t = diff(dL_d_dd1DOT_t, t);
d_dL_d_dd2DOT_dT_t = diff(dL_d_dd2DOT_t, t);

% (d) Substitute Time Free Variables
d_dL_th_dt13DOT_dT = subs(d_dL_th_dt13DOT_dT_t, after_ddt_T_vars, ...
        after_ddt_noT_vars);
d_dL_th_dt35DOT_dT = subs(d_dL_th_dt35DOT_dT_t, after_ddt_T_vars, ...
        after_ddt_noT_vars);
d_dL_d_dd1DOT_dT = subs(d_dL_d_dd1DOT_dT_t, after_ddt_T_vars, ...
        after_ddt_noT_vars);
```

```
d_dL_d_dd2DOT_dT = subs(d_dL_d_dd2DOT_dT_t, after_ddt_T_vars, ...
        after_ddt_noT_vars);

%% Analyze Results
disp('Compiling Lagranges Equations...');

%Put together equation
eq_1_lhs_d = d_dL_d_dd1DOT_dT - dL_d_dd1;
eq_2_lhs_d = d_dL_d_dd2DOT_dT - dL_d_dd2;
eq_1_lhs_th = d_dL_th_dt13DOT_dT - dL_th_dt13;
eq_2_lhs_th = d_dL_th_dt35DOT_dT - dL_th_dt35;

%Generate expression (just for funsies, I guess)
eq_d_1 = eq_1_lhs_d == Q_1_d;
eq_d_2 = eq_2_lhs_d == Q_2_d;
eq_th_1 = eq_1_lhs_th == Q_1_th;
eq_th_2 = eq_2_lhs_th == Q_2_th;

%% Produce versions that are easily implemented in Simulink
% Sort into form
% C1*ddotth13 + C2*ddotth35 + C3 = Q1
% and
% D1*ddotth13 + D2*ddotth35 + D3 = Q2

[C_th, operators_C_th] = coeffs(eq_1_lhs_th, [th_13_DDOT th_35_DDOT]);
[D_th, operators_D_th] = coeffs(eq_2_lhs_th, [th_13_DDOT th_35_DDOT]);

[C_d, operators_C_d] = coeffs(eq_1_lhs_d, [d_1_DDOT d_2_DDOT]);
[D_d, operators_D_d] = coeffs(eq_2_lhs_d, [d_1_DDOT d_2_DDOT]);

%% Compute linearization
disp('Defining linearized equations...');
C_1 = C_th(1); C_2 = C_th(2); C_3 = C_th(3);
D_1 = D_th(1); D_2 = D_th(2); D_3 = D_th(3);
```

```
inv_Amat = [D_2/(C_1*D_2 - C_2*D_1), -C_2/(C_1*D_2 - C_2*D_1); ...
    -D_1/(C_1*D_2 - C_2*D_1),  C_1/(C_1*D_2 - C_2*D_1)];
Bmat = [Q_1_th - C_3; Q_2_th - D_3];
F_nonlin = inv_Amat*Bmat;

J = [diff(F_nonlin(1), theta_13), diff(F_nonlin(1), theta_13_DOT), ...
        diff(F_nonlin(1), theta_35), diff(F_nonlin(1), theta_35_DOT), ...
        diff(F_nonlin(1), F_1), diff(F_nonlin(1), F_2), ...
        diff(F_nonlin(1), F_3_x), diff(F_nonlin(1), F_3_y); ...
     diff(F_nonlin(2), theta_13), diff(F_nonlin(2), theta_13_DOT), ...
        diff(F_nonlin(2), theta_35), diff(F_nonlin(2), theta_35_DOT), ...
        diff(F_nonlin(2), F_1), diff(F_nonlin(2), F_2), ...
        diff(F_nonlin(2), F_3_x), diff(F_nonlin(2), F_3_y)...
    ];
A = [0 1 0 0; J(1,1:4);  0 0 0 1; J(2,1:4)];
B = [0 0 0 0; J(1,5:8); 0 0 0 0; J(2,5:8)];

% Use the formulation Delta \dot{state} = J Delta state

%% Simplify by pulling out some expressions
f1_fn = (l_BU^2 + l_SB^2 + 2*l_BU*l_SB*cos(gamma_B - theta_35));
f2_fn = (l_AR^2 + l_PA^2+ 2*l_AR*l_PA*cos(gamma_A1 + gamma_A2 + theta_13));
f3_fn = cos(gamma_A1 + gamma_A2 + theta_13);
f4_fn = sin(gamma_A1 + gamma_A2 + theta_13);
f5_fn = cos(theta_01 + theta_13 + theta_35);
f6_fn = sin(theta_01 + theta_13 + theta_35);

g1_fn = sin(acos((l_SB + l_BU*cos(gamma_B - theta_35))/f1_fn^(1/2)) ...
        - theta_01 - theta_13 - gamma_B);
g2_fn = sin(theta_01 + acos((l_PA + f3_fn*l_AR)/f2_fn^(1/2)));
g3_fn = (f4_fn*l_AR*l_PA*(l_PA + f3_fn*l_AR));
g4_fn = l_AR*l_PA*(l_PA + f3_fn*l_AR)^2;

orig_vars = [gamma_A1 gamma_A2 gamma_B ...
    l_PA l_AR l_QR l_AS l_AB l_SB l_TU l_BU l_BC ...
    l_AG3x l_AG3y l_BG5y l_BG5x l_BCx l_BCy l_ASx l_ASy ...
    l_PG2a l_SG4a l_AG3 l_BG5 ...
    I0_1 I0_2a IG_2b IG_3 IG_4a IG_4b IG_5 ...
    m_1 m_2a m_2b m_3 m_4a m_4b m_5 ...
    theta_13 theta_35 theta_13_DOT theta_35_DOT];

syms G1 G2 G3 ...
    l1 l2 l3 l4 l5 l6 l7 l8 l9 ...
    L1 L2 L3 L4 L5 L6 L7 L8 ...
    L9 K1 K2 K3 ...
    I1 I2 I3 I4 I5 I6 I7 ...
    m1 m2 m3 m4 m5 m6 m7 ...
    t1 t2 T1 T2

short_vars = [G1 G2 G3 ...
    l1 l2 l3 l4 l5 l6 l7 l8 l9 ...
    L1 L2 L3 L4 L5 L6 L7 L8 ...
    L9 K1 K2 K3 ...
    I1 I2 I3 I4 I5 I6 I7 ...
    m1 m2 m3 m4 m5 m6 m7 ...
    t1 t2 T1 T2];

syms f1 f2 f3 f4 f5 f6 g1 g2 g3 g4
A2 = subs(A, [f5_fn f6_fn g1_fn g2_fn g3_fn g4_fn], [f5 f6 g1 g2 g3 g4]);
B2 = subs(B, [f5_fn f6_fn g1_fn g2_fn g3_fn g4_fn], [f5 f6 g1 g2 g3 g4]);
A2 = subs(A2, [f1_fn f2_fn f3_fn f4_fn], [f1 f2 f3 f4]);
B2 = subs(B2, [f1_fn f2_fn f3_fn f4_fn], [f1 f2 f3 f4]);
A2 = subs(A2, orig_vars, short_vars);
B2 = subs(B2, orig_vars, short_vars);
```

D.4 Comparison Models

In addition to the model above, a simplified version was derived that ignored the impact of the motion of the cylinders on the inertial terms, and related θ and d terms via a projection, realized solely in the actuator force terms. This model acted essentially like a driven double pendulum with nonlinear torque mappings, and served to show that a

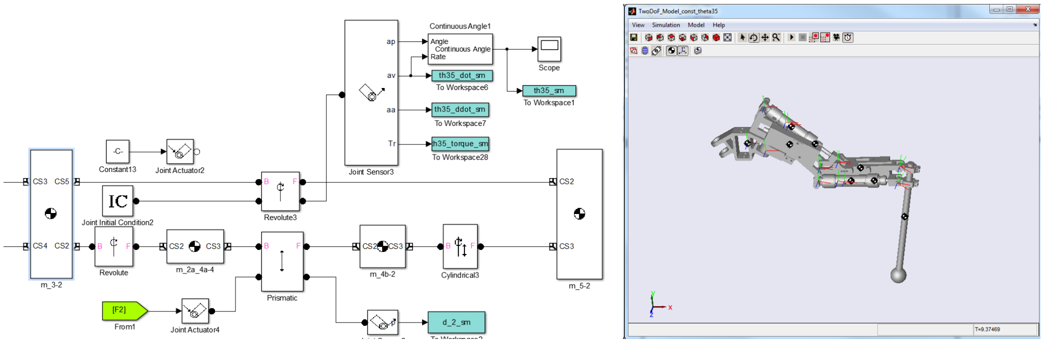

Figure 81: Left: SimMechanics code excerpt. Right: SimMechanics visualization.

simplified model was insufficient – as will be seen shortly, as well as to validate the MATLAB derivation approach.

Using SimMechanics, a physics simulation was constructed in Simulink. First, the SolidWorks model of the arm – the same one used earlier to obtain inertia properties – was exported to SimMechanics First Generation. The blocks that were generated were edited to include forcing and reorganized.

D.4.1 Simple Approximation

The preceding derivation is thorough but overly complex. A much easier solution is to represent the system as a double pendulum with representative links masses and inertias, and treat the forcing as moments. This can then follow a standard derivation. It was also beneficial to do these by hand to check the MATLAB calculations, since the other calculations were too large to do by hand.

The angles defined earlier will be used, but body 3 is expanded to include the cylinder/actuator inertia, such that $m_3 = 1.09$ kg, $I_{G_3} = 0.0038\ \mathrm{kg \cdot m^2}$, and $(I_3)_A = 0.0129\ \mathrm{kg \cdot m^2}$. From equation (201):

$$\begin{aligned} V_3 &= m_3 g \left(\vec{r}_{AG_3} \cdot \hat{j}_0\right) = m_3 g \left((l_{AG_3})_x \hat{i}_3 + (l_{AG_3})_y \hat{j}_3\right) \cdot \hat{j}_0 \\ &= m_3 g \left((l_{AG_3})_x \sin(\theta_{03}) + (l_{AG_3})_y \cos(\theta_{03})\right) \end{aligned} \tag{203}$$

$$\begin{aligned} V_5 &= m_5 g \left(\vec{r}_{AG_5} \cdot \hat{j}_0\right) = m_5 g \left(l_{AB} \hat{i}_3 + (l_{BG_5})_x \hat{i}_5 + (l_{BG_5})_y \hat{j}_5\right) \cdot \hat{j}_0 \\ &= m_5 g \left[l_{AB} \sin(\theta_{03}) \ + (l_{BG_5})_x \sin(\theta_{05}) + (l_{BG_5})_y \cos(\theta_{05})\right] \end{aligned} \tag{204}$$

Key velocity terms are similarly sourced from equations (184) and (188):

$$\vec{\omega}_3 = \dot{\theta}_{13}\hat{k} \tag{205}$$

$$\vec{\omega}_5 = (\dot{\theta}_{13} + \dot{\theta}_{35})\hat{k} \tag{206}$$

$$\begin{aligned}
\vec{v}_{G_5} &= \vec{v}_B + \vec{\omega}_5 \times \vec{r}_{BG_5} \\
&= l_{AB}\dot{\theta}_{13}\hat{j}_3 + (\dot{\theta}_{13} + \dot{\theta}_{35})\hat{k} \times \Big((l_{BG_5})_{x5}\hat{i}_5 + (l_{BG_5})_{y5}\hat{j}_5\Big) \\
&= l_{AB}\dot{\theta}_{13}(\sin\theta_{35}\hat{i}_5 + \cos\theta_{35}\hat{j}_5) - (l_{BG_5})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\hat{i}_5 + (l_{BG_5})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\hat{j}_5 \\
&= \Big(l_{AB}\dot{\theta}_{13}\sin\theta_{35} - (l_{BG_5})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\Big)\,\hat{i}_5 \\
&\quad + \Big(l_{AB}\dot{\theta}_{13}\cos\theta_{35} + (l_{BG_5})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\Big)\,\hat{j}_5
\end{aligned} \tag{207}$$

$$\begin{aligned}
\vec{v}_C &= -\Big(l_{AB}\dot{\theta}_{13}\sin(\theta_{03}) + (l_{BC})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\cos(\theta_{05}) + (l_{BC})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\sin(\theta_{05})\Big)\hat{i}_0 \\
&\quad + \Big(l_{AB}\dot{\theta}_{13}\cos(\theta_{03}) - (l_{BC})_{y5}(\dot{\theta}_{13} + \dot{\theta}_{35})\sin(\theta_{05}) + (l_{BC})_{x5}(\dot{\theta}_{13} + \dot{\theta}_{35})\cos(\theta_{05})\Big)\,\hat{j}_0
\end{aligned} \tag{208}$$

There are just two kinetic energy components:

$$\begin{aligned}
T &= \frac{1}{2}(I_3)_A\omega_3^2 + \frac{1}{2}I_{G_5}\omega_5^2 + \frac{1}{2}m_5 v_{G_5}^2 \\
&= \frac{1}{2}(I_3)_A\dot{\theta}_{13}^2 + \frac{1}{2}I_{G_5}\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right)^2 \\
&\quad + \frac{1}{2}m_5\Big(\dot{\theta}_{13}\, l_{AB}\ \sin(\theta_{35}) - (l_{BG_5})_y\ \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right)\Big)^2 \\
&\quad + \frac{1}{2}m_5\Big((l_{BG_5})_x\ \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) + \dot{\theta}_{13}\, l_{AB}\ \cos(\theta_{35})\Big)^2
\end{aligned} \tag{209}$$

Finally, virtual work is used to define generalized forces:

$$\begin{aligned}
\delta W &= \Gamma_1\delta\omega_3 + \Gamma_2\delta\omega_5 + F_3\delta r_{AC} \\
&= \Gamma_1\delta\theta_{13} + \Gamma_2(\delta\theta_{13} + \delta\theta_{35}) + F_3\delta r_{AC}
\end{aligned} \tag{210}$$

So in the absence of F_3, $Q_1 = \Gamma_1 + \Gamma_2$ and $Q_2 = \Gamma_2$. Then the Lagrangian is defined as $L = T - V$ and Lagrange's equations incur

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i \tag{211}$$

. Assembling the components, the Lagrangian is found to be

$$\begin{aligned} L &= T - V \\ &= \frac{1}{2}(I_3)_A\dot{\theta}_{13}^2 + \frac{1}{2}I_{G_5}\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right)^2 + \\ &+ \frac{1}{2}m_5\Big(\dot{\theta}_{13}\, l_{AB}\, \sin(\theta_{35}) - (l_{BG_5})_y\, \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right)\Big)^2 \\ &+ \frac{1}{2}m_5\Big(\dot{\theta}_{13}\, l_{AB}\, \cos(\theta_{35}) + (l_{BG_5})_x\, \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right)\Big)^2 \\ &- \Big(m_3 g\left((l_{AG_3})_x \sin(\theta_{03}) + (l_{AG_3})_y \cos(\theta_{03})\right) \\ &+ m_5 g\left[l_{AB}\sin(\theta_{03}) + (l_{BG_5})_x \sin(\theta_{05}) + (l_{BG_5})_y \cos(\theta_{05})\right]\Big) \end{aligned} \tag{212}$$

It has state velocity derivatives

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_{13}} = \Bigg(&(I_3)_A\dot{\theta}_{13} + I_{G_5}\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) \\ &+ m_5\Big(\dot{\theta}_{13}\, l_{AB}\, \sin(\theta_{35}) - (l_{BG_5})_y\, \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right)\Big)\left(l_{AB}\, \sin(\theta_{35}) - (l_{BG_5})_y\right) \\ &+ m_5\Big(\dot{\theta}_{13}\, l_{AB}\, \cos(\theta_{35}) + (l_{BG_5})_x\, \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right)\Big)\left(l_{AB}\, \cos(\theta_{35}) + (l_{BG_5})_x\right)\Bigg) \end{aligned} \tag{213}$$

$$\begin{aligned} &= (I_3)_A\dot{\theta}_{13} + I_{G_5}\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) \\ &\quad + m_5\dot{\theta}_{13}\, l_{AB}^2\, \sin^2(\theta_{35}) - m_5\dot{\theta}_{13}\, l_{AB}\, \sin(\theta_{35})\, (l_{BG_5})_y \\ &\quad - m_5(l_{BG_5})_y\, \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) l_{AB}\, \sin(\theta_{35}) + m_5(l_{BG_5})_y^2\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) \\ &\quad + m_5\dot{\theta}_{13}\, l_{AB}^2\, \cos^2(\theta_{35}) + m_5\dot{\theta}_{13}\, l_{AB}\, \cos(\theta_{35})\, (l_{BG_5})_x \\ &\quad + m_5(l_{BG_5})_x\, \left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) l_{AB}\, \cos(\theta_{35}) + m_5(l_{BG_5})_x^2\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) \end{aligned} \tag{214}$$

$$\begin{aligned} &= \left((I_3)_A + m_5\, l_{AB}^2\right)\dot{\theta}_{13} \\ &\quad + \left(I_{G_5} + m_5(l_{BG_5})_y^2 + m_5(l_{BG_5})_x^2\right)\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) \\ &\quad - \left(m_5(l_{BG_5})_y\, l_{AB}\, \sin(\theta_{35}) - m_5(l_{BG_5})_x\, l_{AB}\, \cos(\theta_{35})\right)\left(\dot{\theta}_{13} + \dot{\theta}_{35}\right) \\ &\quad + m_5\, l_{AB}\, \dot{\theta}_{13}\left((l_{BG_5})_x \cos(\theta_{35}) - (l_{BG_5})_y \sin(\theta_{35})\right) \end{aligned} \tag{215}$$

$$
\begin{aligned}
\frac{\partial L}{\partial \dot{\theta}_{35}} &= \Bigg(I_{G_5} \left(\dot{\theta}_{13} + \dot{\theta}_{35} \right) \\
&\qquad - m_5 (l_{BG_5})_y \Big(\dot{\theta}_{13}\, l_{AB}\, \sin(\theta_{35}) - (l_{BG_5})_y \, \left(\dot{\theta}_{13} + \dot{\theta}_{35} \right) \Big) \\
&\qquad + m_5 (l_{BG_5})_x \Big(\dot{\theta}_{13}\, l_{AB}\, \cos(\theta_{35}) + (l_{BG_5})_x \, \left(\dot{\theta}_{13} + \dot{\theta}_{35} \right) \Big) \Bigg) \\
&= \left(I_{G_5} + m_5 (l_{BG_5})_y^2 + m_5 (l_{BG_5})_x^2 \right) \left(\dot{\theta}_{13} + \dot{\theta}_{35} \right) \\
&\qquad + m_5\, l_{AB} \dot{\theta}_{13} \left((l_{BG_5})_x \, \cos(\theta_{35}) - (l_{BG_5})_y \, \sin(\theta_{35}) \right)
\end{aligned} \tag{216}
$$

Position state derivatives:

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_{13}} &= -\Big(m_3 g \left((l_{AG_3})_x \cos(\theta_{03}) - (l_{AG_3})_y \sin(\theta_{03}) \right) \\
&\qquad + m_5 g \left[l_{AB} \cos(\theta_{03}) + (l_{BG_5})_x \cos(\theta_{05}) - (l_{BG_5})_y \sin(\theta_{05}) \right] \Big) \\
&= -\left(m_5 g l_{AB} + m_3 g (l_{AG_3})_x \right) \cos(\theta_{03}) + m_3 g (l_{AG_3})_y \sin(\theta_{03}) \\
&\qquad - m_5 g (l_{BG_5})_x \cos(\theta_{05}) + m_5 g (l_{BG_5})_y \sin(\theta_{05})
\end{aligned} \tag{217}
$$

$$
\begin{aligned}
\frac{\partial L}{\partial \theta_{35}} &= m_5 \Big(\dot{\theta}_{13}\, l_{AB}\, \sin(\theta_{35}) - (l_{BG_5})_y \, \left(\dot{\theta}_{13} + \dot{\theta}_{35} \right) \Big) l_{AB} \dot{\theta}_{13} \cos(\theta_{35}) \\
&\qquad - m_5 \Big(\dot{\theta}_{13}\, l_{AB}\, \cos(\theta_{35}) + (l_{BG_5})_x \, \left(\dot{\theta}_{13} + \dot{\theta}_{35} \right) \Big) l_{AB} \dot{\theta}_{13} \sin(\theta_{35}) \\
&\qquad - m_5 g \left((l_{BG_5})_x \cos(\theta_{05}) - (l_{BG_5})_y \sin(\theta_{05}) \right) \\
&= -m_5 l_{AB} \dot{\theta}_{13} \left((l_{BG_5})_y \cos(\theta_{35}) + (l_{BG_5})_x \sin(\theta_{35}) \right) \left(\dot{\theta}_{13} + \dot{\theta}_{35} \right) \\
&\qquad + m_5 g (l_{BG_5})_y \sin(\theta_{05}) - m_5 g (l_{BG_5})_x \cos(\theta_{05})
\end{aligned} \tag{218}
$$

and time derivatives.

$$
\begin{aligned}
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_{13}} \right) &= \left((I_3)_A + m_5\, l_{AB}^2 \right) \ddot{\theta}_{13} \\
&\qquad + \left(I_{G_5} + m_5 (l_{BG_5})_y^2 + m_5 (l_{BG_5})_x^2 \right) \left(\ddot{\theta}_{13} + \ddot{\theta}_{35} \right) \\
&\qquad - \left(m_5 (l_{BG_5})_y \, l_{AB}\, \cos(\theta_{35}) + m_5 (l_{BG_5})_x \, l_{AB}\, \sin(\theta_{35}) \right) \left(\dot{\theta}_{13} \dot{\theta}_{35} + \dot{\theta}_{35}^2 \right) \\
&\qquad - \left(m_5 (l_{BG_5})_y \, l_{AB}\, \sin(\theta_{35}) - m_5 (l_{BG_5})_x \, l_{AB}\, \cos(\theta_{35}) \right) \left(\ddot{\theta}_{13} + \ddot{\theta}_{35} \right) \\
&\qquad + m_5\, l_{AB}\, \ddot{\theta}_{13} \left((l_{BG_5})_x \cos(\theta_{35}) - (l_{BG_5})_y \sin(\theta_{35}) \right) \\
&\qquad + m_5\, l_{AB}\, \dot{\theta}_{13} \dot{\theta}_{35} \left(-(l_{BG_5})_x \sin(\theta_{35}) - (l_{BG_5})_y \cos(\theta_{35}) \right)
\end{aligned} \tag{219}
$$

$$\begin{aligned}\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_{35}}\right) &= \big(I_{G_5} + m_5(l_{BG_5})_y^2 + m_5(l_{BG_5})_x^2\,\big)\left(\ddot{\theta}_{13} + \ddot{\theta}_{35}\right) \\ &\quad + m_5\, l_{AB}\ddot{\theta}_{13}\left((l_{BG_5})_x \cos(\theta_{35}) - (l_{BG_5})_y \sin(\theta_{35})\right) \\ &\quad - m_5\, l_{AB}\dot{\theta}_{13}\dot{\theta}_{35}\left((l_{BG_5})_x \sin(\theta_{35}) + (l_{BG_5})_y \cos(\theta_{35})\right)\end{aligned} \tag{220}$$

Finally, the equations can be assembled and solved for $\ddot{\theta}_{13}$ and $\ddot{\theta}_{35}$:

$$\begin{aligned}&(I_3)_A + I_{G_5} + m_5\Bigg(\,\left((l_{BG_5})_x + l_{AB}\,\cos(\theta_{35})\right)^2 + \Big(\big(l_{BG_5}\big)_y - l_{AB}\,\sin(\theta_{35})\Big)^2\Bigg)\ddot{\theta}_{35} \\ &+I_{G_5} + m_5\,\Big((l_{BG_5})_x\,\left((l_{BG_5})_x + l_{AB}\,\cos(\theta_{35})\right) + (l_{BG_5})_y\,\Big((l_{BG_5})_y - l_{AB}\,\sin(\theta_{35})\Big)\Big)\,\ddot{\theta}_{13} \\ &+m_3 g\,\Big((l_{AG_3})_x\,\cos(\theta_{01}+\theta_{13}) - (l_{AG_3})_y\,\sin(\theta_{01}+\theta_{13})\Big) \\ &-\Big(m_5\,\Big(l_{AB}\,\dot{\theta}_{35}\,\sin(\theta_{35})\,\Big((l_{BG_5})_x\,\Big(\dot{\theta}_{13}+\dot{\theta}_{35}\Big) + l_{AB}\,\dot{\theta}_{13}\,\cos(\theta_{35})\Big) \\ &+l_{AB}\,\dot{\theta}_{35}\,\cos(\theta_{35})\,\Big((l_{BG_5})_y\,\Big(\dot{\theta}_{13}+\dot{\theta}_{35}\Big) - l_{AB}\,\dot{\theta}_{13}\,\sin(\theta_{35})\Big) \\ &+l_{AB}\,\dot{\theta}_{13}\,\dot{\theta}_{35}\,\sin(\theta_{35})\,\left((l_{BG_5})_x + l_{AB}\,\cos(\theta_{35})\right) \\ &+l_{AB}\,\dot{\theta}_{13}\,\dot{\theta}_{35}\,\cos(\theta_{35})\,\Big((l_{BG_5})_y - l_{AB}\,\sin(\theta_{35})\Big)\,\Big)\Big) \\ &+m_5 g\,\Big(l_{AB}\,\cos(\theta_{01}+\theta_{13}) + (l_{BG_5})_x\,\cos(\theta_{01}+\theta_{13}+\theta_{35}) - (l_{BG_5})_y\,\sin(\theta_{01}+\theta_{13}+\theta_{35})\Big) \\ &= \Gamma_1 + \Gamma_2\end{aligned} \tag{221}$$

$$\begin{aligned}&I_{G_5} + m_5\,\Big((l_{BG_5})_x\,\left((l_{BG_5})_x + l_{AB}\,\cos(\theta_{35})\right) + (l_{BG_5})_y\,\Big((l_{BG_5})_y - l_{AB}\,\sin(\theta_{35})\Big)\Big)\,\ddot{\theta}_{13} \\ &+I_{G_5} + m_5\,\Big({(l_{BG_5})_x}^2 + {(l_{BG_5})_y}^2\Big)\,\ddot{\theta}_{35} \\ &+\Big(m_5\,\Big(l_{AB}\,\dot{\theta}_{13}\,\sin(\theta_{35})\,\Big((l_{BG_5})_x\,\Big(\dot{\theta}_{13}+\dot{\theta}_{35}\Big) + l_{AB}\,\dot{\theta}_{13}\,\cos(\theta_{35})\Big) \\ &+l_{AB}\,\dot{\theta}_{13}\,\cos(\theta_{35})\,\Big((l_{BG_5})_y\,\Big(\dot{\theta}_{13}+\dot{\theta}_{35}\Big) - l_{AB}\,\dot{\theta}_{13}\,\sin(\theta_{35})\Big)\,\Big)\Big) \\ &-\Big(m_5\,\Big(l_{AB}\,\left(l_{BG_5}\right)_x\,\dot{\theta}_{13}\,\dot{\theta}_{35}\,\sin(\theta_{35}) + l_{AB}\,\left(l_{BG_5}\right)_y\,\dot{\theta}_{13}\,\dot{\theta}_{35}\,\cos(\theta_{35})\Big)\Big) \\ &+m_5 g\,\Big((l_{BG_5})_x\,\cos(\theta_{01}+\theta_{13}+\theta_{35}) - (l_{BG_5})_y\,\sin(\theta_{01}+\theta_{13}+\theta_{35})\Big) \\ &= \Gamma_2\end{aligned} \tag{222}$$

For practical use, we need to solve for the accelerations. This can be done by grouping the above equations into terms: $C_1\ddot{\theta}_{13} + C_2\ddot{\theta}_{35} + C_3 = Q_1$, and $D_1\ddot{\theta}_{13} + D_2\ddot{\theta}_{35} + D_3 = Q_2$.

Written in matrix form:

$$\begin{bmatrix} C_1 & C_2 \\ D_1 & D_2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_{13} \\ \ddot{\theta}_{35} \end{bmatrix} = \begin{bmatrix} Q_1 - C_3 \\ Q_2 - C_3 \end{bmatrix} \tag{223}$$

These can then be solved:

$$\begin{bmatrix} \ddot{\theta}_{13} \\ \ddot{\theta}_{35} \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ D_1 & D_2 \end{bmatrix}^{-1} \begin{bmatrix} Q_1 - C_3 \\ Q_2 - C_3 \end{bmatrix}$$

$$\begin{bmatrix} \ddot{\theta}_{13} \\ \ddot{\theta}_{35} \end{bmatrix} = \begin{bmatrix} \frac{D_2}{C_1\,D_2 - C_2\,D_1} & -\frac{C_2}{C_1\,D_2 - C_2\,D_1} \\ -\frac{D_1}{C_1\,D_2 - C_2\,D_1} & \frac{C_1}{C_1\,D_2 - C_2\,D_1} \end{bmatrix} \begin{bmatrix} Q_1 - C_3 \\ Q_2 - C_3 \end{bmatrix} \tag{224}$$

D.4.2 Inertial model in Simulink

Using *SimMechanics*, a physics simulation was constructed in Simulink. First, the SolidWorks model of the arm – the same one used earlier to obtain inertia properties – was exported to SimMechanics First Generation. The blocks that were generated were edited to include forcing and reorganized.

D.4.3 Model Performance

Time-series comparison and spectral analysis were used to compare the performance of the two analytically derived sets of equations of motion and the SimMechanics model.

For the most direct comparison, PID controllers were applied to each model to force a setpoint. Feedforward chirp excitations were added to the control input to act as a force disturbance. The resultant plots could be used to compare performance of models within the desired operating range.

First, each joint was individually excited while the other was held constant using the controller. Figures 82 and 83 show the response of θ_{13} and θ_{35} to excitation of θ_{13}, while Figures 84 and 85 do the same for an excitation to θ_{35}.

Next, both angles are forced at the same time. Figures 86 and 87 show the individual responses of each joint to their own excitation, while Figure 88 demonstrates the response of each angle to the excitation of the opposite joint.

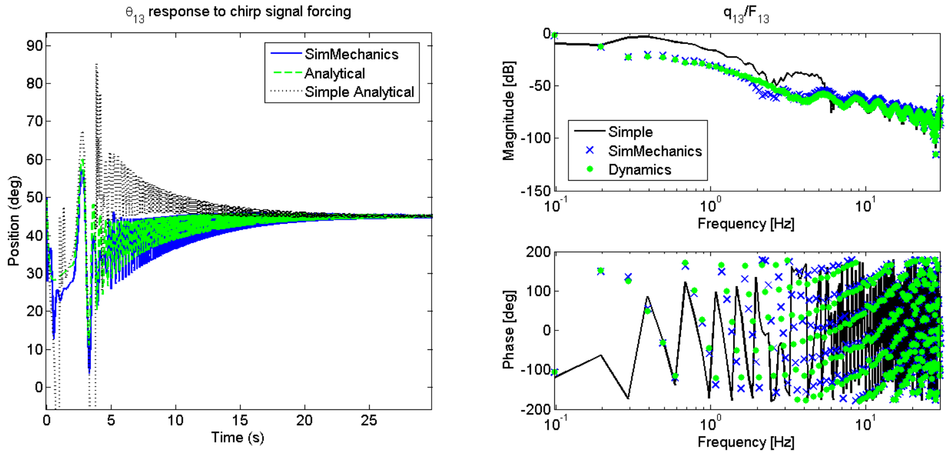

Figure 82: Position and frequency response of θ_{13} when θ_{13} is given a 20 N force chirp excitation.

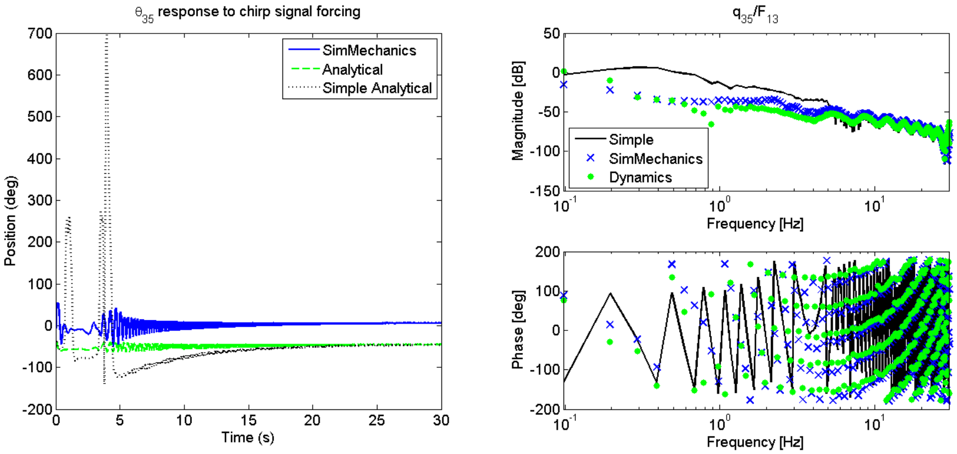

Figure 83: Position and frequency response of θ_{35} when θ_{13} is given a 20 N force chirp excitation.

These figures appear to show overall good correspondence between analytical and Simscape models. Of course, correspondence in the appropriate domains isn't perfect. Figures 89 and 90 demonstrate model correspondence in time for each joint when the other is held fixed (with a weld joint in Simscape, and by setting acceleration to zero in the analytical models). These figures use gravity in lieu of a controller to force the position to remain inside the operating range. While joint θ_{35} matched almost perfectly, joint θ_{13} differs at the extremities. It is also possible to observe a slight phase delay that begins to grow. Since all the geometric quantities were based on trigonometric laws that are valid in this operating range, and as all the inertial and geometric parameters were checked multiple

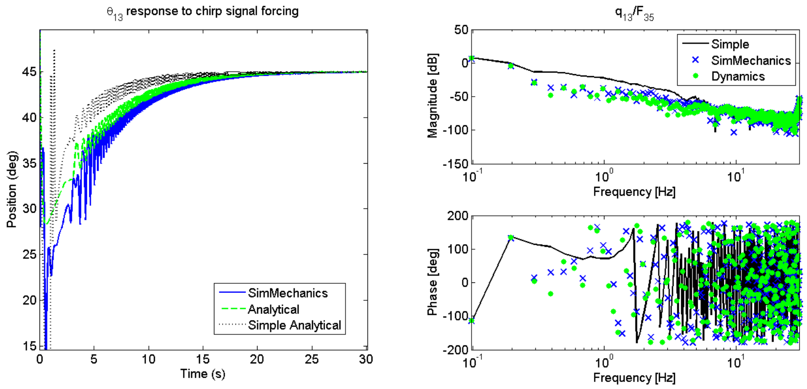

Figure 84: Position and frequency response of θ_{13} when θ_{35} is given a 60 N force chirp excitation.

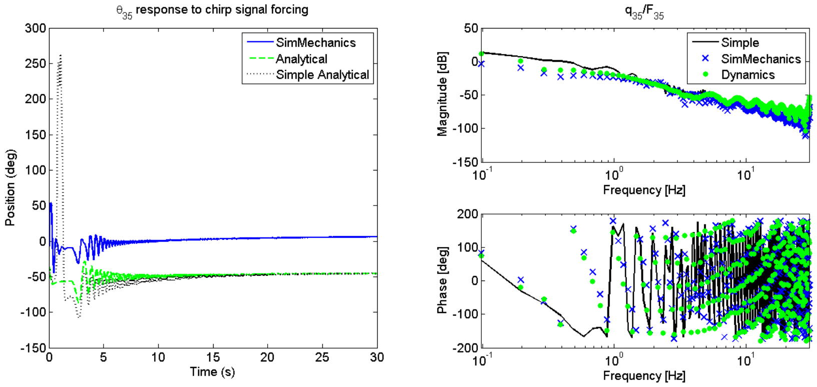

Figure 85: Position and frequency response of θ_{35} when θ_{13} is given a 60 N force chirp excitation.

times, it is assumed that this is the result of some unknown difference between Lagrangian and SimMechanics dynamics derivations.

From these tests, it appears that the geometric constraints incurred by the cylinder/piston action do have a significant effect on model dynamics: while behavior can be captured by simply altering the inertias of a forced double pendulum, magnitudes match better when the detail geometry is included in the derivation of the dynamics. However, for use in control, especially predictive control over a short time horizon, or control with built-in robustness to model error, the simplified model appears sufficient.

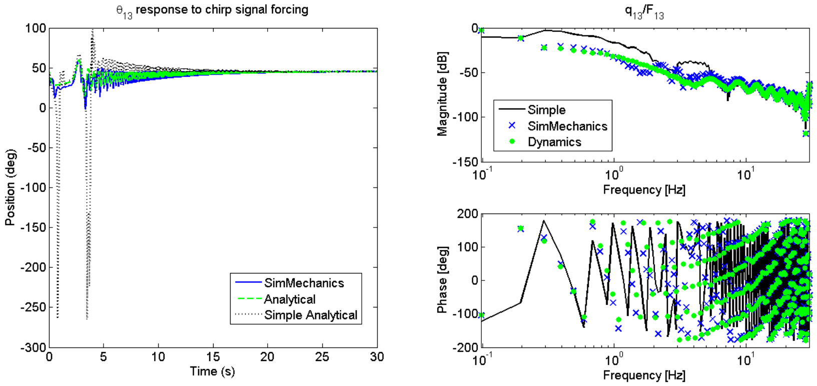

Figure 86: Position and frequency response of θ_{13} to a 60 N force chirp excitation when both angles are excited.

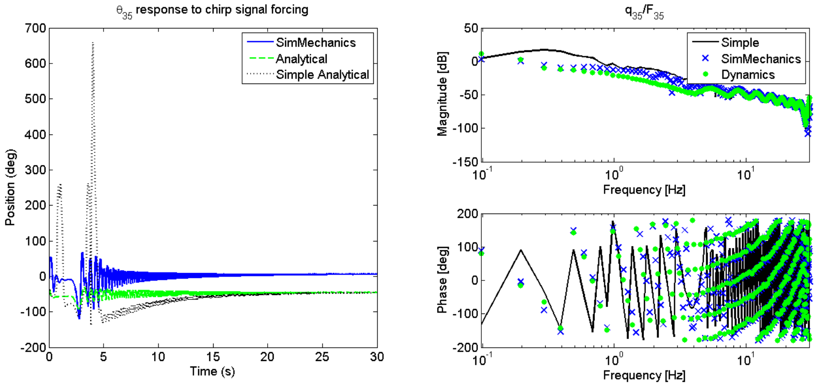

Figure 87: Position and frequency response of θ_{35} to a 20 N force chirp excitation when both angles are excited.

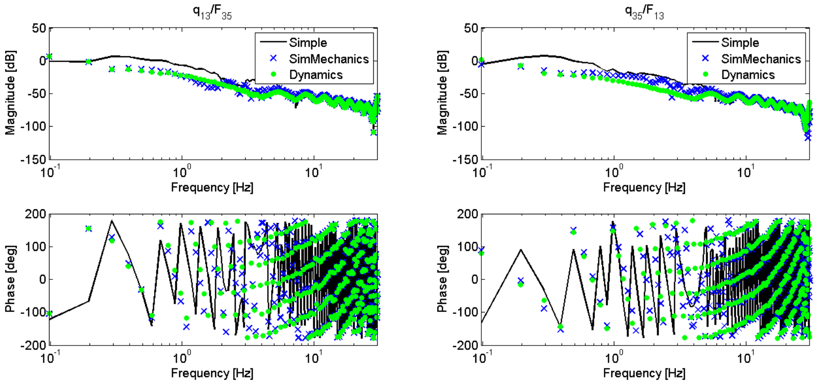

Figure 88: Frequency response of θ_{13} and θ_{35} to excitation of the other joint when both are excited at the same time.

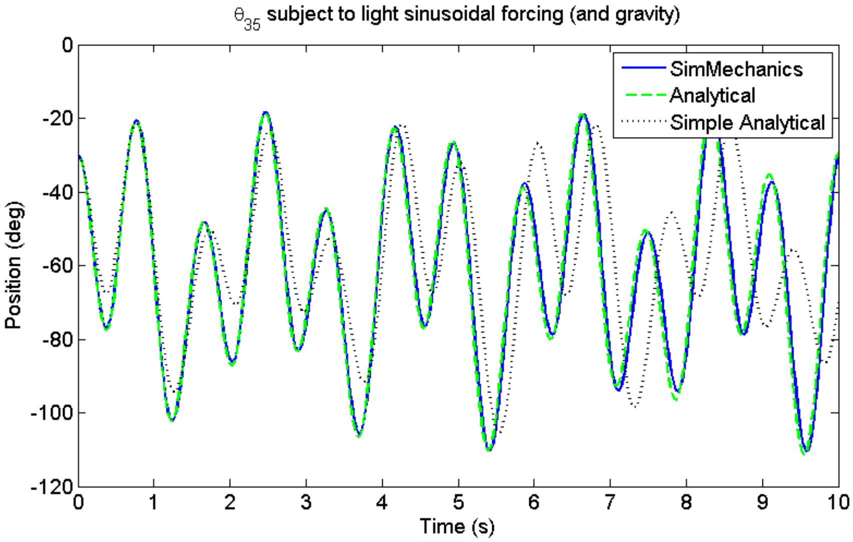

Figure 89: Frequency response of θ_{35} to light sinusoidal forcing (enough to move the arm, but not enough to exceed a realistic operating range).

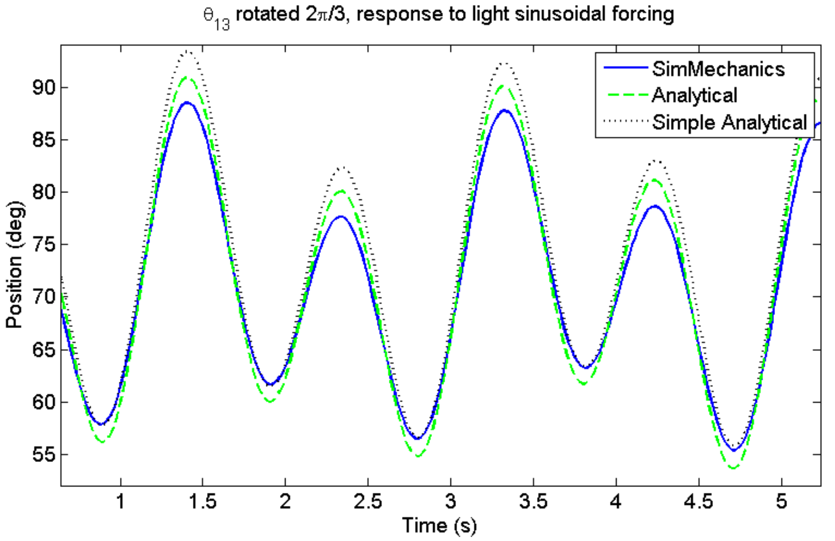

Figure 90: Frequency response of θ_{13} to light sinusoidal forcing (enough to move the arm, but not enough to exceed a realistic operating range). The model has been rotated so that gravity can be used to keep it inside a realistic operating range.

REFERENCES

[1] "Iso 10218: Robots for industrial environments – safety requirements – part 1: Robot," 2006.

[2] Al-Dakkan, K. A., Barth, E. J., and Goldfarb, M., "Dynamic constraint-based energy-saving control of pneumatic servo systems," *Journal of Dynamic Systems, Measurement and Control*, vol. 128, no. 3, pp. 655–62, 2006.

[3] Alessio, A. and Bemporad, A., "A survey on explicit model predictive control," *Nonlinear Model Predictive Control: Towards New Challenging Applications*, vol. 384, pp. 345–369, 2009.

[4] Arevalo, J. C. and Garcia, E., "Impedance control for legged robots: An insight into the concepts involved," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, pp. 1400–1411, November 2012.

[5] Armstrong-Helouvry, B., Dupont, P., and Canudasdewit, C., "Survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–1138, 1994.

[6] Aschemann, H. and Schindele, D., "Nonlinear model predictive control of a high-speed linear axis driven by pneumatic muscles," in *American Control Conference*, pp. 3017–3022, IEEE, 2008.

[7] Bemporad, A., "Reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 415–419, 1998.

[8] Bendov, D. and Salcudean, S. E., "A force-controlled pneumatic actuator for use in teleoperation masters," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 938–943, 1993.

[9] Bicchi, A., Peshkin, M. A., and Colgate, J. E., "Safety for physical human–robot interaction," in *Springer handbook of robotics*, pp. 1335–1348, Springer, 2008.

[10] Bobrow, J. E. and Jabbari, F., "Adaptive pneumatic force actuation and position control," *Transactions of the ASME. Journal of Dynamic Systems, Measurement and Control*, vol. 113, no. 2, pp. 267–72, 1991.

[11] Bone, G. M. and Shu, N., "Experimental comparison of position tracking control algorithms for pneumatic cylinder actuators," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 5, pp. 557–61, 2007.

[12] Borrelli, F., *Receding Horizon Control*, vol. 290 of *Lecture Notes in Control and Information Sciences*, ch. 4, pp. 75–101. Springer Berlin Heidelberg, 2003.

[13] Buerger, S. P., *Stable, High-Force, Low-Impedance Robotic Actuators for Human-Interactive Machines.* Dissertation, Massachusetts Institute of Technology, June 2005.

[14] Camacho, E. F. and Bordons, C., "Nonlinear model predictive control: An introductory review," *Assessment and Future Directions of Nonlinear Model Predictive Control*, vol. 358, pp. 1–16, 2007.

[15] Cannon, M., "C21 model predictive control – lecture notes," Oct. 2013.

[16] Carneiro, J. F. and de Almeida, F. G., "Modeling pneumatic servovalves using neural networks," in *2006 IEEE International Symposium on Intelligent Control*, pp. 790–795, 2006.

[17] Chen, W.-H., "Stability analysis of classic finite horizon model predictive control," *International Journal of Control, Automation and Systems*, vol. 8, no. 2, pp. 187–197, 2010.

[18] Chickh, L., Poignet, P., Pierrot, F., and Michelin, M., "A generalized predictive force controller for electropneumatic cylinders," in *NOLCOS'10: 8th IFAC Symposium on Nonlinear Control Systems*, 2010.

[19] Chipalkatty, R., Daepp, H. G., Egerstedt, M., and Book, W., "Human-in-the-loop: Mpc for shared control of a quadruped rescue robot," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4556–4561, September 25 - 30, 2011 2011.

[20] Conceicao, A. G. S., Dorea, C. E. T., and Sb, J. C. L. B., "Predictive control of an omnidirectional mobile robot with friction compensation," in *2010 Latin American Robotics Symposium and Intelligent Robotic Meeting (LARS)*, pp. 30–35, 2010.

[21] Daepp, H. G. and Book, W., "Model predictive control for compliant pneumatic systems," in *ASME 2014 Dynamic Systems and Control Conference (DSCC)*, 2014.

[22] Daepp, H. G. and Book, W., "Predictive compensation for control of pneumatic actuators," in *Fluid Power Net International (FPNI) PhD Symposium*, ASME, 2014.

[23] Daepp, H. G. and Book, W., "Impedance behavior of controllers for compliant positioning of a pneumatically actuated system," in *ASME Dynamic Systems and Control Conference (DSCC)*, 2015.

[24] Daepp, H. G., Book, W., Kim, T., and Radecki, P., "An interactive simulation for a fluid-powered legged rescue robot," in *2010 International Symposium on Flexible Automation*, 2010.

[25] Daepp, H. G., "Development of a multi-platform simulation for a pneumatically-actuated quadruped robot," thesis, Georgia Institute of Technology, 2011.

[26] Daepp, H. G. Book, W., "Value of a high fidelity actuator model for dynamic simulation of a pneumatic rescue robot," in *IFAC World Congress*, 24-29 August 2014.

[27] De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A., "An atlas of physical human-robot interaction," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008.

[28] Di Cairano, S. and Bemporad, A., "Model predictive control tuning by controller matching," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 185–190, 2010.

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

[30] Driessen, B. J. and Sadegh, N., "On the discontinuity of the costates for optimal control problems with coulomb friction," *Optimal Control Applications & Methods*, vol. 22, no. 4, pp. 197–200, 2001.

[31] Durbha, V. and Li, P. Y., "Passive bilateral tele-operation and human power amplification with pneumatic actuators," in *Proceedings of the ASME 2009 Dynamic Systems and Control Conference*, 2009.

[32] Falugi, P. and Mayne, D. Q., "Model predictive control for tracking random references," in *2013 European Control Conference (ECC)*, 2013 European Control Conference (ECC), pp. 518–23, IEEE, 17-19 July 2013 2013.

[33] Ferretti, G., Magnani, G. A., and Rocco, P., "Impedance control for elastic joints industrial manipulators," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 488–498, 2004.

[34] FESTO, *Datasheet: Proportional directional control valves MPYE*, 2015.

[35] Fischer, G. S., Iordachita, I., Csoma, C., Tokuda, J., DiMaio, S. P., Tempany, C. M., Hata, N., and Fichtinger, G., "Mri-compatible pneumatic robot for transperineal prostate needle placement," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 3, pp. 295–305, 2008.

[36] Garabini, M., Passaglia, A., Belo, F., Salaris, P., and Bicchi, A., "Optimality principles in variable stiffness control: The vsa hammer," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3770–3775, 2011.

[37] Gilbert, E. G. and Kolmanovsky, I., "Fast reference governors for systems with state and control constraints and disturbance inputs," *International Journal of Robust and Nonlinear Control*, vol. 9, no. 15, pp. 1117–1141, 1999.

[38] Goldfarb, M., Barth, E. J., Gogola, M. A., and Wehrmeyer, J. A., "Design and energetic characterization of a liquid-propellant-powered actuator for self-powered robots," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 2, pp. 254–262, 2003.

[39] Haddadin, S., Albu-Schaeffer, Affer, A., and Hirzinger, G., *Safe physical human-robot interaction: measurements, analysis and new insights*, pp. 395–407. Springer, 2011.

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

[41] Haddadin, S., Haddadin, S., Khoury, A., Rokahr, T., Parusel, S., Burgkart, R., Bicchi, A., and Albu-Schaeffer, A., "On making robots understand safety: Embedding injury knowledge into control," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1578–1602, 2012.

[42] Han Koo, L., Gi Sang, C., and Gi Heung, C., "A study on tracking position control of pneumatic actuators," *Mechatronics*, vol. 12, no. 6, pp. 813–31, 2002.

[43] Hogan, N. and Buerger, S. P., "Impedance and interaction control," in *Robotics and automation handbook* (Kurfess, T., ed.), ch. 19, CRC Press, 2005.

[44] Hoshino, K. and Kawabuchi, I., "Control of generated force and stiffness in pneumatic air cylinder actuator," in *1st IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 1119–1124, IEEE, 20-22 Feb. 2006 2006.

[45] Houska, B., Ferreau, H., and Diehl, M., "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[46] Hyun, D., Yang, H. S., Park, J., and Shim, Y., "Variable stiffness mechanism for human-friendly robots," *Mechanism and Machine Theory*, vol. 45, no. 6, pp. 880–897, 2010.

[47] Jamaludin, Z., Van Brussel, H., and Swevers, J., "Friction compensation of an xy feed table using friction-model-based feedforward and an inverse-model-based disturbance observer," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3848–3853, 2009.

[48] Jung-Jun, P. and Jae-Bok, S., "Collision analysis and evaluation of collision safety for service robots working in human environments," in *International Conference on Advanced Robotics (ICAR)*, 2009.

[49] Ke, J., Wang, J., Jia, N., Yang, L., and Wu, Q. H., "Energy efficiency analysis and optimal control of servo pneumatic cylinders," in *Proceedings of 2005 IEEE Conference on Control Applications*, pp. 541–546, 2005.

[50] Kogiso, K. and Hirata, K., "Reference governor for constrained systems with time-varying references," *Robotics and Autonomous Systems*, vol. 57, no. 3, pp. 289–295, 2009.

[51] Kriegsmann, M., "Servocontrol with pneumatic actuators," *Machine Design*, vol. 79, no. Copyright 2008, The Institution of Engineering and Technology, pp. 78–80, 2007.

[52] Lee, H. K., Choi, G. S., and Choi, G. H., "A study on tracking position control of pneumatic actuators," *Mechatronics*, vol. 12, no. 6, pp. 813–31, 2002.

[53] Lee, J. H., "Model predictive control: Review of the three decades of development," *International Journal of Control Automation and Systems*, vol. 9, no. 3, pp. 415–424, 2011.

[54] Luk, B. L., Collie, A. A., and Billingsley, J., "Robug ii: An intelligent wall climbing robot," in *Proceedings od the 1991 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2342–2347, 1991.

[55] Luk, B. L., Collie, A. A., Piefort, V., and Virk, G. S., "Robug iii: a teleoperated climbing and walking robot," in *UKACC International Conference on Control*, vol. 1, pp. 347–352, 1996.

[56] Maciejowski, J. M., *Predictive control: with constraints.* Pearson education, 2002.

[57] Matousek, P., "Predictive control of servo-pneumatic system," *Annals of Faculty Engineering Hunedoara – International Journal of Engineering*, vol. IX, pp. 21–26, 2011.

[58] Mattingley, J. and Boyd, S., "Cvxgen: a code generator for embedded convex optimization," *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

[59] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[60] Mehdi, H. and Boubaker, O., "Stiffness and impedance control using lyapunov theory for robot-aided rehabilitation," *International Journal of Social Robotics*, vol. 4, no. 1, pp. 107–119, 2012.

[61] Moritz, E. and Haake, S., *The Engineering of Sport 6: Volume 3: Developments for Innovation.* The Engineering of Sport 6, Springer New York, 2010.

[62] Muske, K. R. and Badgwell, T. A., "Disturbance modeling for offset-free linear model predictive control," *Journal of Process Control*, vol. 12, no. 5, pp. 617–632, 2002.

[63] Najafi, F. and Hejrati, B., "Impedance control of a pneumatic actuator using fast switching on/off solenoid valves for tasks containing contact," in *17th. Annual (International) Conference on Mechanical Engineering (ISME)*, 2009.

[64] Nelson, G. M. and Quinn, R. D., "Posture control of a cockroach-like robot," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, vol. 1, pp. 157–162, 1998.

[65] Pratt, G. A., Williamson, M. M., Dillworth, P., Pratt, J., and Wright, A., "Stiffness isn't everything," in *experimental robotics IV*, pp. 253–262, Springer, 1997.

[66] Pratt, J. E. and Krupp, B. T., "Series elastic actuators for legged robots," *Unmanned Ground Vehicle Technology Vi*, vol. 5422, pp. 135–144, 2004.

[67] Richardson, R., Brown, M., Bhakta, B., and Levesley, M., "Impedance control for a pneumatic robot-based around pole-placement, joint space controllers," *Control Engineering Practice*, vol. 13, no. 3, pp. 291–303, 2005.

[68] Riofrio, J. A., Al-Dakkan, K., Hofacker, M. E., and Barth, E. J., "Control-based design of free-piston stirling engines," in *American Control Conference*, pp. 1533–1538, 2008.

[69] Riofrio, J. A. and Barth, E. J., "Design of a free piston pneumatic compressor as a mobile robot power supply," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 235–240, 2005.

[70] Robinson, D. W., *Design and analysis of series elasticity in closed-loop actuator force control.* Dissertation, Massachusetts Institute of Technology, 2000.

[71] Rossiter, J. A., *Model-based predictive control: a practical approach.* CRC press, 2013.

[72] Runzi, C. and Kay-Soon, L., "A repetitive model predictive control approach for precision tracking of a linear motion system," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1955–1962, 2009.

[73] Schulte, H. and Hahn, H., "Fuzzy state feedback gain scheduling control of servo-pneumatic actuators," *Control Engineering Practice*, vol. 12, no. 5, pp. 639–50, 2004.

[74] Scokaert, P. O. M., Mayne, D. Q., and Rawlings, J. B., "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999.

[75] Shen, X. R. and Goldfarb, M., "On the enhanced passivity of pneumatically actuated impedance-type haptic interfaces," *IEEE Transactions on Robotics*, vol. 22, no. 3, pp. 470–480, 2006.

[76] Shen, X. R. and Goldfarb, M., "Simultaneous force and stiffness control of a pneumatic actuator," *Journal of Dynamic Systems Measurement and Control-Transactions of the ASME*, vol. 129, no. 4, pp. 425–434, 2007.

[77] Shu, N. and Bone, G. M., "Development of a nonlinear dynamic model for a servo pneumatic positioning system," in *2005 IEEE International Conference on Mechatronics and Automation*, vol. 1, pp. 43–48, 2005.

[78] Slotine, J.-J. E., Li, W., and others, *Applied nonlinear control.* Prentice-Hall Englewood Cliffs, NJ, 1991.

[79] Stewart, D. E. and Anitescu, M., "Optimal control of systems with discontinuous differential equations," *Numerische Mathematik*, vol. 114, no. 4, pp. 653–695, 2010.

[80] Tagliamonte, N. L. and Accoto, D., "Passivity constraints for the impedance control of series elastic actuators," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 228, no. 3, pp. 138–153, 2014.

[81] Tan, H. Z., Srinivasan, M. A., Eberman, B., and Cheng, B., "Human factors for the design of force-reflecting haptic interfaces," *Dynamic Systems and Control*, vol. 55, no. 1, pp. 353–359, 1994.

[82] Taware, A., Pradhan, N., and Tao, G., "Feedback linearization based adaptive friction compensation for a sandwich nonlinear system," in *Proceedings of the 2001 American Control Conference*, vol. 1, pp. 588–593, 2001.

[83] Taware, A., Tao, G., and Teolis, C., "Design and analysis of a hybrid control scheme for sandwich nonsmooth nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 145–150, 2002.

[84] Thomas, M. B. and Maul, G. P., "Considerations on a mass-based system representation of a pneumatic cylinder," *Journal of Fluids Engineering*, vol. 131, no. 7, pp. 0411011–04110110, 2009.

[85] Todorov, E., Hu, C. Y., Simpkins, A., and Movellan, J., "Identification and control of a pneumatic robot," in *3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 373–380, 2010.

[86] Todorov, E., "Optimality principles in sensorimotor control," *Nature neuroscience*, vol. 7, no. 9, pp. 907–915, 2004.

[87] Tsagarakis, N. G., Laffranchi, M., Vanderborght, B., and Caldwell, D. G., "A compact soft actuator unit for small scale human friendly robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4356–4362, 2009.

[88] Tsuji, T. and Tanaka, Y., "Tracking control properties of human-robotic systems based on impedance control," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 4, pp. 523–535, 2005.

[89] Tung, E. D., Anwar, G., and Tomizuka, M., "Low-velocity friction compensation and feedforward solution based on repetitive control," *Journal of Dynamic Systems Measurement and Control*, vol. 115, no. 2A, pp. 279–284, 1993.

[90] Tzafestas, C. S., M'Sirdi, N. K., and Manamani, N., "Adaptive impedance control applied to a pneumatic legged robot," *Journal of Intelligent & Robotic Systems*, vol. 20, no. 2-4, pp. 105–129, 1997.

[91] Van Ham, R., Sugar, T. G., Vanderborght, B., Hollander, K. W., and Lefeber, D., "Compliant actuator designs review of actuators with passive adjustable compliance/controllable stiffness for robotic applications," *IEEE Robotics & Automation Magazine*, vol. 16, no. 3, pp. 81–94, 2009.

[92] Villani, L. and De Schutter, J., "Force control," in *Robotics and automation handbook*, pp. 161–185, Springer, 2008.

[93] Wait, K. W. and Goldfarb, M., "Enhanced performance and stability in pneumatic servosystems with supplemental mechanical damping," *Journal of Dynamic Systems, Measurement and Control*, vol. 132, no. 4, p. 041012, 2010.

[94] Wang, J. H., Kotta, U., and Ke, J., "Tracking control of nonlinear pneumatic actuator systems using static state feedback linearization of the input-output map," *Proceedings of the Estonian Academy of Sciences-Physics Mathematics*, vol. 56, no. 1, pp. 47–66, 2007.

[95] Wang, L., "Implementation of predictive control systems," in *Model predictive control system design and implementation using MATLAB®*, Advances in Industrial Control, ch. 10, pp. 333–366, Springer London, 2009.

[96] WANG, X., STOORVOGEL, A. A., SABERI, A., GRIP, H. F., ROY, S., and SANNUTI, P., "Stabilization of a class of sandwich systems via state feedback," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2156–2160, 2010.

[97] WYETH, G., "Demonstrating the safety and performance of a velocity sourced series elastic actuator," in *2008 IEEE International Conference on Robotics and Automation*, pp. 3642–3647, 2008.

[98] YANG, W. and BOYD, S., "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[99] ZHU, G. G., MA, J., and SCHOCK, H., "An iterative algorithm for model-based predictive control of an electro-pneumatic valve actuator," in *2009 American Control Conference*, pp. 2684–2689, 2009.

[100] ZHU, Y. and BARTH, E. J., "Impedance control of a pneumatic actuator for contact tasks," in *2005 IEEE International Conference on Robotics and Automation*, vol. 2005, pp. 987–992, April 18 - 22 2005.

[101] ZINN, M., KHATIB, O., ROTH, B., and SALISBURY, J. K., "Playing it safe," *IEEE Robotics & Automation Magazine*, vol. 11, no. 2, pp. 12–21, 2004.