

**MODEL-BASED CONTROL OF REMOTE ACTUATION
BASED ON CABLE-CONDUIT TRANSMISSIONS**

by

Andrew Borowski

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Mechanical Engineering

Spring 2018

© 2018 Andrew Borowski
All Rights Reserved

**MODEL-BASED CONTROL OF REMOTE ACTUATION
BASED ON CABLE-CONDUIT TRANSMISSIONS**

by

Andrew Borowski

Approved: _____
Ajay K. Prasad, Ph.D.
Chair of the Department of Mechanical Engineering

Approved: _____
Babatunde A. Ogunnaike, Ph.D.
Dean of the College of Engineering

Approved: _____
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Fabrizio Sergi, PhD., Asst. Professor of Biomedical Engineering at the University of Delaware not only for helping me work through my research, but also for making sure to put a focus on development as a professional.

I would also like to thank my co-advisor Dustyn Roberts, P.E., PhD., now Asst. Professor at the Temple University College of Engineering for giving me opportunities to do work in education and helping out through the rough times I was in after coming to UD.

Thirdly, thanks to Alex Metz, a student who worked with me on parts of this project as part of his undergraduate research, for always being there to bounce ideas off of.

Finally, a thanks to Jill Higginson, PhD. and Ioannis Poulakakis, PhD., professors of Mechanical Engineering at the University of Delaware for participating on my thesis committee.

This thesis is dedicated to my wife Melissa, who has kept me sane for the last n years, and (hopefully) for the next k as well. Oh, who am I kidding? That ship sailed long ago. Sorry, Midd.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF CODE LISTINGS	xi
ABSTRACT	xii
 Chapter	
1 BACKGROUND AND INTRODUCTION	1
1.1 Post-Stroke Rehabilitation	1
1.2 Robot-Mediated Rehabilitation	3
1.3 Wearable Robots	4
1.4 Cable-Conduit and Other Remote Actuation Transmissions	5
1.5 Modeling & Control Efforts	8
2 DYNAMIC MODEL	13
2.1 Model Formulation	13
2.1.1 Discretization	16
2.1.1.1 Scenario 1: Mobile segment	17
2.1.1.2 Scenario 2: Slack Segment	19
2.1.1.3 Scenario 3: Stationary Segment	20
2.1.1.4 Scenario 4: Partially mobile segment	20
2.1.2 Motion Propagation	20
2.2 Model Implementation	22
2.3 Solver Methodology	24
2.3.1 Background	24
2.3.2 Linear Formulation	25

2.3.3	Proof of Existence and Uniqueness of Solutions	29
2.4	Model Validation	35
2.4.1	Experimental Apparatus	35
2.4.2	Parameter Estimation	36
2.4.3	Experiment 1: Behavioral Verification	37
2.4.4	Experiment 2: Position-Force Mode	39
2.4.5	Experiment 3: Position-Position Mode	42
2.4.6	Validation Results	44
3	MODEL-BASED STATE ESTIMATION IN REAL TIME	45
3.1	Kalman Filtering	46
3.1.1	Conventional Extended Kalman Filter	50
3.2	Redundant Sensor Fusion	53
3.2.1	Direction of Motion Known	54
3.2.2	Direction of Motion Unknown	56
3.2.3	Online Parameter Estimation	60
4	CONTROLLER DESIGN	64
4.1	Stiction Compensation	64
4.2	Dynamic Friction Compensation	65
4.3	Simulated Controller	68
4.3.1	Simulation 1: One Cable	68
4.3.2	Simulation 2: Two Cables	72
5	DISCUSSION	75
5.1	Summary of Contributions	75
5.1.1	Limitations	76
5.2	Research Significance	76
5.3	Future Work	77
5.4	Closing Remarks	78
	BIBLIOGRAPHY	79

Appendix

SEGMENT VELOCITY APPROXIMATION 86

LIST OF TABLES

2.1	Examples of functional forms compatible and incompatible with linear solution method	30
2.2	System parameters used for model simulation	37

LIST OF FIGURES

1.1	Photographs of several existing non-autonomous gait retraining exoskeletons	1
1.2	Schematic diagram of a monodirectional cable-conduit transmission	5
1.3	Schematic diagram of cable-conduit transmission in bidirectional pull-pull configuration	6
1.4	A cable-conduit based robotic ankle exoskeleton	7
1.5	Kaneko et al.'s original lumped-mass cable-conduit model	9
1.6	Schematic diagram of of the model of Agrawal et al.	11
2.1	Force balance in a discrete cable segment	14
2.2	Schematic of the proposed new model	21
2.3	The nonlinear sign function.	26
2.4	Performance benefits of novel linearizing solver	29
2.5	Bounding the quantity τD	33
2.6	Bounding the quantity $u_{1,t-1}$	33
2.7	SolidWorks rendering and photo of benchtop cable-conduit transmission test apparatus	36
2.8	Input torque waveform for Experiment 1	37
2.9	Results from Behavioral Verification Experiment 1a	38
2.10	Results from Behavioral Verification Experiment 1b	40

2.11	Force-Position Domain Representation of Experiment 2	41
2.12	Input-output relationships from Experiment 2	42
2.13	Bias in Experiment 2 Residuals with respect to velocity	43
2.14	Results from Experiment 3	44
3.1	Model chatter caused by sensor noise	45
3.2	One term of a Jacobian linearization of the model equations	53
3.3	Source of non-zero values in Jacobian elements	54
3.4	Basic sensor-based state estimation	55
3.5	Effect of sensor count and their quality on estimation error	55
3.6	Time-series behavior for the simple estimator	56
3.7	Extrapolation error in a sensor-fusion scheme	57
3.8	Prediction errors for a sensor at a given location	59
3.9	Divergence of tension slope regressions due to incorrect friction parameter	62
4.1	Model-based linear controller	65
4.2	Model-based disturbance observer	66
4.3	Model-based feed-forward friction compensation	67
4.4	Simulated controller performance	69
4.5	Simulated controller phase	70
4.6	Controller behavior in inertialess environment	71
4.7	Two-cable controller simulation	73
4.8	Virtual Environment Accuracy in controlled two-cable system	74

A.1	Segment Velocity Determination from Node Velocities	86
-----	---	----

LIST OF CODE LISTINGS

2.1	Pseudocode of Cable-Conduit Model implementation	23
2.2	Generator function for allowable sign permutations.	27
4.1	Static friction compensation logic	65

ABSTRACT

In the field of rehabilitation robotics, wearable systems need to optimize mass distribution to reduce dynamic loading on distal joints. When this optimization requires actuators be placed at locations distant from the joint they actuate, a cable-conduit transmission can be used to transmit forces to where they are needed. While desirable for their low mass and flexibility, cable-conduit transmissions introduce large frictional forces into the system that may result in the non-natural motion of the interacting individual. Model based controllers can reduce the dynamic loading introduced by cable-conduit transmissions, but presently there exists no model suitable for controller use when the environment is non-passive.

This work presents a new dynamic model of a cable-conduit transmission designed to model interactions with non-passive environments, such as a human physically interacting with a cable-conduit actuated robot. This non-linear model features bi-directional propagation of motion within the transmission and captures the associated frictional phenomena. The model is validated in a physical prototype through experiments involving interaction with a human subject. To enable use in a real-time controller application, a novel solution method, that operates 2 orders of magnitude faster than traditional methods, is described and proven.

The proposed model is evaluated for use as a tool to estimate states and mechanical parameters during system operation, combining information from redundant noisy sensors. Finally, designs for control schemes utilizing the model are discussed, and several simulations are conducted to demonstrate improved system performance when using a model-based controller. Development of this model will allow for future researchers to include model-based low level control in their devices, improving performance.

Chapter 1

BACKGROUND AND INTRODUCTION

1.1 Post-Stroke Rehabilitation

For the 240,000 physical therapists in the United States [3] and millions more worldwide, improving clinical outcomes for their patients is a perpetual struggle. For individuals among the 795,000 annual stroke patients who suffer hemiparesis [4], the path back to a normal life through physical therapy is long and difficult. As many as 50% of survivors will never recover the ability to walk at a normal speed [5], or use of the paretic arm to complete day-to-day tasks (like opening a jar) [6], leading to a severely diminished quality of life, including the possibility of dependence [7]. Regardless of the magnitude of recovery achieved by a given individual, the process of motor retraining is long and costly [8].

A typical rehabilitation program to restore motor function combines an array of different training methods intended to maximize improvement of capability and

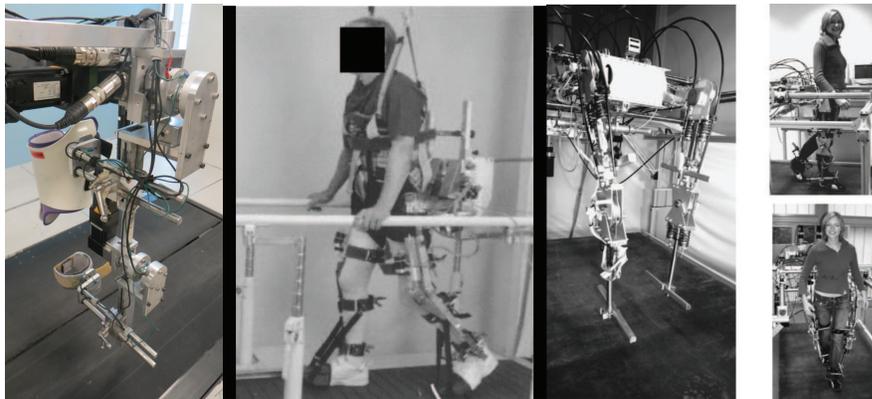


Figure 1.1: Examples of lab-based gait retraining exoskeletons. From left to right, the ALEX, the LOKOMAT [1], and the LOPES [2].

transfer of these improvements from training to everyday tasks. One category is to repeatedly perform a given task under supervision, for example the sit-to-stand task [9]. Training of this type has been shown to improve performance of the specific task practiced [10], but may not generalize to similar but nonidentical movements after stroke. It is believed that recovery in repetitive movement exercises is accomplished through *neuroplasticity*. Since the motor deficiency is caused by lesioning of regions of the brain responsible for motion, recovery of function requires that the nervous system establish a new pathway to control a given motor task [11]. Repetitive motion training is known to strengthen endogenous plasticity processes in the brain [12], and may also invoke changes to the neural representation of the task, although only some aspects of the mechanism employed to establish these new pathways are understood [13]. However, studies using fMRI have shown that restoration of a given function is correlated with structural changes in a new, uninjured region of the brain [14, 15].

For tasks the patient cannot perform unassisted, training with support may be used. An alternative technique for gait retraining is partial body weight support on a treadmill [16, 17]. During this type of training, therapists help guide limbs as needed to assist motion. These specialized retraining sessions are accompanied by neurodevelopmental training (such as that of the Bobath approach [18]) where therapists attempt to provide external forces and torques to elicit responses that cause a patient to compensate such that their motion more closely follows that of a healthy individual. Other approaches can be used to target specific problems; for example learned disuse of a paretic arm can be addressed by constraint-induced movement therapy [19] restricting use of the healthy limb. While innumerable studies are conducted in search of new, more effective strategies, systematic reviews [20] show that these are among the most clinically effective strategies known. Still, ongoing research in motor learning and neuroplasticity as pertain to stroke [21, 22, 23] guides the development of new strategies (such as split-belt training [24]) that will hopefully improve long-term prognoses for patients.

To the end of improving clinical outcomes, current efforts focus primarily on 1)

increasing the availability and dosage of therapy, on the accepted premise that increased training time correlates to improved outcomes [25], and 2) improving the effectiveness of training provided at invoking motor relearning. Robot-mediated rehabilitation is a means being studied to further efforts toward both of these goals.

1.2 Robot-Mediated Rehabilitation

Since a large portion of rehabilitation programs is based on repetitive, therapist-assisted motion, roboticists began developing systems to automate the mechanical tasks a therapist would perform during a rehabilitation session. Doing so would lower the burden on the therapist, increasing availability of therapy, while providing improved repeatability for the assistance administered. For a number of years, the state of the art in robotic lower-limb rehabilitation consisted primarily of devices that live in the lab and provide training as needed during sessions. While the exact effectiveness of robotic gait trainers, like the ALEX [26], LOPES [2], and LOKOMAT [27] (Fig. 1.1), is the topic of current research, a systematic review found past efforts roughly equal in effecting positive clinical outcomes when compared to traditional therapist mediated training [28]. Since clinic-based robot-mediated rehabilitation has not yet been shown to reliably improve outcomes when replacing traditional therapy, the current use strategy is for robot mediated training to increase availability and quantity of training to an individual. Concurrently, effort is being placed on development of time-variant control strategies designed to challenge individuals at a level commensurate with their current ability [29]. Some of these efforts strive to increase gait speed or stride length [30] while decreasing asymmetry during training, while other strategies seek to use the after-effects of adaptation [31] to a perturbed environment (such as in split-belt treadmill training [24]) to effect the desired change after doffing the device. It is thought that these subject adaptive control strategies might improve clinical outcomes beyond the expected improvement from dosage increases alone.

1.3 Wearable Robots

Toward the end of improving availability of rehabilitation robots, and consequently dosage of robot-assisted therapeutic programs, roboticists' goal has shifted from developing a stationary robotic system that can provide sessions of training in a lab or clinic, to creating autonomous, user-friendly devices that can be worn to provide continual assistance or training during day-to-day tasks. A salient caveat is that for the system to be truly wearable, serious consideration must be given to effects of the system's own weight and configuration [32]. Power source, actuator, mechanical linkages, and control electronics all impose significant loads on a wearer, and the interconnections between them provide ergonomic challenges.

Some of these aspects have proven easier to address than others; for example wiring for sensors can easily be woven into clothing. When possible, system components like batteries and control electronics can be placed at the hip or in a backpack, where the effects of their mass on natural motion is minimized. This design philosophy can be seen in almost all wearable devices, including the commercially successful ReWalk system [33], which is currently undergoing clinical trials for gait rehabilitation of individuals suffering spinal cord injury [34]. The field of soft robotics has evolved to create devices that use specially designed clothing or fabric straps in lieu of a rigid frame [35, 36] in an effort to reduce system mass. Additionally, when delivering assistive torques to a distal joint, like the ankle, inertia from even a small co-located actuator is incredibly detrimental. Even a small mass invokes large changes to gait kinematics and increases in joint moments required to walk [37]. In an autonomous device, these drawbacks are typically too large for the device to have a positive metabolic effect when supported by the wearer. To date, the only autonomous, active robotic devices showing net metabolic reductions have been small monoarticular ankle devices [38, 39], and few robotic devices altogether have seen clinical adoption.

Ideally, the actuator would be placed alongside the control hardware and power source at a central location as to minimally impact human-robot system energetics [37], and force would be transmitted remotely to the joint of interest. However, to

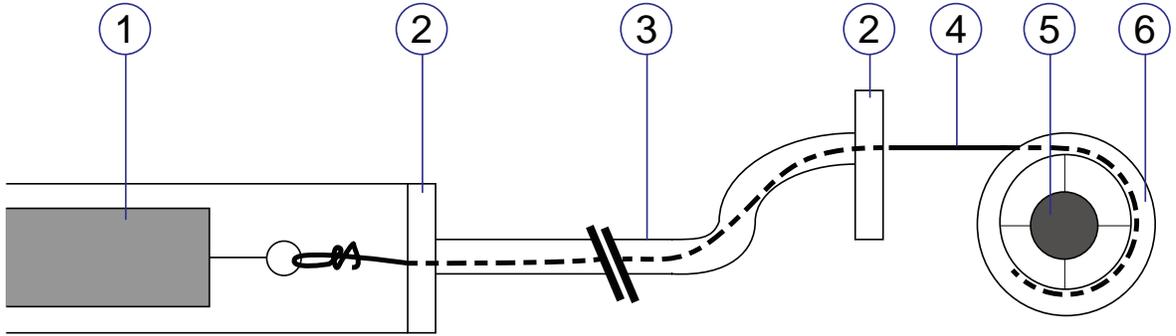


Figure 1.2: Schematic diagram of a monodirectional cable-conduit transmission for linear motion. An arbitrary device (1) is manipulated by inner cable (4) as motor shaft (5) turns pulley (6). Mounting plates (2) hold outer conduit (3) in place on both ends.

accomplish this requires the use of a *remote actuation transmission*, which presents its own set of challenges.

1.4 Cable-Conduit and Other Remote Actuation Transmissions

A straightforward implementation of a remote actuation transmission might involve use of existing and well-developed hydraulic or pneumatic systems. However, these are not well suited to systems designed to be worn by humans. The high pressure required for pneumatic systems to have sufficient power density presents an intrinsic safety risk, while also meaning that there must be either massive gas ballasts or a constantly running compressor. Hydraulic systems suffer from similar issues, while they can have excellent power density, hydraulic fluid leaks are a common ailment that is unacceptable in a wearable system. A direct belt, chain, or cable drive is also ineffective, since the distance and path between actuator and joint is variable throughout motion and including pulleys along the way increase the complexity of the device while making it less safe and reliable.

A popular and effective solution for this particular remote actuation challenge has become the cable-conduit transmission [35, 40, 41]. Shown in Fig. 1.2, a cable-conduit transmission consists of a mobile inner cable routed through a fixed outer

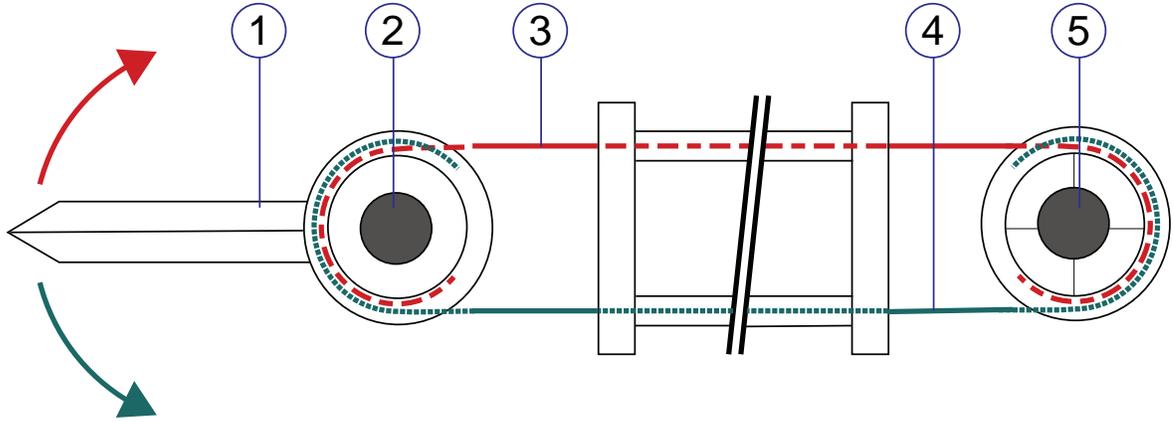


Figure 1.3: Schematic diagram of a cable-conduit transmission in bidirectional pull-pull configuration for rotary motion. Clockwise motion of motor shaft and pulley (5) engages the upper cable (3,red) resulting in clockwise motion of output shaft and pulley (2) and corresponding motion of the attached device (1). Conversely, counterclockwise motion engages the lower cable (4,green).

sheath. The cable can be connected to a pulley at the end for rotary motion, or directly to the end of a device for linear motion. Since a single cable cannot reliably transmit compressive forces in any configuration, when bidirectional control of motion is required, these conduits can be arranged in tandem to form a *pull-pull configuration* (Fig. 1.3), where each of a pair of cable-conduit systems is responsible for motion in a single direction.

A number of systems currently being used for research purposes use cable-conduit actuation to manipulate joints. Researchers at Carnegie Mellon University have developed a monoarticular ankle exoskeleton testbed actuated via a unidirectional cable-conduit transmission [40], shown in Fig. 1.4. This system was used to great effectiveness in a human-in-loop optimization scheme to reduce the metabolic cost of walking by as much as 24% when only actuating one ankle [39]. In this scheme, the optimizer chose when to provide assistance and when to remain in *transparent mode*, whereby the device attempts to impart no force on its wearer, accomplished in this system via slacking of the single cable. Another example is soft exosuits developed

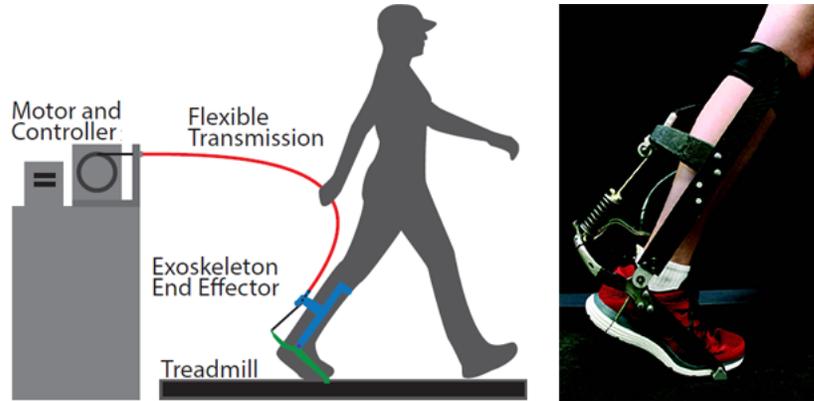


Figure 1.4: A monoarticular cable-conduit driven exoskeleton used in [39] to reduce the metabolic cost of walking.

by researchers at Harvard University, where the exosuit itself is made of a lightweight fabric and cables provide actuation of the suit beyond its natural elasticity [35].

These transmissions are selected because have a number of desirable properties. They are typically lightweight, as a relatively thin steel cable or fabric string can withstand large tensions. Material selection is flexible, as the only poor choices are those that would allow the conduit to collapse and pinch the cable, or that would have exceptionally large coefficients of friction between cable and conduit. Cable-conduit transmissions also provide a mechanically flexible means of transferring power, as the conduit is free to change shape during motion.

These benefits beget symmetrically significant hindrances and drawbacks, however. First, when compared to other transmissions, the efficiency of cable driven systems is poor. This is due to the presence of large, nonlinear frictional effects between the cable and conduit that vary with cable path, tension, and material properties. It has been shown that this causes a generic cable-conduit system to display input-dependent stability properties [42]. This instability can be attributed to backlash in the system which needs explicit compensation, else the temporal delay between force applied at the input and propagation of that force to the output caused by the friction itself [43, 42] will severely limit available control bandwidth [44]. Lastly, displaying

zero force is nontrivial in a pull-pull configuration since users cannot rely on slacking the cable, and interaction must be guided by sensor feedback [45].

In practice, this set of issues means that simple feed-forward control of systems integrating cable-conduit transmissions is infeasible, as friction will cause both large tracking errors and backlash that leads to diminished control bandwidth and possible instability. The transmission’s own dynamics are also reflected to the user—for example, when an actuator applies zero force, all the friction in the system is displayed at the output. To proceed, roboticists must look at means of feedback control, where having an accurate model of the system is valuable both to guide controller design and allow for state observation.

1.5 Modeling & Control Efforts

Most previous efforts to control for friction in cable-conduit transmissions have been focused on either model-free feedback or compensation laws based on highly simplified models. Townsend et al. worked with explicit friction compensation, coupled with proportional-integral force feedback control, but found this resulted in limit cycle behavior due to static friction in the system [43], such that desired system behavior could not be achieved. The earliest modeling efforts for cable-conduit systems were not for use as a transmission, but were done for the sake of electricians attempting to understand difficulty in drawing electrical wire through conduits [46]. After doing work in these early identification efforts, Kaneko et al. developed the first dynamic models for the cable-conduit transmission [47, 42]. Taking the most popular approach when trying to model a new system, they developed a lumped-mass formulation resulting in a number of spring-mass-damper systems in series, shown from their original paper in Fig. 1.5. The core of this model was simple Coulomb friction,

$$F_\mu = \mu \cdot F_n \tag{1.1}$$

Where F_μ is the frictional force, μ is the frictional coefficient, and F_n the normal force between cable and conduit. The elastic coefficient k of the springs in the model

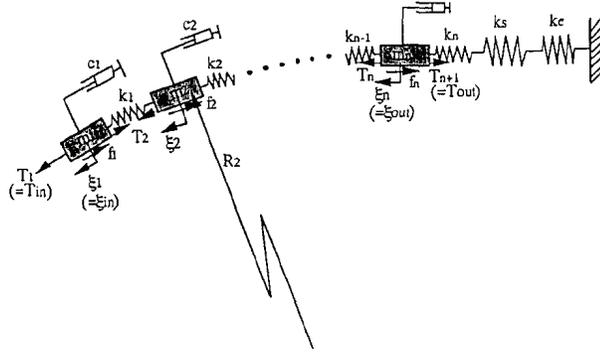


Figure 1.5: As presented in [48], the original lumped-mass model developed by Kaneko et al. consisting of a number of mass-spring-damper systems chain together in series.

is adjusted as the adimensionalized friction constant λ as, $k = k_{\text{cable}} \frac{\lambda}{\exp(\lambda) - 1}$. This formulation presented significant limitations, however. First and foremost, the mass-spring-damper system used to simulate the cable cannot replicate the phenomenon of partial motion due to its linear nature. It retained the undesirable property that curvature of the cable must remain constant throughout simulation, and additionally could only accept environments that could be represented by a spring, as a consequence of the linear nature of the model. Changing pretension in the cable was also nontrivial, and pretension needed to be constant throughout the entire cable. In practice, pretension is configuration specific and varies through the cable since friction still effects the system during tensioning, making this a difficult constraint to work with.

This model would be improved by Palli & Melchiorri, [49], choosing to include a dynamic Dahl friction model [50, 51]

$$\frac{\partial F}{\partial x} = \sigma_0 \cdot \text{sign} \left(1 - \text{sgn}(v) \frac{F}{\mu F_n} \right) \left| 1 - \text{sgn}(v) \frac{F}{\mu F_n} \right|^{\delta_D} \quad (1.2)$$

instead of the Coulomb model inherent to Kaneko et al.'s implementation. In this model, σ_0 is contact stiffness in the static condition ($v = 0$), μF_n is the static friction, F is the frictional force being studied, and δ_D is a free positive parameter that can be tuned to adjust the shape of hysteresis [52].

The specific nature of frictional interactions in the cable would later be studied in-depth by Do et al. [53, 54, 55] who determined that a specially modified Bouc-Wen frictional model most accurately matches the behavior of cable-conduit systems in pull-pull configuration. This model is given by the equation

$$F_f = k_x(\dot{x}, \ddot{x})x + k_\zeta\zeta + \nu\dot{x} + F_0 \quad (1.3)$$

where F_f is the frictional force, F_0 is a constant offset, ν is a viscous friction coefficient, and dotted variables represent time derivatives of the position state x . ζ is a nonphysical internal state updated differentially by

$$\dot{\zeta} = \rho(\dot{x} + \lambda\text{sign}(\ddot{x}) - \sigma|\dot{x} + \lambda\text{sign}(\ddot{x})||\zeta|^{n-1}\zeta + (\sigma - 1)(\dot{x} + \lambda\text{sign}(\ddot{x}))|\zeta|^n) \quad (1.4)$$

where $\rho > 0, \sigma \geq 0, n \geq 1$ are shaping factors, and the positional scaling factor k_x is given by

$$k_x(\dot{x}, \ddot{x}) = \frac{\bar{k}_x}{\exp(2\dot{x}) + 1} (\exp(2\dot{x}) + \text{sign}(x)\text{sign}(\ddot{x})) \quad (1.5)$$

However, models based on this formulation still fail to capture the phenomenon of partial motion within the cable, despite tracking well during motion. For applications in human-interacting robotics, capturing accurately the behavior as the transmission engages is key for many control objectives, including transparent behavior.

More recently, a number of groups have developed nonlinear compensation laws for use in their feedback controllers [53, 56, 57], which provide significant performance benefits above simple standard linear feedback controllers, such as PID control. However, they are based on simplified models, which assume a constant cable configuration or interaction with a passive environment. Both of these assumptions are counterproductive for use in human-interacting robotics, as the human will invariably add energy to the system during motion when wearing a robot, and almost every movement a cable-conduit transmission could be used to assist would see the cable's bend radius vary over its duration. To date, there have been no published efforts to use a nonlinear cable conduit model as an observer for a state feedback controller.

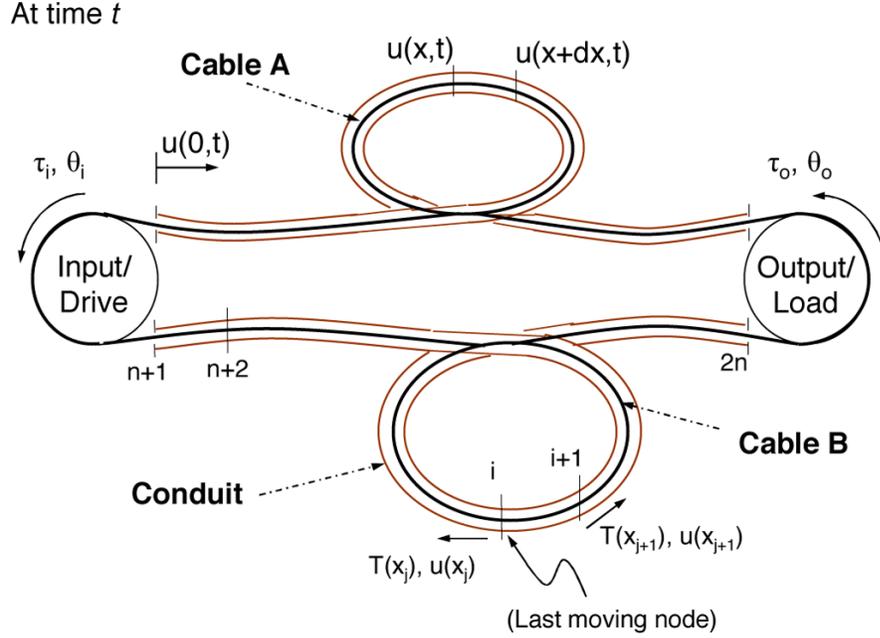


Figure 1.6: Schematic of the distributed model developed by Agrawal et. al [58] featuring nonlinear friction applied to an elastic cable.

Realizing the implications of the inherent limitations of lumped-mass models, an alternative formulation was proposed by Agrawal et al. [59, 58] Instead of approximating the cable as a series of mass-spring-dampers, the cable itself is considered to be a massless series of elastic elements, connected by nodes. By integrating friction over any given segment, the difference in tension between nodes can be obtained, and since the cable itself is a spring, the displacement of nodes follows from Hooke's Law. A schematic of this formulation is shown in Fig. 1.6. In this way, the behavior of the cable-conduit can be studied free of some of the restrictions of the lumped-mass formulation, however, the model was applied only to demonstrate the case of interaction with a passive environment. This follows from two model properties: first, their system of equations is underdefined in the absence of an explicit relationship f between the tension at the output T_O and the displacement of the end of the cable x_O such that $T_O = f(x_O)$; and second, an asymmetry in the model that only allows motion to propagate in one direction. In a nonpassive system, part of the cable must begin

to move if force from either direction exceeds static friction, but their implementation can only consider force from one direction. Additionally, this formulation involves non-linear effects in the state variables, meaning that solution of states is computationally expensive.

My work extends the formulation of Agrawal to allow its use in a controller. I have first modified the formulation to allow non-passive environments, such as would be encountered in a human-interacting application. This required the introduction of *bidirectional motion propagation*, a feature not present in any previous model. The specifics of the new formulation are presented in Chapter 2. Additionally, I have developed a novel method to solve the nonlinear equations present in this formulation in $\mathcal{O}(n^2)$ time. The innovation in solution of the system equations allows us to explore the use of this model as a state estimator, as presented in Chapter 3. The model is considered for use in friction compensation control schemes in Chapter 4.

Chapter 2

DYNAMIC MODEL

In this chapter, I will present a full derivation of the proposed dynamic model of a cable-conduit transmission, and describe the software implementation I have developed. Additionally, this chapter contains information about a method of reducing the nonlinear system to a linear one for purposes of rapid solution, and validation experiments against a physical system.

2.1 Model Formulation

We begin our model formulation by assuming that the cable is a massless elastic element, with a constant stiffness k_c over its entire length. The assumption that the cable has no mass means that we will be neglecting the effects of cable inertia on the dynamics of the system. Secondly, we assume that motion is one-dimensional, only allowing motion of the cable along the axis of the conduit, and not throughout the cross section of the conduit. This assumption means that we presume that the contact force is solely a function of tension and bend radius, and neglect any contributions due to the transverse location of the cable within the conduit. Our final assumption is that when a segment of the cable is not in motion, its stationary state persists until static friction is overcome by the action of the adjacent segments. For a stationary cable, neither tension nor motion can propagate. This assumption lets us use currently stationary nodes as a boundary condition to solve for motion of a partially-moving cable.

To begin solving for system dynamics, we consider an infinitesimal segment of the *conduit*¹. Because of our assumption of neglecting the inertia of the cable, Newton’s first law will be written as $\sum F = 0$, where F are all the external forces applied to the segment. Due to our assumption of no transverse motion, we only have to worry about forces along the axial direction of the conduit. In our model, we have only three forces acting on the cable over this segment, as shown in Fig. 2.1: tension from one end, tension at the other end, and the frictional interaction between cable and conduit.

If we denote one end of our infinitesimal segment to be at location x in the axial direction of the conduit, then the other side’s position is given by $x + dx$. Let the tensions at their respective ends of the segment be denoted by $T(x)$ and $T(x + dx)$, and friction $F_\mu(x)$. For this system, the equilibrium of forces along the axial direction yields

$$T(x + dx) = T(x) + F_\mu(x) \tag{2.1}$$

¹ We consider the conduit, rather than inner cable, because we need to solve for the friction through the entire conduit, and don’t need to consider parts of the cable not experiencing friction, outside of the conduit.

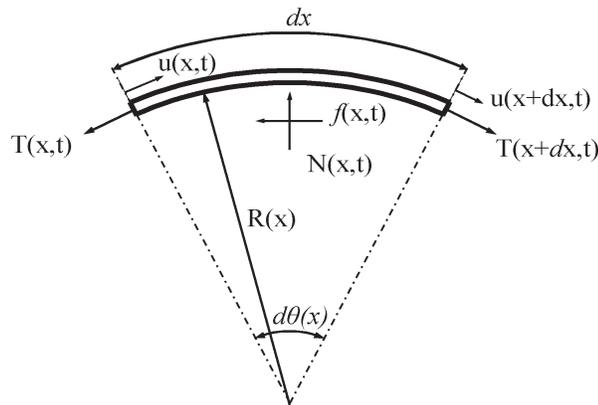


Figure 2.1: Balance of forces across a given segment of cable at location $u(x, t)$ and length dx . Tension $T(x, t)$ is applied from both ends, resulting in a normal force $N(x, t)$ to exist between cable and conduit because of the cable’s curvature $R(x)$. This results in friction $f(x, t)$ resisting the motion of the cable. From [58]

To calculate $F_\mu(x)$, we need to know the normal force $F_n(x)$ between cable and conduit over the segment. If we are given that the cable's radius of curvature is specified by the continuous function $R(x)$, we know that the segment sweeps out an arc of central angle $d\theta$ over its infinitesimal length such that $d\theta = dx/R(x)$. Geometry then gives us that

$$F_n(x) = T(x) \sin\left(\frac{d\theta}{2}\right) + T(x + dx) \sin\left(\frac{d\theta}{2}\right) \quad (2.2)$$

If we approximate that changes in tension over the differential segment are small, and take the small angle approximation for sin, we can proceed as

$$F_n(x) \approx 2 \cdot T(x) \sin\left(\frac{d\theta}{2}\right) \approx T(x) d\theta \quad (2.3)$$

If, for the moment, we assume a Coulomb friction model and let the coefficient of dynamic friction be μ_k , this gives us the frictional relationship,

$$|F_\mu(x)| \leq \mu_k F_n(x) = \frac{\mu_k T(x) dx}{R(x)} \quad (2.4)$$

where friction is represented as an inequality because it cannot be permitted to add energy to the system.

From this relationship, we can see that if the tension differential across a moving infinitesimal segment decreases below the maximum value of $F_\mu(x)$,

$$|T(x) - T(x + dx)| dx = \Delta T(x) dx < F_\mu(x) = \frac{\mu_k T(x) dx}{R(x)} \quad (2.5)$$

the cable will not move at that location. Otherwise friction will resist motion (shown here by velocity $v(x)$) at its maximum value, and provide the tension differential of

$$\Delta T dx = \frac{\mu_k T(x) dx}{R(x)} \cdot \text{sign}(v(x)) \quad (2.6)$$

To incorporate static friction in this Coulomb friction model, we must introduce the coefficient of static friction μ_s , and establish the result of the inequality:

$$|T(x) - T(x + dx)| dx \stackrel{?}{\geq} \frac{\mu_s T(x) dx}{R(x)} \quad (2.7)$$

To study the motion of the inner cable, we revisit our assumption that the cable is represented by an elastic element of stiffness k_c . Then, if we define the cable displacement at axial coordinate x to be $u(x)$, Hooke's Law $T = k_c * du/dx$ in conjunction with Eqn. 2.6 and our assumption that the tension and position of stationary nodes can't change combine to give us that

$$\frac{d^2u(x)}{dx^2} = \frac{du(x)}{dx} \frac{\mu_k}{R(x)} \text{sign}(v(x)) \quad (2.8)$$

At this point, it is convenient to address the issue of cable slacking. While the lumped mass formulation would exert negative tension under cable compression, this behavior is nonphysical². We can instead assert that the tension of any compressed segment should be identically zero, and restrict our calculated solutions to cables under tension.

2.1.1 Discretization

The differential equations representing this model have no closed-form analytical solution, so it is necessary to use a computational approach based on discretization. To proceed, we split the cable into a series of n sub cables, which we will call segments. The interconnections between segments will occur at nodes, of which there are $n+1$. As with all finite element methods, increasing n improves accuracy at the cost of computational effort. If the total length of the cable is L , it is straightforward that we can divide it into equal segments of $\Delta x = L/n$. While the examples and implementation here do specify segment length equally, this solution is not necessary, and different implementations may be followed to optimize the use of computational resources. In practice, there would be benefits to having large segments where curvature is most nearly constant,

² While some compressive force can be transmitted due to rigidity of the cable, the magnitude should be much smaller than forces transmitted through tension. The lumped mass formulation has no element to enforce this asymmetry. Additionally, this causes the cable to have large transverse deviations within the conduit, a violation of our assumptions, so we need to handle this case separately.

and smaller segments where curvature has a nonzero derivative. This would give the best numerical performance for the least computational effort.

To proceed, let us define T_i the tension at node i , and u_i the relative displacement at the same node for a given instant. We must consider the radius of curvature to be a constant over this finite element, via some evaluation of the function $R(x)$. A logical choice for R_i , this chosen value, would be the average value of the function $R(x)$ on the domain of the element. We must also take the segment velocity to be a constant rather than a continuously variable function. Previous works have taken $v_i = u_i(t) - u_i(t - 1)$, which works in the case of passive environments. However, the bidirectional nature of motion propagation in a model for interaction with nonpassive environments means that this approach can result in non-passivity of the model itself. We instead choose the mean of both nodes velocities, $v_i = \frac{1}{2}((u_i(t) - u_i(t - 1)) + (u_{i+1}(t) - u_{i+1}(t - 1)))$, a discussion of this choice is provided in the appendix.

We can apply these assumptions to our differential equations, and integrate as appropriate to obtain discrete behavior. Note that all integrations are performed in the forward direction here; there exist symmetric reverse direction integrations that must be performed for bidirectional motion propagation³. We have four cases to consider for the motion of any given cable,

2.1.1.1 Scenario 1: Mobile segment

This behavior makes up the core of the model. To obtain the system equations, we must integrate our differential equations over a finite segment. To enable this integration for arbitrary curvature, we must assume the curvature to be constant over the segment, and will notice that the segment must have a constant velocity over its length. We first integrate Eqn. 2.6 over an arbitrary segment from x_i to $x_i + dx$ as

$$\int_{x_i}^{x_i+dx} \frac{\frac{dT(x,t)}{dx}}{T(x,t)} dx = \int_{x_i}^{x_i+dx} \frac{\mu \cdot \text{sign}(v(x_i, t))}{R(x_i)} dx \quad (2.9)$$

³ However, the only effect reversing the integration direction has is that indices flip direction, and we go in descending order, $N + 1, N, N - 1, \dots$

This integration yields the result

$$T(x_i + dx := x_{i+1}, t) = T(x_i, t) \exp \frac{\mu dx \cdot \text{sign}(v(x_i, t))}{R(x_i)} \quad (2.10)$$

The form of this equation is of particular interest, it relates the tension at the start of a given segment to the tension at the end by an adimensional constant of proportionality that is greater or less than one depending on the direction of motion in the cable. Returning to our differential Hooke's law, we can also perform an integration relating this tension change to the change in axial displacement over the segment,

$$\int_{x_i}^{x_i+dx} \frac{du(x, t)}{dx} dx = \int_{x_i}^{x_i+dx} k_c^{-1} T(x_i, t) \exp \frac{\mu dx \cdot \text{sign}(v(x_i, t))}{R(x_i)} dx \quad (2.11)$$

which has the closed form solution given by

$$u(x_i + dx, t) - u(x_i, t) = \frac{T(x_i, t)R(x_i)\text{sign}(v(x_i, t))}{k_c\mu} \left(\exp \left(\frac{\mu dx \cdot \text{sign}(v(x_i, t))}{R(x_i)} \right) - 1 \right) \quad (2.12)$$

This result follows our expectations based on Hooke's Law itself; the stretch of the cable is proportional to the inverse of the cable's elastic constant, times the change in tension (force) over the same cable segment. If we adopt the compact notation $T_i := T(x_i, t)$, $u_i := u(x_i, t)$ and $S_i := v(x_i, t)$, and make appropriate 'copies' for each of the N mobile segments (And thus, $N + 1$ nodes) being considered, we then form a

system of equations based on the above results (2.10 and 2.12)

$$\begin{aligned}
0 &= T_1 \exp\left(\frac{\mu dx S_1}{R(x_1)}\right) - T_2 \\
0 &= T_2 \exp\left(\frac{\mu dx S_2}{R(x_2)}\right) - T_3 \\
&\vdots \\
0 &= T_N \exp\left(\frac{\mu dx S_N}{R(x_N)}\right) - T_{N+1} \\
0 &= u_1 - u_2 + T_1 \frac{R(x_1) S_1}{k_c \mu} \left(\exp\left(\frac{\mu dx \cdot S_1}{R(x_1)}\right) - 1 \right) \\
0 &= u_2 - u_3 + T_2 \frac{R(x_2) S_2}{k_c \mu} \left(\exp\left(\frac{\mu dx \cdot S_2}{R(x_2)}\right) - 1 \right) \\
&\vdots \\
0 &= u_N - u_{N+1} + T_N \frac{R(x_N) S_N}{k_c \mu} \left(\exp\left(\frac{\mu dx \cdot S_N}{R(x_N)}\right) - 1 \right)
\end{aligned} \tag{2.13}$$

When this system is augmented by the two required constraint equations, which set a state variable at location a to the measured value A , such as given by

$$u_a = A, \tag{2.14}$$

then the system is fully defined and has a single unique solution for all state variables, under one condition: at least one constraint must act upon a displacement state variable, otherwise the tension equations will be overdefined ($N + 1$ equations in N variables) and the displacement equations underdefined ($N - 1$ equations in N variables).

2.1.1.2 Scenario 2: Slack Segment

When we have that two consecutive node positions, u_i and u_{i+1} , such that $u_{i+1} - u_i < 0$ then the cable is slack, as its length is less than its resting length. In this state, the tension of the slack nodes $T_{i,i+1}$ is identically zero, and no further calculation is required to output the state at this time⁴.

⁴ It may be beneficial to perform the calculations as if the segment were mobile anyway, and only output 0. This makes it such that at such a time where the cable is no longer

2.1.1.3 Scenario 3: Stationary Segment

When a segment is entirely stationary, it cannot transmit any tension and it (by definition) isn't moving. The tensions and displacements can merely be copied over from the previous time, as

$$\begin{aligned}
 T_i(t) &= T_i(t - \Delta t) \\
 T_{i+1}(t) &= T_{i+1}(t - \Delta t) \\
 u_i(t) &= u_i(t - \Delta t) \\
 u_{i+1}(t) &= u_{i+1}(t - \Delta t)
 \end{aligned}
 \tag{2.15}$$

2.1.1.4 Scenario 4: Partially mobile segment

In this scenario, one end of the cable segment has overcome static friction and is in motion, but the other has not yet. From our assumptions, this stationary node cannot have its tension nor its position change. It therefore provides a boundary condition for solution of the segment in question, acting as a grounded spring according to Hooke's law. However, since we don't know the tension distribution over the partially moving segment we are forced to take an approximation by averaging the tensions at the mobile and stationary node. Then if the last moving node is k , this boundary condition is given by

$$u_{k+1} - u_k = \frac{T_k + T_{k+1}}{2k_c} dx
 \tag{2.16}$$

which is used to augment the system given in Eq. 2.13, with $N = k - 1$ in place of one of the constraint equations.

2.1.2 Motion Propagation

As the state of the cable is updated, it is necessary to keep track of what discrete nodes have overcome static friction and begun to move. Of particular interest is the node which exists on the boundary of both a mobile and stationary segment, which we

slack, the simulation can continue normally instead of having to deal with special contingencies for 'un-slacking.'

will refer to this as the *last moving node* (LMN). In the case of this model, we allow for non-passive environments. An immediate consequence is that motion can propagate from either end of the cable, and there is nothing to forbid motion from propagating in both directions concurrently. For this reason, we must keep track of *two* last moving nodes; one that propagates from the proximal end of the cable to the distal one, and one at the distal end of the cable, that propagates toward the proximal side in response to applied force/motion.

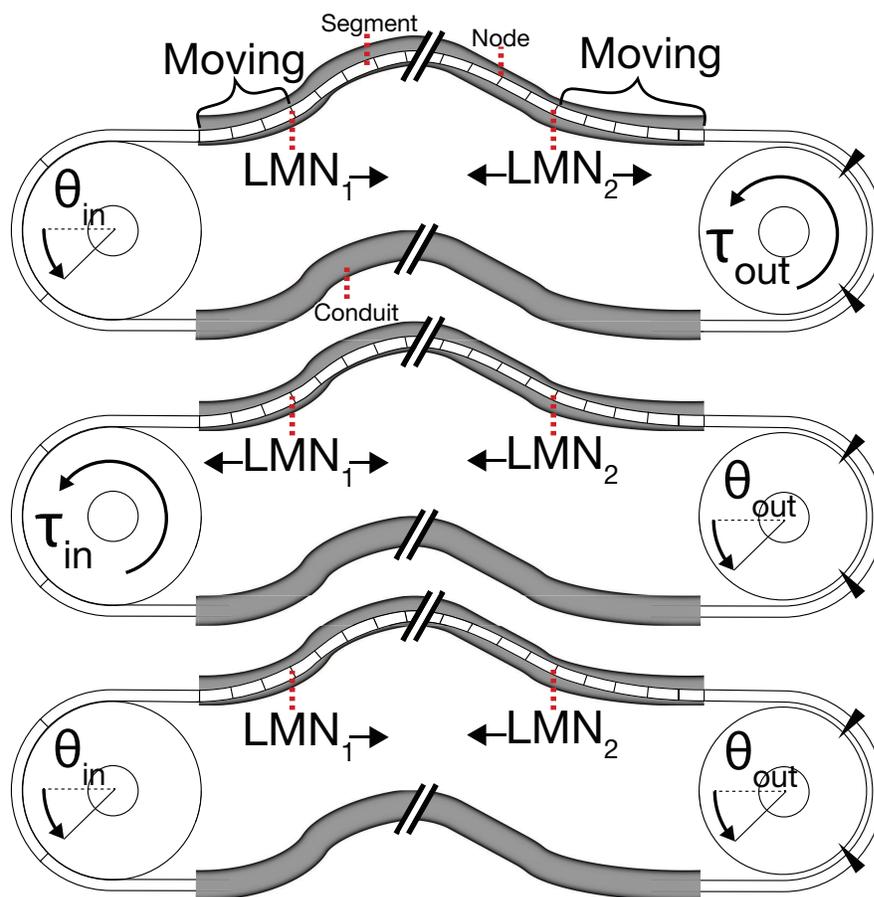


Figure 2.2: Schematic diagram of the proposed cable-conduit actuation model, featuring all 3 operating modes (Position-Torque, Torque-Position, Position-Position). Note the model's ability to change the number of mobile cable segments at both ends of the conduit, as appropriate, regardless of input mode.

We can obtain the propagation inequality through the same integration technique used to convert the differential equation into the discrete equations. In this way we find that if node k is the LMN at either cable end, that node $k \pm 1$ (For proximal and distal end, respectively) becomes the new last moving node at such a time when

$$T_{k\pm 1}S_k \geq T_k S_k \exp\left(\frac{\mu_s \Delta x S_k}{R_k}\right) \quad (2.17)$$

Note that we cannot divide across the common factor S_k , due to the inequality. Dividing across a negative value would flip the inequality, changing the meaning of the equation.

When attempting to solve for state variables, bidirectional motion propagation means that it may be needed to carry through calculations for multiple numerically isolated systems⁵. In order to do so, we need to specify at least one input, and one output variable. Additionally, at least one of these two must be a position – the position equations (Second half of Eq. 2.13) cannot be solved without at least one position to define them in the world frame since each equation calculates the change in an individual segment’s length, not any absolute positions.

Until such a time when both last moving nodes k_1 and k_2 coincide, the two distinct (possibly) mobile regions at both ends of the conduit must be treated as separate systems for solutions. When the LMNs coincide, the cable begins to move *en masse* and force/motion can be transmitted between both ends of the cable-conduit system. While this mode comprises most of the operation of the model, most of the complexity comes from outside this mode.

2.2 Model Implementation

The model has been implemented in MATLAB such that each of the two main cables participating in the pull-pull configuration is treated as an entirely distinct

⁵ At such a time when both proximal and distal end of the cable are in motion, but there is a stationary segment between them.

entity. Additionally, when a cable is undergoing partial motion, each section (proximal mobile section, stationary section, distal mobile section) is treated independently. High-level program flow is shown as pseudocode in the listing below.

Listing 2.1: Pseudocode of Cable-Conduit Model implementation

```
1 [Displacements, Tensions] = initialConditions;
2 for each(timestep)
3     for each(cable)
4         if(no motion)
5             check if motion should begin
6         if(partial motion)
7             Solve proximal subcable
8             Solve distal subcable
9             Solve stationary subcable
10        elseif(mass motion)
11            Solve entire cable
12        Check if LMN update required
13        if(LMN changed)
14            Redo cable solution
15    end
16    Calculate pulley torques, angles
17 end
```

There are some interesting details and consequences of this implementation. While the post-solution check for changes to the last moving node is performed according to Eqn. 2.17, the pre-solution check uses a different criterion based on principles of the model's construction. If we are using *position* as an input for a given cable end, we know that at least the first node MUST be mobile if $\theta(t) \neq \theta(t - \Delta t)$. Additionally,

if the input velocity is nonzero and of opposite sign than the previous iteration, then the cable has stopped and changed direction since the previous iteration. Therefore, the model must stop motion of all other nodes in the affected (sub)cable and calculate if they would have overcome static friction from the direction change.

A second interesting feature not immediately obvious from the model derivation is that changes to the last-moving node require repeating that time instant’s calculations. However, failing to do so would impose an arbitrary limit on the speed at which motion can propagate through the cable, since the LMN could only ever advance by a single node each iteration⁶. This velocity limit is given by

$$V_{P \text{ Limit}} = \frac{L}{N \cdot dt} \quad (2.18)$$

Since the true propagation speed is a function of mechanical system parameters, not our choices for the number finite elements or timestep, our model will re-calculate a new solution at any time t where the last moving node is updated, to allow for consecutive solutions where the last moving node(s) have advanced by more than a single segment.

2.3 Solver Methodology

2.3.1 Background

At a fundamental level, the model exists as a set of nonlinear equations with a single unique numerical solution; however there does not exist a closed-form solution for the system state variables. Generally speaking, numerical solutions for nonlinear equations are found via iterative guess-and-check style optimization solutions. MATLAB’s nonlinear solver, *fsolve*, uses the so-called ‘trust-region dogleg method’ for numerical solution, the details of which are outside the scope of this paper. However, the important properties of this numerical solver are as follows:

1. The algorithm provides no assurance that it will generate a solution

⁶ This follows from the fact that the tension differential between first and second non-moving nodes cannot change, so there’s no way for the LMN to advance by 2 without re-solving after it advances by 1.

2. The algorithm provides no assurance that the solution it provides, if any, is the true solution, merely that it approximates equivalence in the given system
3. The algorithm is not guaranteed to converge within any amount of time

These properties are not very good for a simulation, but even worse for use in a real-time model based controller. We know that since we have a set of dynamic equations, they must have a single unique solution that describes the system state. The possibility of finding no solution, or worse, an incorrect solution, has devastating implications for the robustness of any controller utilizing that solver. Additionally, the time costs of such an algorithm are unacceptable. In testing, even the best-case scenario where the solver is given the previous state as its starting point, the rate of solution did not approach speeds needed for controller use, as shown later in Fig. 2.4. A different approach was required for this purpose.

2.3.2 Linear Formulation

To redefine our problem statement, we take note of the form of equations in Eqn. 2.13. These equations are nonlinear solely because of the presence of the *sign* function. As such, we can introduce a simplification to bypass the non-linearity introduced by the sign function (Fig. 2.3). Recall that

$$S_i = \text{sign}(v_i) = \begin{cases} -1 & v_i < 0 \\ 0 & v_i = 0 \\ 1 & v_i > 0 \end{cases} \quad (2.19)$$

The important feature of this function is that despite a continuous range of inputs, it can assume only three distinct values. We also observe that despite the fact that there are $2N$ equations in the system describing the system state, there are only N distinct arguments to the sign function. The set of conditions to consider is then provided by the 3^N possible permutations of the values of the N sign functions included in the dynamic equations.

In Big-O notation, commonly used to describe the time cost of algorithms in terms of the size of their input, we can assign the worst-case time cost of this search operation then as $\mathcal{O}(3^N)$. Exponential scaling with the number of parameters is still less than desirable⁷, as in this case a cable with only 5 segments would have as many as 243 possible outcomes for the set of nonlinearities. There are more beneficial properties to take note of first. First, recognize that the argument to the function is *segment velocity*. If the segment velocity is zero, the segment would be stationary. But by the existence of the equation, that segment must be mobile. This contradiction indicates that we can generally neglect the possibility that $\text{sign}(v_i) = 0$,⁸ improving us from 3^N to 2^N permutations.

A second observation is that for the model at hand, many of these permutations are nonsensical. Consider a 5-segment cable whose sign functions output the set $[-1, 1, 1, 1, -1]$. This configuration cannot, in fact, exist - the lack of inertia in the model means that wave propagation is impossible, ergo motion in the cable must be

⁷ Though, far superior to no guarantee of convergence provided by a conventional nonlinear solve

⁸ With one exception – there can possibly be a "stationary" segment in a fully mobile cable if the motion direction changes within that segment, and the velocity estimation method finds it to be identically zero over the segment despite the motion.

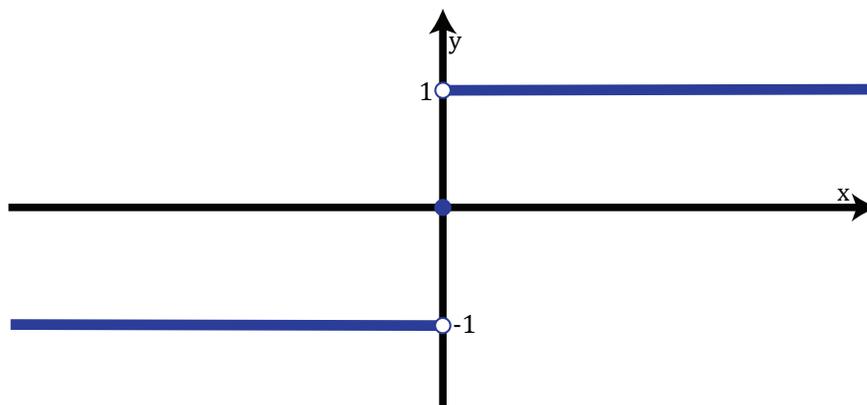


Figure 2.3: The nonlinear sign function.

described completely by 3 regions:

- Region 1: A region of cable moving in the same direction as the node at end A and extending from end A to the unknown node k_1
- Region 2: A region of cable moving in the same direction as the node at end B and extending from end B to the unknown node k_2
- Region 3: A region in the center of the cable, between regions 1 and 2, that is stationary due to friction

This constraint further rules out permutations. In fact, for $N \geq 2$, the number of permissible permutations n_p is given by $n_p = 4 \cdot (N - 1)$. We have thus developed an algorithm that selects all admissible values for the velocity of cable nodes, shown below.

Listing 2.2: Generator function for allowable sign permutations.

```
1 function [signCells] = generateSignumSet(size)
2 %Two basic cases - monodirectional in either direction
3 caseB = ones(1,size) * 1;
4 caseA = ones(1,size) * -1;
5 signCells = {caseA,caseB};
6 %Simple bidirectional motion, stretching
7 for i = 2:size
8 signCells{i+1} = [ones(1,i-1) -1*ones(1,size-(i-1))];
9 end
10 %Simple bidirectional motion, compressing
11 l = length(signCells);
12 for i = 3:l
13 signCells{l+i-2} = -1 * signCells{i};
14 end
```

```

15 % Bidirectional motion, stretching
16 % single mean-stationary segment
17 l2 = length(signCells);
18 for i = 2:size -1
19 signCells{i+l2-1} = [ones(1,i-1) 0 -1*ones(1,size-(i-1)-1)
    ];
20 end
21 % Bidirectional motion, compressing,
22 % single mean-stationary segment
23 l3 = length(signCells);
24 idx = 1;
25 for i = l2+1:l3
26 signCells{l3 + idx} = -1 * signCells{i};
27 idx = idx + 1;
28 end
29 end

```

Having reduced the possibilities for the possible values of the sign function to a small handful, it's time to exploit them. Recalling that the *only* nonlinearity present was the sign function, we start by assuming that an arbitrary one of the possibilities is correct, and substitute those values for the sign function, and proceed with solution of the now linear, algebraic set of equations. We test the solution (if it exists) for validity of the assumption made, and repeat until the correct assumption, and thus the solution for the system state is correct. While it doesn't follow trivially that this solution method works, a rigorous proof is provided in the following subsection.

As a final optimization, we recognize that changes in direction of motion are relatively infrequent for most applications. By testing the most recent set of correct assumptions first, we amortize our time cost for the assumption stage to an expected value of $\mathcal{O}(1)$. Since the time cost of linear system solutions is worst-case $\mathcal{O}(n^3)$, our

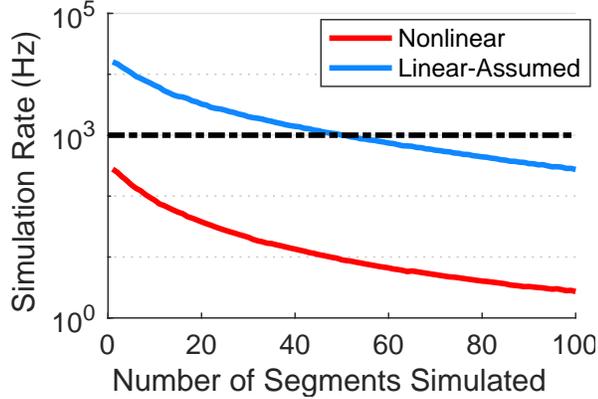


Figure 2.4: On a normal desktop computer, the presented novel solution method outperforms MATLAB’s *fsolve* by nearly two orders of magnitude.

overall worst-case time cost is therefore $\mathcal{O}(n^3 \cdot (4n - 4)) = \mathcal{O}(n^4)$, and expected time cost $\mathcal{O}(n^3)$. We can observe the effects of this solution method in Fig. 2.4 – the new linear solver outperforms the best optimized version of MATLAB’s nonlinear solver by nearly two orders of magnitude on a typical desktop PC⁹, even when MATLAB’s solver is programmed to use the previous solution as initial condition.

Since most control loops run at 1 kilohertz, the classical solver is not fast enough to run so much as a single segment simulation quickly enough, however our novel solver can run up to a 50 segment cable within the window provided by a controller cycle. Given this efficient, powerful simulation and modeling tool, we can begin to implement it into a real-time controller.

2.3.3 Proof of Existence and Uniqueness of Solutions

To begin proving the solution method, we first recall the exact nature of our equations given in Eq. 2.13. There are two types of equations, those solely in tension variables, and those both in tension and displacement variables. Momentarily, let us consider only those that deal with tensions alone.

⁹ Dell Precision T1700 Workstation, containing an Intel Xeon E3-1226v3 clocked to 3.30 GHz. MATLAB *bench* command reports the following reference times: LU: 0.1716s, FFT: 0.1251s, ODE: 0.0563s, Sparse: 0.0928s, 2-D: 0.2602s, 3-D: 0.3078s

There are then n equations in $n + 1$ unknowns, the T_i states. By our problem setup, we ensure we have at least one tension known¹⁰. The n equations take the general form

$$0 = T_{i,t}f(\text{sign}(\text{mean}(u_{i,t} - u_{i,t-1}, u_{i+1,t} - u_{i+1,t-1}))) - T_{i+1,t} \quad (2.20)$$

where f is an arbitrary nonlinear function, $T_{i,t}, u_{i,t}$ are tension and displacement state variables at discrete time instant t . The important feature of this functional form is that **all states appearing inside a nonlinear function are within a sign function as well**. That is, this solution technique only works if the equations are such that the nonlinear portion of the equation can take a fixed, countable number of discrete values. Table 2.1 gives examples of functional forms compatible and incompatible with this method.

Allowable	Not Allowable
$x \cdot f(\text{sign}(x))$	$f(x \cdot \text{sign}(x))$
$f(\text{sign}(g(x)))$	$f(x) \cdot g(\text{sign}(x))$

Table 2.1: Examples of functional forms compatible and incompatible with linear solution method

Given our specific functional form, and the nature of the sign function, it's apparent that it can only take the forms given by

$$0 = T_{i,t}f(k_{i,j}) - T_{i+1,t} \quad (2.21)$$

where $k_{i,j}$ is the value assumed under j for the sign function associated with segment i . Let us then assume every possibility for the arguments to the nonlinearity. For a

¹⁰ We permit position-position to be an input as well. However, we can proceed assuming we know at least one variable without losing generality since we ensure our full system has enough constraints to fully define the unknowns. Specifically, Position-Position provides information about tensions through the position equation

two-node system¹¹, there are then the following possibilities for the sole segment

$$\begin{aligned}
0 &= f(1)T_{i,t} - T_{i+1,t} \\
0 &= (f(0) \equiv 1)T_{i,t} - T_{i+1,t} \\
0 &= f(-1)T_{i,t} - T_{i+1,t}
\end{aligned}
\tag{2.22}$$

Choosing assumption $j = [k_{1,j}, k_{2,j}, \dots, k_{N,j}]$ from the set of all possible assumptions spanning allowable permutations of the sign functions, we place the equation in matrix form then augment with the constraint from the known value, $T_{1,t} = \tau$, to give us the linear system

$$\begin{bmatrix} 0 \\ \tau \end{bmatrix} = \begin{bmatrix} f(k_{1,j}) & -1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} T_{1,t} \\ T_{2,t} \end{bmatrix}
\tag{2.23}$$

which, for our system, has the straightforward solution for a given assumption j

$$\begin{aligned}
T_{1,t,j} &= \tau \\
T_{2,t,j} &= f(k_{1,j})\tau
\end{aligned}
\tag{2.24}$$

We then refer this closed-form solution to our other set of equations dealing in node displacements (Given in the second half of Eqn. 2.13) to obtain post-assumption node displacements, and use these displacements to calculate values for the quantity we previously assumed the sign of, segment velocity.

Now, we can compare our assumed values under assumption j to the true value. We require only that for every segment i , the assumed signs, $k_{i,j}$ equal their post calculated values, $k_{i,j} = k_{i,j}^+$. When this condition is true, we have equivalence, for each i , of the nonlinear system equations and our simplified system, since

$$f(\text{sign}(\text{mean}(u_{i,t} - u_{i,t-1}, u_{i+1,t} - u_{i+1,t-1}))) = f(k_{i,j})
\tag{2.25}$$

This is a consequence of what the assumption means: when we assume motion is toward the proximal end, friction must act toward the distal end, and in the absence

¹¹ Larger systems merely produce more possibilities, as we must select one k per segment, and (in the absence of further optimizations,) the number of possibilities scales exponentially with the number of segments, as previously discussed.

of other forces, this means that tension increases over the segment. Mathematically, this manifests itself by the exponential quantity being greater than 1. When we assume motion toward the distal end, the inverse is true and the argument to the exponential being negative makes it output a quantity less than one.

When we look at the position equations, we're adding new information to the system from the one required position measurement. Since the exponential in position equations is $(\exp(\dots) - 1)$, instead of being greater than/less than one, the argument makes the term change in sign. If our assumption was incorrect, this new position information will contradict with the information about tensions gained via assumption, and the model will produce node velocities that imply segment velocities whose signs do not match those previously assumed.

As an example, assume a single segment, bounded by two nodes, of some cable that moved in the positive direction at the current time instant. We measure a tension τ at the input, and a displacement from 0 to x between the previous and current timestep at the output. Let us incorrectly choose to assume the cable moved in the negative direction (i.e $S_1 = -1$). Then $1 > f(-1)$. From our position equations, we have

$$\begin{aligned} 0 &= u_1 - x + \tau \frac{R}{k_c \mu} (S_1)(f(-1) - 1) \\ &\rightarrow u_1 = x - \tau D_{1,-1} \end{aligned} \tag{2.26}$$

where D is a positive constant combining all terms in the tension coefficient, $\frac{RS_i}{k_c \mu} (f(-1) - 1)$. We calculate the sign of the velocity's segment as

$$\begin{aligned} \text{sign}(v_1) &= \text{sign}\left(\frac{1}{2}(u_{1,t} - u_{1,t-1}) + (u_{2,t} - u_{2,t-1})\right) \\ \text{sign}(v_1) &= \text{sign}\left(\left((x - \tau D_{1,-1}) - u_{1,t-1}\right) + (x - 0)\right) \\ \text{sign}(v_1) &= \text{sign}(2x - \tau D_{1,-1} - u_{1,t-1}) \end{aligned} \tag{2.27}$$

However, we can place a conservative upper bound on both quantities on the right. Since the term $\tau D_{1,-1}$ represents the change in length due to the frictional losses in a segment, if $\tau D_{1,-1} \geq x$ then that cable segment would be slack at the current time,

since the displacement of both nodes relative to the segment is toward the middle. This is a contradiction if $\tau \neq 0$, the presence of that measurement means the cable is not slack; and if $\tau = 0$, this term still obeys our desired bound (since it is identically zero). Fig. 2.5 visualizes this relationship. Since Node 2's position is known, when we visualize this relationship, it becomes clear that violation of the stated bound results in a segment length less than the slack length. Therefore, it is true that $\tau D_{1,-1} < x$.

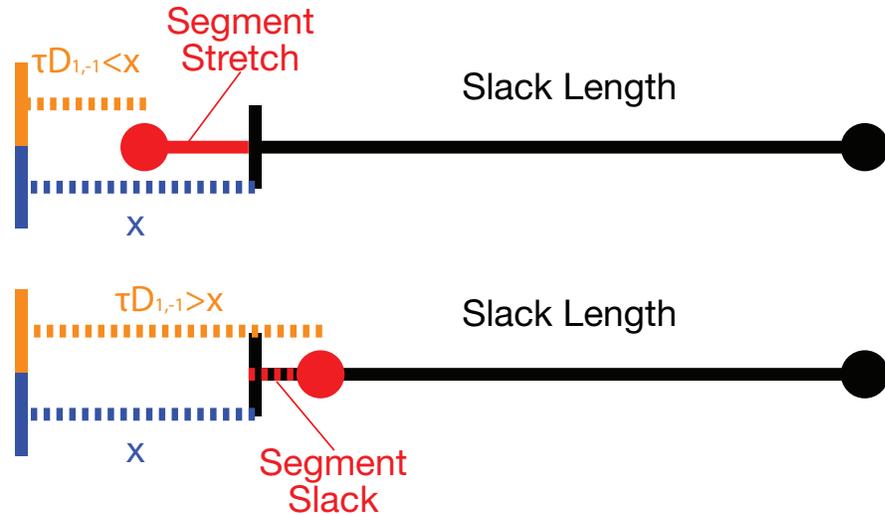


Figure 2.5: Visualization of the bounding of the quantity $\tau D_{1,-1}$. When the quantity exceeds the bound, the final length of the cable segment is less than its rest length and is therefore slack. This contradicts the fact that τ used to calculate this quantity is nonzero, and thus claims that the cable is not slack.

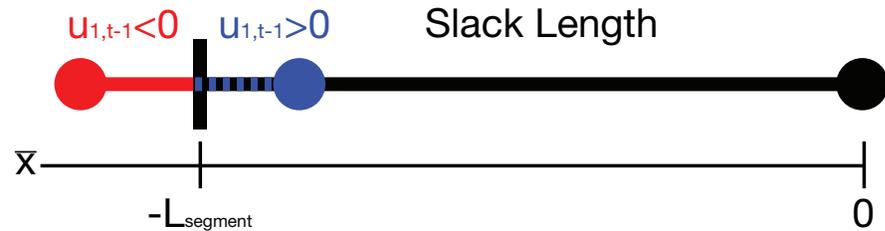


Figure 2.6: Visualization of the bounding of the quantity $u_{1,t-1}$. If this quantity were to take on any positive value, the length of the segment would be less than its rest length at the previous time, and therefore slack.

We could present a similar appeal to bound $u_{1,t-1}$, but there is a stronger argument available. Consider the state of the cable at the previous time: we have defined our coordinate system such that $u_{2,t-1} = 0$. Let's align this to a coordinate system \bar{x} of the true position of a node with respect to the conduit, rather than relative displacements, such that $\bar{x}_{2,t-1} = 0$ as well. In this coordinate system, $\bar{x}_{1,t-1} = u_{1,t-1} - L_{segment}$, where $L_{segment}$ is the cable segment's slack length. The current length of the segment is expressed as $\bar{x}_{2,t-1} - \bar{x}_{1,t-1} = (0) - (u_{1,t-1} - L_{segment})$. For the cable to be non-slack, we require that $\bar{x}_{2,t-1} - \bar{x}_{1,t-1} \geq L_{segment}$. Clearly this is only the case when $u_{1,t-1} \leq 0$. Therefore, with frames defined as we have here, the previous position of Node 1, $u_{1,t-1}$ is strictly nonpositive since the model cannot produce slack segments. Fig. 2.6 visualizes this relationship. Since choosing a different reference frame in this 1-D system results only in offsetting all quantities by a constant, we can extend this result to any frame without loss of generality.

Therefore, if we define $\gamma \equiv \tau D_{1,-1} + u_{1,t-1}$ we can state the inequality $\gamma < x$ and define an $\epsilon > 0$ such that $\gamma = x - \epsilon$ and rewrite our sign calculation as

$$\begin{aligned}
\text{sign}(v_1) &= \text{sign}(2x - \gamma) \\
\text{sign}(v_1) &= \text{sign}(2x - (x - \epsilon)) \\
\text{sign}(v_1) &= \text{sign}(x + \epsilon) \\
\text{sign}(v_1) &= 1
\end{aligned} \tag{2.28}$$

This is a contradiction on the incorrectly made assumption that $\text{sign}(v_1) = -1$.

We can trivially reject the second possible incorrect assumption of $\text{sign}(v_1) = 0$, by appealing to the changes in measured tensions and positions. Eqn. 2.15 states that when a segment is not in motion, its state variables are retained from the previous time step. However, our closed-form solution here indicates that $u_1 = u_2$ if $k_{1,j} = 0$. This is in general a contradiction of its own, unless $x = 0$ and $u_{1,t-1} = 0$ – which alternatively contradicts our construction that the true motion was in the positive direction.

Since the proof has symmetry with the opposite motion case, we can be assured that this mathematical formulation rejects incorrect assumptions.

Additionally, since there can only be one true solution of the nonlinear system, and we have equivalence of any valid assumption scheme and the original nonlinear system, we know that once we find a single valid solution, all other solutions will be invalid and we can stop without considering unchecked solutions.

2.4 Model Validation

2.4.1 Experimental Apparatus

By way of verification of our model, we designed and constructed a test-bed cable-conduit transmission to compare the model’s predictions to measured dynamics. The system connects a DC motor and a handle designed to apply torques along the flexion/extension axis of a human wrist via a pull-pull cable-conduit transmission. In this configuration, the human applies effort to accomplish motion while the remote actuator is controlled to display a desired environment for the human to interact with.

The design of the test bed is shown in Fig. 2.7. Tensioning of the system is facilitated by adjustable sliding mounts for the conduit. The system has a non-unity transmission ratio R specified by the ratio of radii between motor and handle pulleys, the value of which is approximately 5. Torque is supplied by a DC motor (Maxon motors 355679). Torque from the motor is measured via current sensing. For static tests, the load pulley can be locked to its support structure by a bolt such that the applied torque can be measured by the 6 channel force/torque sensor (ATI mini40, resolution 0.5 mNm, 4 Nm at full-scale output). Position can be measured at both ends via encoders present at both the actuator (2000 cpt) and load shaft (10000 cpt). When a human is present, an additional passive degree of freedom at the handle compensates for any misalignment between the anatomical and robotic flexion/extension axes.

Each cable consists of a 2 m wound steel wire sheath with HDPE liner with an approximate inner diameter of 1.75 mm (Lexco Cable 408187), and an inner cable with a 7x19 stranded stainless steel core coated in nylon to a final outer diameter of 0.75 mm (Sava Cable 210149). The cables are arranged in a typical pull-pull configuration used to allow for bidirectional transmission of motion and force, with cables crossing

at the output pulley to ensure sufficient engagement between cable and pulley over a large range of motion.

2.4.2 Parameter Estimation

Our implementation of the model is described by 8 free parameters. Where possible, parameters were set by the design and then measured directly: this was possible for the cable length, curvature function, and both pulley radii. To estimate k_c ,

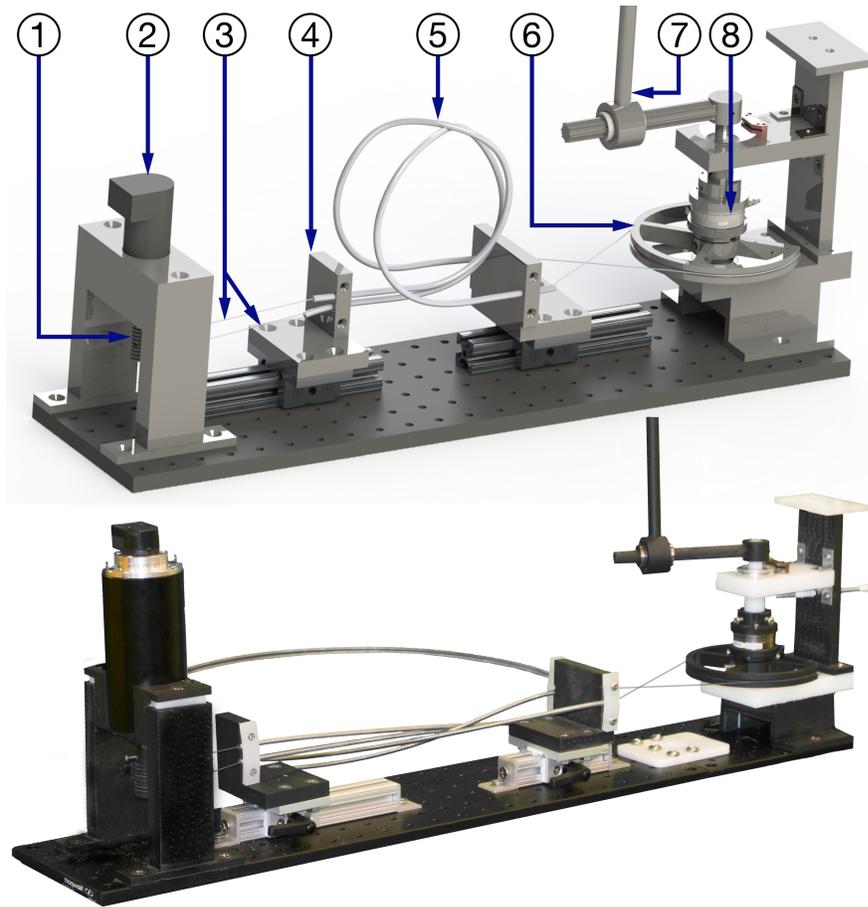


Figure 2.7: Top: Rendering of test apparatus design. The system is comprised of an input motor (2) and its pulley (1), which drives two steel inner cables (3) through their conduits (5), causing motion of the load pulley (6) and therefore the handle for human interface (7). Mobile conduit supports (4) pretension the cable, and load torque is measured by loadcell (8). Bottom: Photograph of the constructed system.

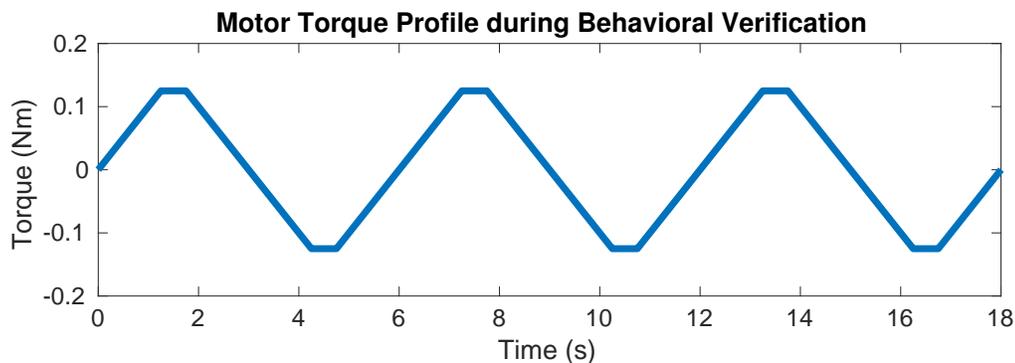


Figure 2.8: The chopped triangular torque waveform applied by the motor during the behavioral verification experiments. Torque at the output is amplified by the transmission ratio R , around a factor of 5.

the spring constant of the cable, the relationship between cable strain and tension was measured in-situ with the output pulley blocked. Pretension in the cable is prescribed by adjusting the cable mounts by displacing an attached spring a fixed distance. Both friction coefficients were tuned by hand to minimize the difference between model and measured results in an experiment independent from those reported below. The values of these parameters are listed below in Table. 2.2.

Table 2.2: System parameters used for model simulation

L	$R(x)$	r_{input}	r_{load}	K	T_0	μ_d	μ_s
2 m	0.28 m	20 mm	104 mm	$2 \frac{\text{kN}}{\text{m}}$	10 N	0.21	0.63

2.4.3 Experiment 1: Behavioral Verification

The objective of Experiment 1 was to ensure that the model responds correctly to changes in physical parameters. Here, we varied the pre-tension in the cable as well as its bend radius. To do so, the output pulley was locked in position, and the motor applied a torque which ramped at a rate of 0.1 Nm/s to a maximum of 0.125 Nm, then dwelling for 0.5 seconds before reversing direction. Several periods of this input are shown in Fig. 2.8.

In Experiment 1a, the first experiment conducted under this paradigm, the effect of variations in pretension was studied. Pretension in the cables of the physical system was set to 4 distinct levels using the spring tensioning method previously described. The motor applied the previously discussed tension profile for a period of 30 seconds.

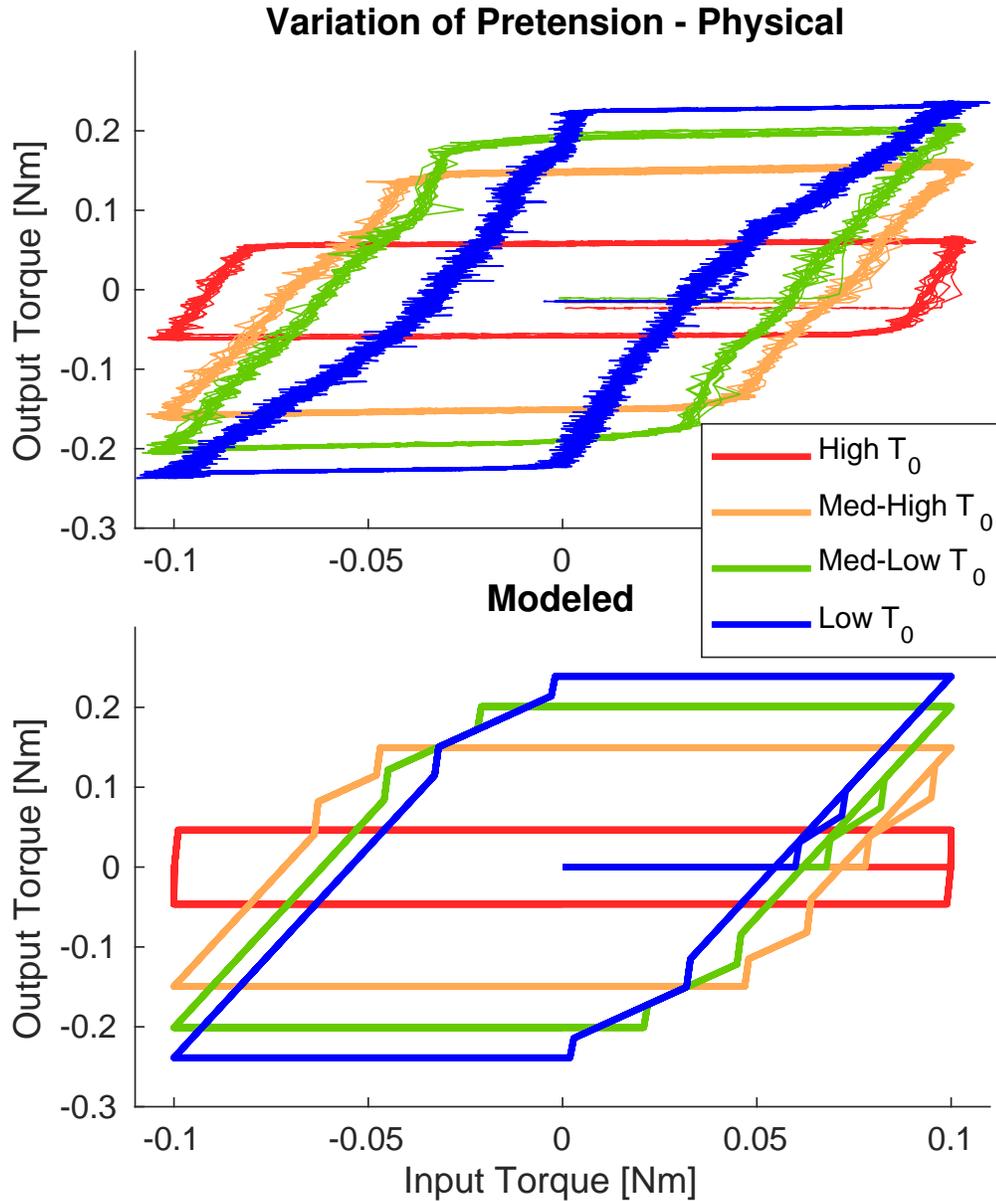


Figure 2.9: Exp. 1a: Physical system and modeled behavior when pretension in the cables is varied. The model reproduces the trends in behavior introduced by variations in pretension correctly.

The model operated in torque-position mode, and was supplied with the position of the (fixed) output handle, and the torque profile applied by the motor as inputs. Only the pretension was changed between trials, all other factors were held constant throughout. Fig. 2.9 shows the results of this experiment. In agreement with Eqn. 2.13, we observe that increasing the pretension increases the friction, decreasing efficiency. Modeled peak output torque was within 0.02 Nm of measured values.

Experiment 1b instead focused on the effects of changing the radius of curvature. To study this effect with as little complication as possible, curvature was varied by changing the number of loops the 2 meter long conduit completed between its mounting plates from 1 to 3 and keeping these loops as circular as possible. From the number of loops in the cable, and with the assumption of perfect circularity, values for $R(x)$ were estimated and given to the model to compare against the real system’s behavior. The ability of the model to accurately describe transmission dynamics as path changes is of particular importance for wearable robotics applications, where cable path would be time variant as a function of joint angles proximal to the joint being actuated. In this experiment, pretension was set to a constant level after the cable path was altered, to ensure consistency of results. As shown in Fig. 2.10, increasing the curvature of the cable results in larger backlash and a decrease in transmission efficiency, as expected, both in the physical system and in simulation. In this experiment, modeled peak torque remained within 0.03 Nm of the true value, and the width of the backlash was within 0.02 Nm of measured.

2.4.4 Experiment 2: Position-Force Mode

Experiment 2 validated our model’s capacity to capture dynamic interaction with an active environment. For this experiment, the motor was controlled to apply a torque τ , expressed in Nm, in response to angular displacement θ , in radians, of the output handle. This configuration is similar to virtual tunnels used in haptics applications, however by controlling for input torque instead of output, the rendered

system will display backlash as a result of transmission friction. The tunnel is described

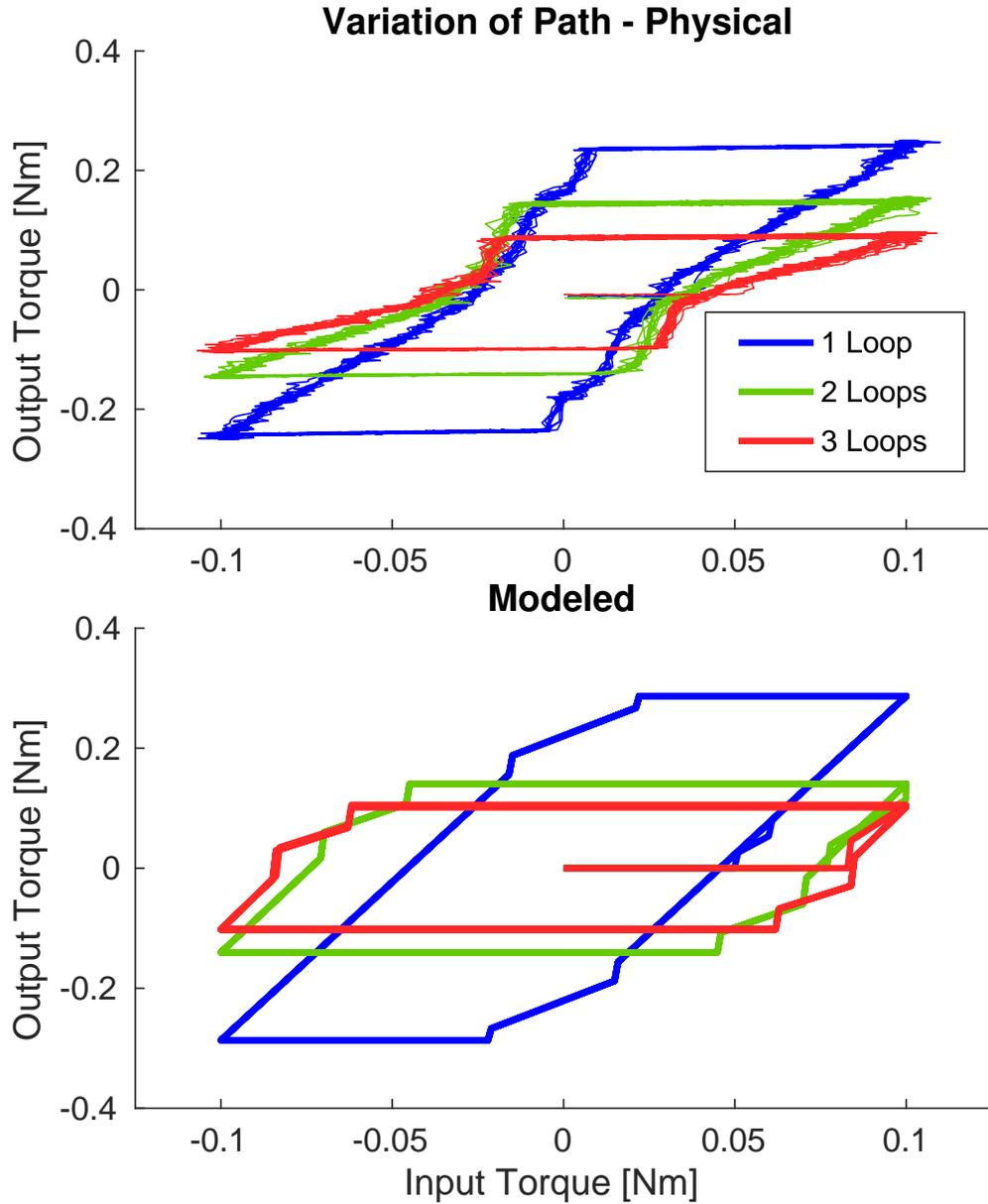


Figure 2.10: Exp. 1b: Physical system and modeled behavior when the radius of curvature of the cable is altered by changing the number of loops the cable completes.

by the equation

$$\tau(\theta) = \begin{cases} \min(-\frac{1}{2}(\theta + 0.05), 0.2) & \theta < -0.05 \\ 0 & |\theta| \leq 0.05 \\ \max(-\frac{1}{2}(\theta - 0.05), -0.2) & \theta > 0.05 \end{cases} \quad (2.29)$$

and is shown in green in Fig. 2.11.

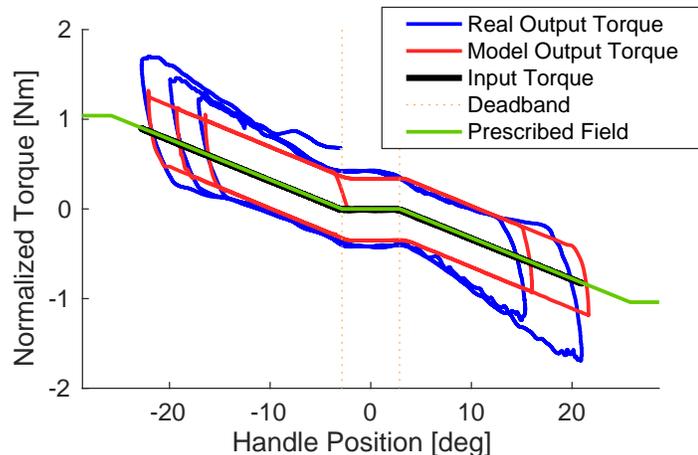


Figure 2.11: Experiment 2: The virtual tunnel as designed and as rendered by the cable-conduit transmission. Numerical accuracy of the simulation is very high through most of the domain, with systematic errors in some regions as a result of an effect not included in the Coulomb model.

During the experiment, a participant was instructed to move the handle in a periodic fashion. They were to sweep to one side to an angle of at least 20 degrees, hold momentarily, then sweep to the opposite side with the same magnitude angle. The model was provided with the trajectory imposed by the human and the corresponding torque requested from the motor to repeat the experiment in simulation. The cable was set to a fixed pretension with one loop in the conduit, with all other parameters unchanged. Results, shown both as timeseries and in input-output domains in Fig. 2.12, demonstrate fair tracking of the true output by the model, with some deviations present from un-modeled dynamics. The systematic nature of this is made clearer by observing the results in the position-torque domain (Fig. 2.11) that describes the tunnel

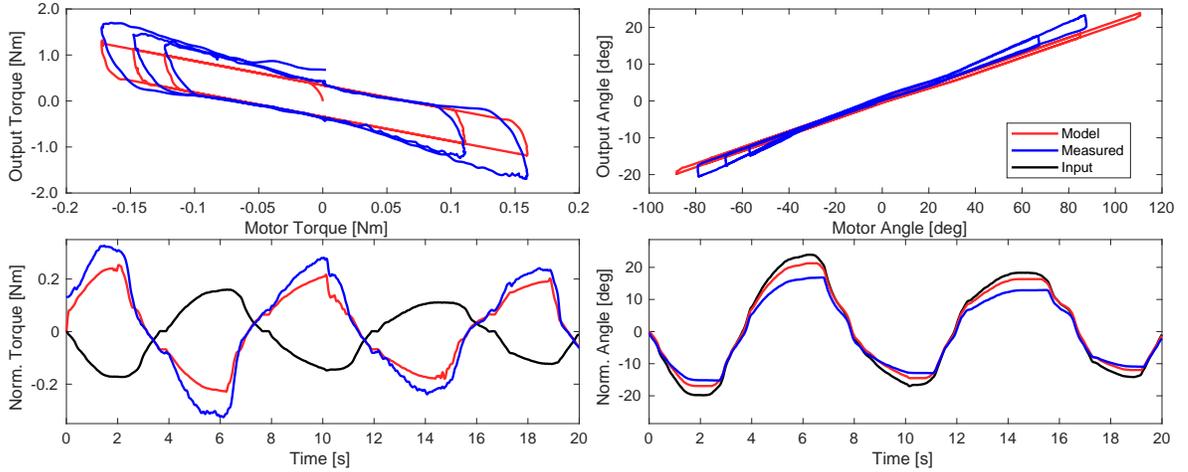


Figure 2.12: Experiment 2: Viewing torques and positions in the time domain shows that the modeled behavior is temporally in phase with that of the physical system. Modeled angle is correct to within 3 degrees throughout the experiment.

itself. Numerical accuracy is high, with the root-mean-squared error being $0.27Nm$, largely due to a divergence in the slope for two regions. This suggests that the simple Coulomb model elected for model development will need to be replaced with a more accurate model for controller use.

A piece of evidence agreeing that Coulomb friction is to blame is to observe model residuals binned by cable velocity (As here approximated by handle velocity). This relationship is shown in Fig. 2.13, where there is visibly some dependence between model residuals and handle velocity. Taking a first-order, linear approximation of this relationship, we can include a viscous friction term that decreases the bias in residuals from the previously reported $0.27Nm$ by about $1/3$ to $0.19Nm$.

2.4.5 Experiment 3: Position-Position Mode

For the third experiment, we wanted to validate the capability of our model to operate when supplied with two position measurements and no torques. To do so, we study the case where the motor is controlled to track the position of the output. For a high-friction transmission, intuition suggests that when input and output are

subjected to the same trajectory in this way, each source is responsible for moving half of the cable, and as such must compensate for 50% of the system’s friction. Our experiment involved a human participant imposing an arbitrary periodic trajectory to the load pulley, and a sufficiently stiff PI position controller tracked the trajectory $\theta_{des} = R\theta_{load}$. In practice, it was expected that the distribution of tension would not be a 50/50 split due to non-idealities and we would observe some asymmetry in measured torques.

Unlike other experiments, we ran the model in two scenarios—first, the ideal condition, where the model was given a perfect position controller that specified explicitly that $\theta_{motor} = R\theta_{load}$, to represent the ideal case that we expect to match our intuitive result. Second, the model was provided the true position achieved by the motor, such that it would attempt to recreate non-idealities seen in the physical system and its simple position controller. Figure 2.14 shows firstly, that when the ideal control law is applied, the model predicts the intuitive load-sharing result, with torque profiles at input and output identical when normalized for transmission ratio.

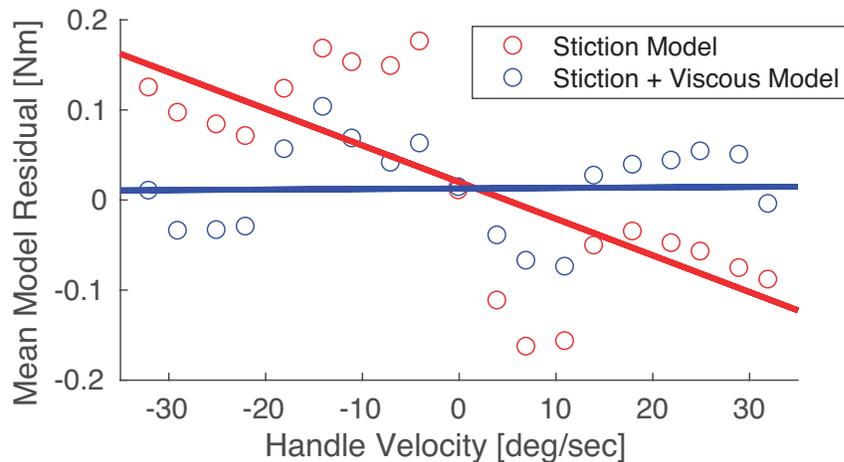


Figure 2.13: Experiment 2: Grouping residuals with respect to handle velocity shows the need for viscous friction to be included in the basic friction model. A first-order approximation of this viscous term reduces the bias in residuals. The remaining residuals resemble friction due to the Stribeck effect.

2.4.6 Validation Results

Overall, the modeled behaviors are successful at representing those observed in the physical system. The model reacts correctly to changes in physical parameters, and correctly simulates backlash in and re-engagement of the transmission. Numerical accuracy is fair, but is demonstrably improvable via incorporation of a more sophisticated core frictional model than the chosen Coulomb friction. We have sufficient evidence for the capability of this model to consider its feasibility for use for state estimation in a controller.

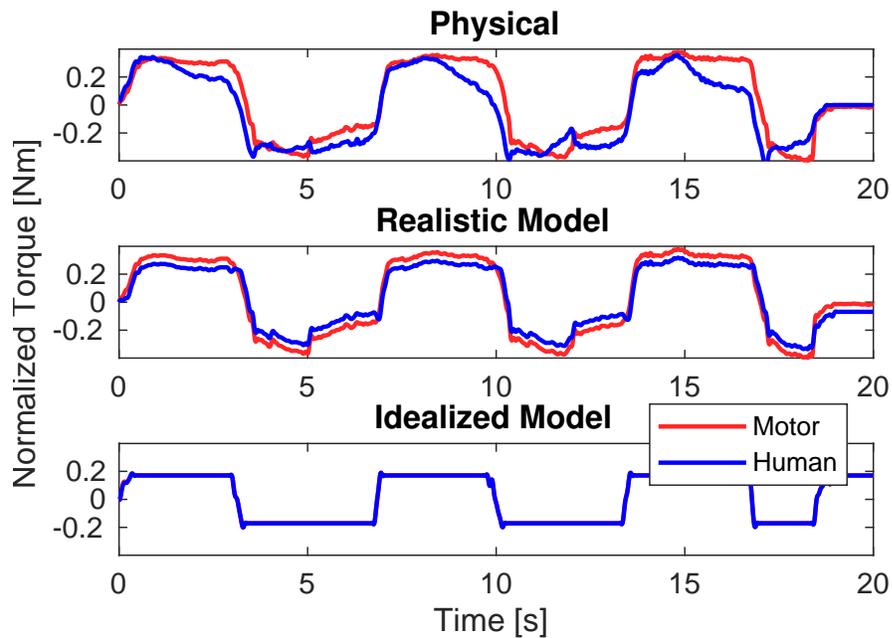


Figure 2.14: Exp. 3: A comparison of physical, realistically modeled, and ideally modeled position-follower systems. The idealized model correctly splits effort between both distal and proximal inputs evenly. Using measured position controller response as one input for dynamic model (Realistic Model) reproduces some of the asymmetry seen in the experiment.

Chapter 3

MODEL-BASED STATE ESTIMATION IN REAL TIME

The model alone has intrinsic merit for its ability to simulate and reproduce behaviors evident in the physical system, but the best potential application for the model is for use in a feedback controller. However, out-of-the-box, the model performs poorly in the absence of at least a force/position measurement at the input and output. If one is missing altogether, the system is underdefined during mass-motion (And only one mobile segment can be studied during partial motion, not both) making solving for arbitrary states impossible. If a measurement is noisy, the model is prone to chattering at the onset of motion, as shown by a toy model in Fig. 3.1. In practice, the consequences of noisy measurements are worse, since each segment of the cable is

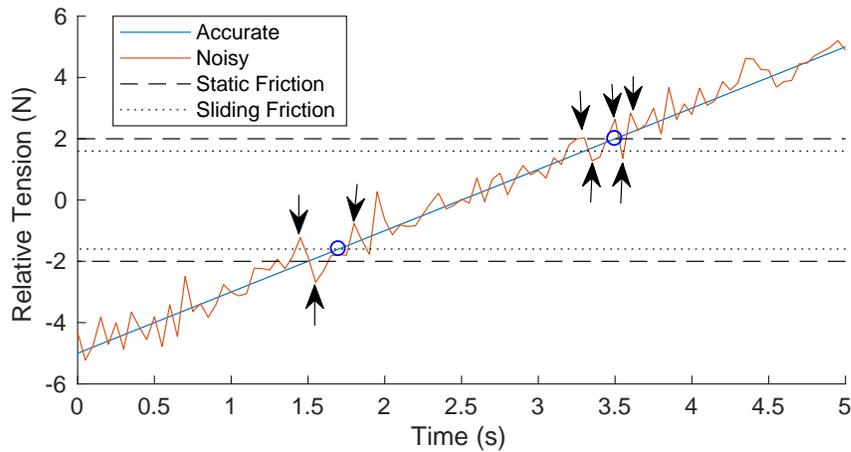


Figure 3.1: A toy model where motion of the cable is started or stopped based off a sensor reading. While the true system (blue) stops and starts only once (blue circles), a moderately noisy sensor (orange) causes the toy model to change motion 8 times (black arrows).

free to stick and slip as an individual unit with its own frictional forces. While a low-pass filter can mitigate this, it will not remove the effect entirely without introducing substantial delay.

State measurements are easy to obtain with high precision at the input; a motor typically can sense its torque output and position. Conversely, they may not be available at the output in a wearable system due to space and weight constraints. For these practical applications, we need to look at cases where the physical system employs sub-optimal sensing strategies, and integrate these measurements with our model to provide robust state estimates usable by a controller.

3.1 Kalman Filtering

A common means to use a dynamic model as a state observer for a feedback controller (e.g. full-state feedback) is to wrap it in a Kalman filter. Kalman filtering is a technique to integrate model predictions with sensor measurements to generate estimates of states alongside model-filtered values for the available sensors.

A common use case for the Kalman filter is to correct for model inaccuracies that would cause outputs to drift away from the true state over time. The classic example for this scenario is dead reckoning of an object's position based on motor commands being integrated with a relatively noisy GPS sensor. In this case, integration of the sensor measurement can be performed using a model-based estimate provided by a Kalman filter, and is able to prevent nonphysical drift in the position estimate inherent to dead reckoning methods. The Kalman filter avoids introducing large amounts of noise inherent to the GPS sensor and discontinuities that might occur from other methods of preventing drift, like a bump switch at a known position that instantly resets the position of the object when activated.

In a typical application, filtering has two stages: prediction, and update. The prediction stage uses the previously estimated state (or an initial guess) and the known input signal to form an a-priori guess of what the next state should be, and based on past sensor performance, a new estimate for noise in the model. After measurement,

the update stage can take place. Here, the sensor readings are integrated, and a new estimate for the state is provided considering both model and sensor readings. The process is repeated at the next iteration with the same sequence of prediction and update.

To utilize the Kalman filter, it is necessary to put dynamic equations into a discrete-time matrix form. If the system can be represented in time-invariant state-space form,

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (3.1)$$

then the *state-transition matrix* for discrete timestep Δt can be given by $F = \exp(A\Delta t)$ such that the zero-input response of the system is given by $x(t+\Delta t) = Fx(t)$. A similar matrix B' can be calculated for the zero-state response, given by the integral

$$\int_0^{\Delta t} e^{-A\tau} Bu(\tau) d\tau \quad (3.2)$$

The prediction stage at iteration k starts by generating an *a-priori* estimate of the state, by using F , the state-transition matrix, and our last known state $\hat{x}_{k-1|k-1}$, to form the predicted zero-input response. By adding the predicted zero-state response from the commanded input u , we find

$$\hat{x}_{k|k-1} = F\hat{x}_{k-1|k-1} + B'u \quad (3.3)$$

An a-priori estimate of the covariance matrix $P_{(k|k-1)}$, which is used as an automatically updating weighting factor between all available sensors and the models predictions. At this stage, some uncertainty Q is also factored into the covariance to prevent the filter from 'locking in' on a single source of information. The update law is given by

$$P_{k|k-1} = FP_{k-1|k-1}F^{-1} + Q \quad (3.4)$$

After making the model-based prediction, a sensor measurement z_k is taken. However, our prediction also provides an estimate of what the sensors *should* have measured, through the observation matrix H . At this update step, we also maintain

a measurement covariance matrix S with its respective measurement noise R . We therefore calculate the difference between expected and actual measurement, \tilde{y} ,

$$\tilde{y}_k = z_k - H\hat{x}_{k|k-1} \quad (3.5)$$

and update the measurement covariance,

$$S_k = HP_{k|k-1}H^T + R \quad (3.6)$$

The key step is then to calculate the *Kalman gain*, K , which is used to incorporate the error between model and measurement to drive the state estimate, given by

$$K_k = P_{k|k-1}H^TS_k \quad (3.7)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\tilde{y}_k \quad (3.8)$$

This updated state estimate can be used in real-time by the controller at this point, but there is still one required step: updating the model covariance P ,

$$P_{k|k} = (I - K_kH_k)P_{k|k-1}(I - K_kH_k)^T + K_kRK_k^T \quad (3.9)$$

and if desired, calculating a a-posteriori measurement residual, which can be used to monitor accuracy of measurements,

$$\tilde{y}_{k|k} = z_k - H\hat{x}_{k|k} \quad (3.10)$$

In its basic formulation, the Kalman filter only functions on linear systems. A straightforward modification to the system, the Extended Kalman filter, allows for the use of nonlinear systems by linearizing the nonlinear state transition function about the current state estimate by taking the Jacobian, and using the resulting approximated state transition matrix for update calculations, although the full equations can be used during the prediction step since it doesn't rely on multiplication with covariance matrices. The same process can be applied to the observation equations as needed. If

the dynamics are well conditioned with continuous first derivative (and preferably with low magnitudes in higher order derivatives), and the update frequency of the filter is fast enough that changes to the operating point iteration to iteration are small, this provides a sufficient approximation for the filter to remain effective.

Our first efforts at using this model as an observer were based on the premise of Kalman filtering. Since the system is nonlinear, a traditional linear Kalman filtering approach is unsuitable. However, our solution method *produces* an equivalent linear system, which we sought to use for filtering. While linearizing a system for use in a KF is a standard approach (called the Extended Kalman Filter or EKF, which will be discussed in the next section), the linearization method used by our solver is dissimilar to the typical method, so we call this method the E*KF. In this method, the full model is run to generate the a-priori state estimate, and the correct linear matrices are extracted for the covariance prediction and update steps.

Since our solver generates systems of the form

$$\mathbf{A}\vec{x} = \vec{B} \tag{3.11}$$

where

$$\vec{B} = \begin{bmatrix} \vec{0} \\ \text{Measurement 1} \\ \text{Measurement 2} \end{bmatrix} \tag{3.12}$$

we must adjust to the form expected by the filter, $x_{[t]} = Fx_{[t-1]} + Bu_{[t-1]}$. Since we previously used our linear matrix to acquire a solution, we know it is of full rank, and therefore invertible by the invertible matrix theorem, so we can write

$$\begin{aligned} \mathbf{A}\vec{x} &= \vec{B} \\ \mathbf{A}^{-1}\mathbf{A}\vec{x} &= \mathbf{A}^{-1}\vec{B} \\ \vec{x} &= \mathbf{A}^{-1}\vec{B} \end{aligned} \tag{3.13}$$

However, an issue is immediately apparent. $\mathbf{A}^{-1}\vec{B}$ represents the second half of the filter update equation for input, and no term appears in the previous state. If we force it into that format, we find that

$$x_{[t]} = [\mathbf{0}]x_{[t-1]} + \mathbf{A}^{-1}\vec{B}_{[t-1]} \quad (3.14)$$

Since the state transition matrix \mathbf{F} is a matrix of zeros, most of the Kalman filter equations break down. Clearly, covariance updates are impossible, since not only is the previous covariance discarded by front multiplication by a zero matrix, \mathbf{F} is not invertible. Upon careful consideration, it makes sense that this approach is non-functional, since the model equations become purely algebraic under our linearization. Therefore, the next logical thing to try is to undo the step that cost us our state dependence, and try a traditional EKF.

3.1.1 Conventional Extended Kalman Filter

The standard extended Kalman filter relies on computing the Jacobian of the system equations about the current operating point to perform the update steps after using the nonlinear system to perform the prediction. To consider the feasibility of this approach, let us consider a 3 node, 2 segment cable. We can present a simplified version of the system of equations, rolling together all constants into c_i and d_i , to give us

$$\begin{aligned} T_1 &= \text{measured} = K_1 \\ T_2 &= T_1 * \exp(c_1 \text{sign}(v_1)) \\ T_3 &= T_2 * \exp(c_2 \text{sign}(v_2)) \\ u_1 &= \text{measured} = K_2 \\ u_2 &= u_1 + d_1 * (\exp(c_1 \text{sign}(v_1)) - 1) \\ u_3 &= u_2 + d_2 * (\exp(c_2 \text{sign}(v_2)) - 1) \end{aligned} \quad (3.15)$$

We also have to define the velocities. Let us assume that we have unit timestep, then ideally we use our previous choice of

$$v_i = \frac{1}{2}((u_i^t - u_i^{t-1}) + (u_{i+1}^t - u_{i+1}^{t-1})) \quad (3.16)$$

but for estimation, we *could* relax this to the incorrect¹

$$v_i = u_i^t - u_i^{t-1} \quad (3.17)$$

First, let's use the correct velocity formula. If we label the individual equations from 3.15 as e1 through e6 in order, we can attempt to solve symbolically in MATLAB for the state variables using the Symbolic toolbox.

```
1 eqnset = solve([e1; e2; e3; e4; e5; e6], [t1 t2 t3 u1 u2 u3]);
```

Unfortunately, MATLAB correctly reports that this system has no closed-form solution. Let us try instead the incorrect velocity definition given in 3.17, and again seek out a symbolic solution. This time, we find closed-form solutions of the form

$$\begin{aligned} T_2 &= K_1 \cdot \exp(c_1 * \text{sign}(K_2 - u_1^{t-1})) \\ u_2 &= K_2 + d_1 (\exp(c_1 \text{sign}(K_2 - u_1^{t-1})) - 1) \end{aligned} \quad (3.18)$$

Neglecting higher-indexed equations for the moment, we know that for use in an EKF, we require the Jacobian of these equations with respect to the state variables. We recognize that the only nonzero term for the Jacobian of the two provided equations is in u_1 , and the corresponding partial derivatives are given by

$$\begin{aligned} \frac{\partial T_1^t}{\partial u_1^{t-1}} &= -2c_1 \cdot K_1 \cdot \delta(K_1 - u_1^{t-1}) \exp(c_1 \text{sign}(K_2 - u_1^{t-1})) \\ \frac{\partial u_1^t}{\partial u_1^{t-1}} &= -2c_1 d_1 \delta(K_1 - u_1^{t-1}) \exp(c_1 \text{sign}(K_2 - u_1^{t-1})) \end{aligned} \quad (3.19)$$

where δ is the Dirac delta function,

$$\delta(x) = \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases} \quad (3.20)$$

¹ Recall from earlier, taking a node velocity to be the velocity of its adjacent segment can cause model nonpassivity. However, we're not worried if we make small errors in estimation, and can consider this as a possibility if the errors it causes would prove to be small.

The argument to this Delta function is the velocity of the first node. Thus, this Jacobian is only nonzero when node 1 is *stationary*, a condition where the calculation is uninteresting since stationary nodes cannot transmit motion or force. This pattern continues through all elements of the Jacobian in all state variables, meaning that the linearized \mathbf{F} matrix is the same as found before, identically $[\mathbf{0}]$.

It is apparent that the delta function arises from the sign function, since

$$\frac{d}{dx}\text{sign}(x) = \begin{cases} 0 & x \neq 0 \\ 2\delta(0) & x = 0 \end{cases} \quad (3.21)$$

Let us take an approach common in friction estimation, substituting $\text{sign}(x)$ by a continuous approximation, $\tanh(Kx)$ with K chosen appropriately such that approximation error is small over the domain of possible x values. We could repeat the previous calculation, but would discover a similar result, $\mathbf{F} \approx [\mathbf{0}]$ with variations from zero strictly a consequence of the approximation error between sign and \tanh . Instead, let's try to use this approximation to allow use of the proper velocity estimate (Eqn. 3.16). While this does not provide a closed-form solution, we can take a Taylor series approximation².

We ask MATLAB to use the new equations (as shown below) with the correct velocity expression, a \tanh approximation for the sign function, and taking a fifth order Taylor series,

```
1 e1 = t2 == t1 * taylor(exp(c1*tanh(k*v1)),u2,'Order',5);
```

then carry the solution through to a Jacobian matrix. Due to the high order Taylor expansion taken in the name of numerical accuracy, the number of terms in the output is immense and will not be transcribed here. However, a plot of one element (Fig. 3.2) shows that we find no meaningful relationship, and (with the exception places where double-precision isn't enough to avoid divide-by-zero problems) the derivative

² Which was not possible previously, the Taylor series expansion of the sign function doesn't exist.

is identically zero except in a region around where the velocity would be, again, zero. In practice, the 'peak' would be about 1000 times narrower, here k has been set such that 1 m/s is the cutoff for 'slow' velocities.

To confirm that this low-velocity effect is solely an effect of the tanh approximation, we can compare similar plots for the poor velocity approximation, as in Fig. 3.3. From these plots, we observe that the use of sign instead of tanh is the only source of the apparent 'effect.' At this stage, we must conclude that a Kalman filtering approach is unsuitable for this model, and seek other methods to utilize the model in a controller.

3.2 Redundant Sensor Fusion

Recently, the viability of utilizing a carbon nanotube based coating as an axial strain sensor has been demonstrated[60]. In theory, several coated segments could be located along the cable to give extra measurements of cable tension. However, the noise properties and error inherent to this type of sensor mean that high-level processing is needed to estimate cable tension. Additionally, the difference in location of the sensors means they cannot be directly compared, but first need to be subjected to a transformation so that they present the same information. As a result, difficult signal analysis is needed to parse this information into a usable format.

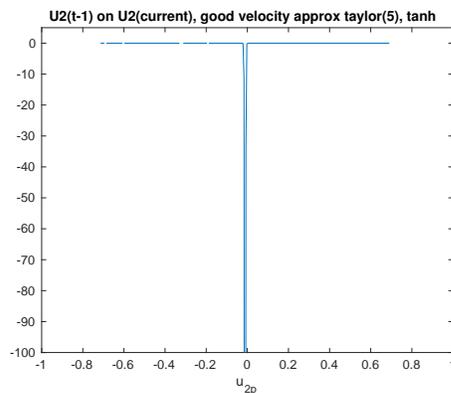


Figure 3.2: Jacobian linearization of the tanh approximation of sign to the first 5 terms of a Taylor series. There is no dependence on the previous state at any nonzero velocity, making this unsuitable for Kalman filtering.

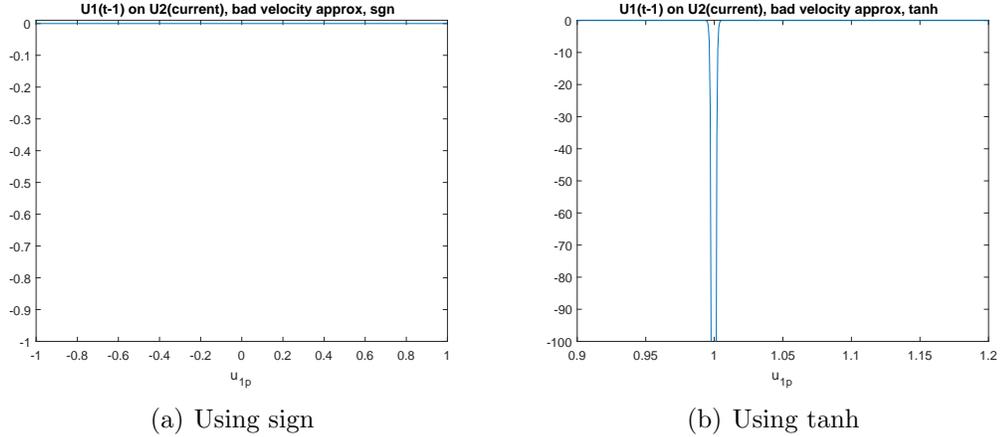


Figure 3.3: When trying to linearize the system using single-ended velocity approximations, there still remains no dependence of previous states on the new state at any nonzero velocity. Artifacts in (b) stem from approximating the $sign(x)$ function by $\tanh(x)$, and do not represent the system dynamics.

3.2.1 Direction of Motion Known

Initially, let us assume that we have one ‘reliable’ tension sensor that can correctly assign the directions of motion in the cable. Once the model has selected the correct assumption for values of the sign functions, our algebraic equations can be solved given *any* measurement of position and tension. Assuming also that we have a high-resolution encoder available in the system, we can obtain a state estimate \hat{x}_i for each measurement of cable tension available. This property makes comparison of the multiple measurements relatively straightforward. When we assume one noisy sensor at each of 11 nodes in a representative cable, Fig. 3.4 demonstrates the values of \hat{x}_i obtained at a representative time instant through this method, where we have assumed each tension sensor is contaminated by white gaussian noise. If we assume that the output tension is the metric of interest, Fig. 3.5 describes the error in this measurement as a function of the availability and quality of sensors.

To fuse the measurements together in this case, the minimum-variance unbiased estimator is exactly the sample mean. If we take the mean of each \hat{x} at every time

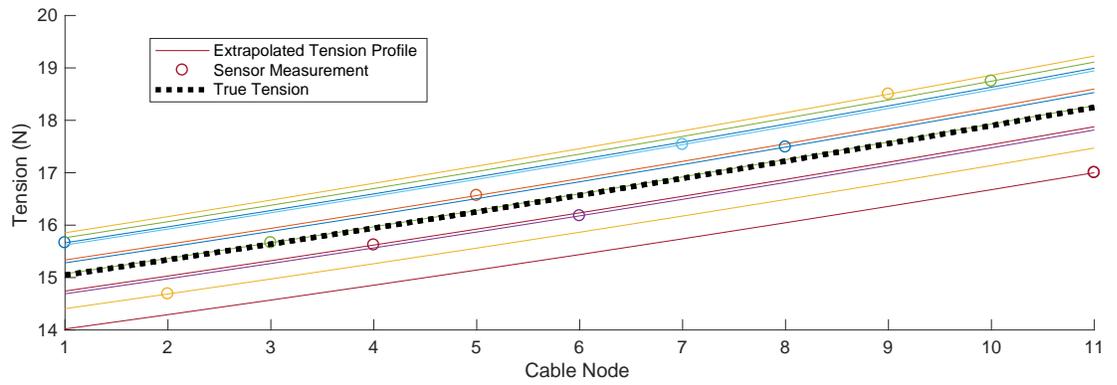


Figure 3.4: Assuming knowledge of motion direction, the model extracts tensions across the entire cable for each sensor measurement. Sensor noise causes discrepancy between each prediction and the true state.

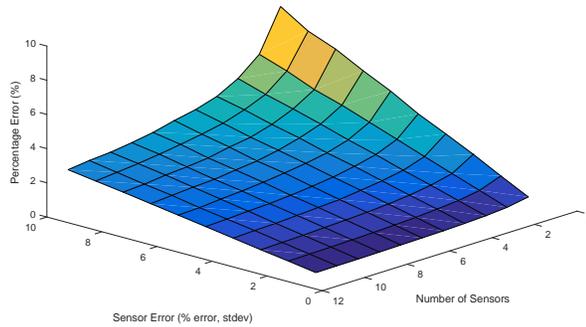


Figure 3.5: Absolute percentage error given a cable with variable number of available sensors, at different levels of white Gaussian sensor noise. Displayed values are the mean of 10,000 simulations each.

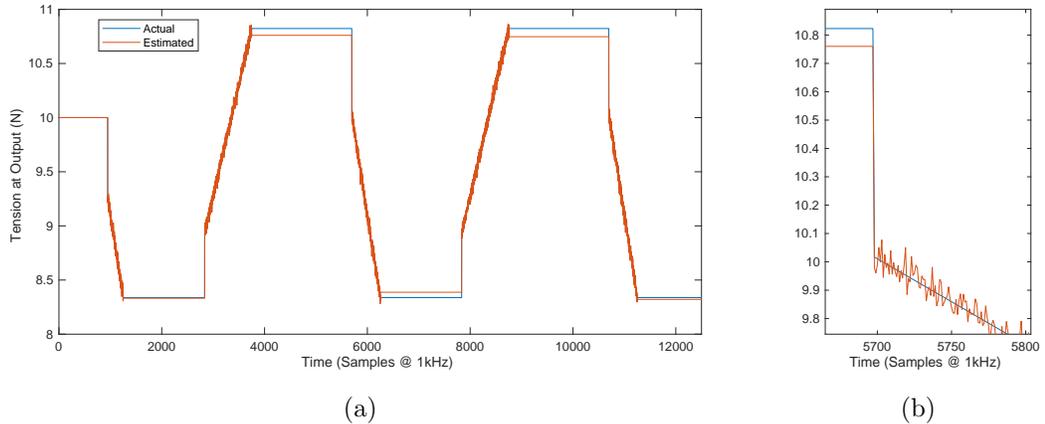


Figure 3.6: Predicted output by noisy sensors being transformed and averaged by the model. Predictions are made over the course of an experiment (a) in which the tension is ramped up and down, and a view zoomed for detail(b). The estimated signal resembles white gaussian noise about the true value with no time-domain effects. When motion stops, error is equal to that of the last moment when the cable was in motion, since an immobile cable cannot transmit force.

instant in a simulation, we observe that we obtain a very accurate result, even when neglecting the tensions provided by the accurate sensor³. An example timeseries is shown in Fig. 3.6.

3.2.2 Direction of Motion Unknown

In practice, we cannot assume that we have *a-priori* knowledge of the direction of motion of each cable segment. In this case, each sensor must be allowed to create its estimate of motion based on the previous time-step’s estimate. We can then seek a consensus between sensors to choose the true motion direction(s) and last-moving nodes. An interesting consequence of this new procedure is that sensors are being used to extrapolate values beyond their own position. Since the model allows bidirectional motion propagation, these extrapolated values are subject to errors.

³ But not its determination of the sign-assumptions

Let's first consider the case of mass motion. As an example, assume we have a 10-segment cable for which tension increases monotonically from the near end (Node 1) to the far end (Node 11) during mass motion. At time t , our measurement from the tension sensor at Node 6 measures a value significantly lower than the true tension at Node 6 due to sensor noise. It is possible (if not likely) that the error in this measurement would make the model conclude that *due to the friction in the cable, tension cannot increase monotonically, as friction would overcome the tension difference in the cable at Node k* . In a case where node k is in the interpolated region ($k < 6$), this signals the model that it needs to decrease the last-moving node and move away from mass motion. However, when this occurs in the extrapolated region, $k > 6$, the model can conclude that *in order to remain in mass motion, the region distal to node k must be moving in the opposite direction and experiencing a decrease in tension between segments*. An example of this phenomenon is shown in Fig. 3.7. Note that the errors always occur in the region *after* the sensor, as predicted.

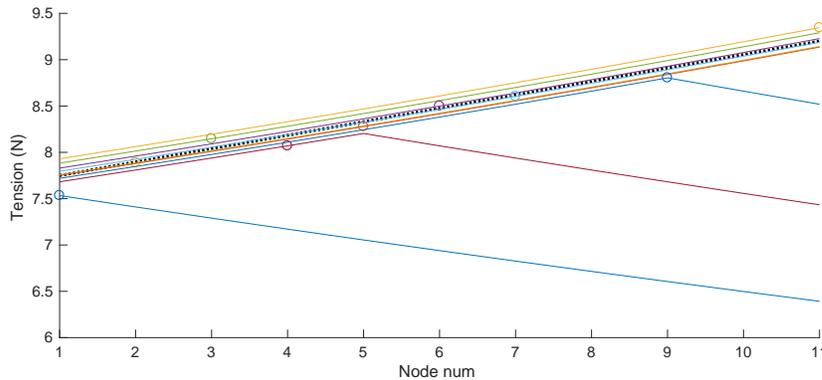


Figure 3.7: Predictions about cable tension made by noisy tension sensors. Sensors located at a given node (colored circles) that report a lower tension than the true tension (black dots) at a given time instant form a prediction (colored lines) are subject to an extrapolation error, in which the direction of motion is incorrect for some nodes beyond the location of the sensor.

This trend persists across all shapes for the tension profile. While altering the slope decreases the probability that a large extrapolation error will occur (By increasing

the tension differential between segments, and thus placing the system further away from the threshold where motion would cease), the possibility cannot be precluded entirely. This leads to position-dependent accuracy of the sensor. While it follows trivially that the sensor at node n is the best predictor of the tension at node n , it is observed that the RMS prediction error for nodes in the extrapolated region $k > n$ is significantly worse than for those in the interpolated region. Fig. 3.8 shows this effect across 4 cable configurations.

In order to proceed in the presence of these erroneous extrapolations, some decision algorithm (e.g. most common direction) is needed to determine which set of positions and directions should be used passed forward to the next iteration. However, using *only* the agreeing systems would introduce a bias to the estimation process – the poor extrapolations are systematically from measurements that are significantly above or below (depending on the direction of motion) the true value, such that directly excluding them would bias the measurement to be less or greater than the true value, respectively. To unbiased the estimation, once consensus is reached, the linear solution matrix A is generated and is held constant across all sensors, and each sensor calculates the output tension based on this same matrix.

In partial motion, there's the additional problem of deciding what nodes are in motion and which are not. Our model holds invariant that nodes tension can only change if they are in motion. Since each sensor is noisy and its value will change iteration to iteration, it will **at least** believe itself to be in motion. However, during partial motion, we only need a single sensor in each mobile segment to solve the system since the first stationary node provides a boundary condition. On the segment proximal to the motor, the system will always have a high-quality encoder and can trust its measurements to accurately describe the propagation of motion through that half of the cable. Distally, however, we cannot assume we have a sensor with sufficient noise characteristics to solve for motion propagation alone. Instead, we must trust a consensus of low-quality sensors. For this analysis, each sensor in the segments known to be mobile from the previous time instant is considered to solve for motion, and a

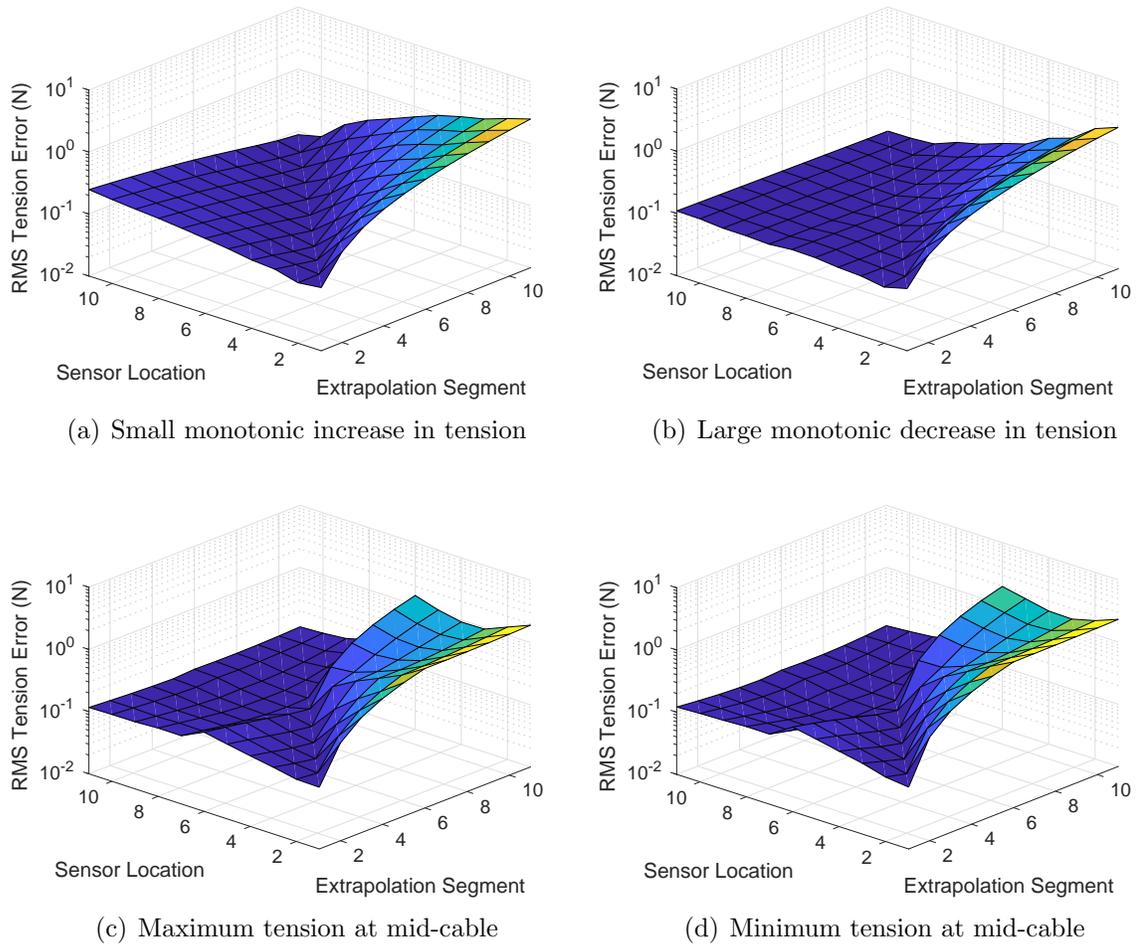


Figure 3.8: Under each possible movement condition, a cable was simulated with a sensor at each node. These plots report the RMS error in the estimate formed by a sensor at a given location at each other location in the cable. Extrapolation error, where the sensor is located proximally to the node of interest, is worse due to errors in estimating the direction of motion beyond the last available measurement.

consensus function (such as the median last-moving-node⁴ determines how many nodes should be considered to be in motion at the current time.

⁴ We elect the median instead of the mean—the mean would over-weight sensors who determine that the cable should stop due to noise. Imagine a set of last-moving-nodes [6, 7, 9, 7, 0, 0, 0] – The mean LMN is 4, which no sensor believes to be the truth, but the median is 6, and is ensured to be believed by at least one sensor.

This analysis can begin to provide weights with which to estimate the tension at every node as a function of all available sensors. However, these relative errors alone assume that each sensor is of equal quality. In practice, it is likely that each of these sensors will have different numerical accuracy and worse still, time-varying noise properties. In preliminary testing, carbon nanotube based sensors applied to the cable in our test transmission had widely varying noise properties which changed over time. For this reason, a practical system likely requires a means to estimate the noise properties, and thus a second weighting term, for each sensor on-line.

One method that has been shown to perform this function is the least-squares method for calculation of autocovariance matrices[61]. This method provides unbiased estimates which necessarily converge, and since the solution space is convex, solutions can be found via a computationally efficient method. By combining this autocovariance matrix with an error covariance matrix derived from the extrapolation errors in Fig. 3.8, it should be possible to form sufficiently accurate state estimates for use in a real-time controller.

3.2.3 Online Parameter Estimation

In order to utilize this model, it must be supplied with values or functions that describe R , the radius of curvature of the cable, μ , the frictional coefficient(s) of the cable, and K , the longitudinal stiffness of the cable. In all testing done so far, these quantities were measured or estimated in an offline fashion to obtain the displayed results. However, were it possible to update these parameters with the system online, numerical accuracy of the model could be improved, and it might be possible for the update law to capture systematic differences between the modeled and real dynamics.

Our lab is working on implementing a sensing scheme to measure the bend radius of the conduit during motion based on fiber-Bragg gratings spaced along the outside of the conduit. These sensors measure strain optically since deforming the grating has an effect on the frequency of light waves that are transmitted and reflected by the grating. An optical interrogator can measure these reflected frequencies, and that can

be used to calculate the strain in the fiber at the location of the grating. With several of these in parallel at the same longitudinal cable coordinate but at different angular coordinates, this strain measurement on the conduit’s surface can be transformed into a bend radius. Placing several sets of gratings along the cable’s length gives additional measurements of bend radius, and fitting to a function with enforced continuity of derivatives allows for estimation of the curvature of the entire cable.

Alternatively, if most of the cable’s path is known (e.g. it’s routed through a fixed path in an exosuit, and only has variable curvature at the joints), it may be possible to use other available measurements (e.g. a goniometer reading) to estimate the entire cable’s path, and thus curvature.

This leaves two parameters, the stiffness K_c and the frictional coefficient(s), μ . Past efforts at online compensation for friction have involved utilizing well known rigid-body system dynamics into an extended Kalman based friction estimator (EKBF) where the friction experienced at a point is treated as a random constant under a first-order Gauss-Markov formulation. The filter then estimates these constants based off the error between measured states, and those that would be expected from the frictionless rigid body dynamics [62]. This approach isn’t feasible for our model, as we have no dynamics whatsoever not caused by friction itself. Treating friction as a random variable is akin to ignoring the model entirely. A different approach was used by Srang and Yamakita [63] where an *unscented* Kalman filter was used with the model states defined as the constants within their frictional model itself, a combined Stribeck and viscous friction model. The Unscented Kalman filter, described in [64], replaces partial derivatives with the unscented transform which allows for the integration of model parameters as states which can be tuned by the filter itself to drive toward convergence. To convince ourselves of the feasibility of any estimation approach, let’s first convince ourselves that the model shows some behavior which any update law can observe to adjust parameters.

Recalling Eqn. 2.13, it is beneficial that we have a set of equations involving frictional coefficient μ but not cable elasticity K_c . The relationship between $|\Delta T|$ and

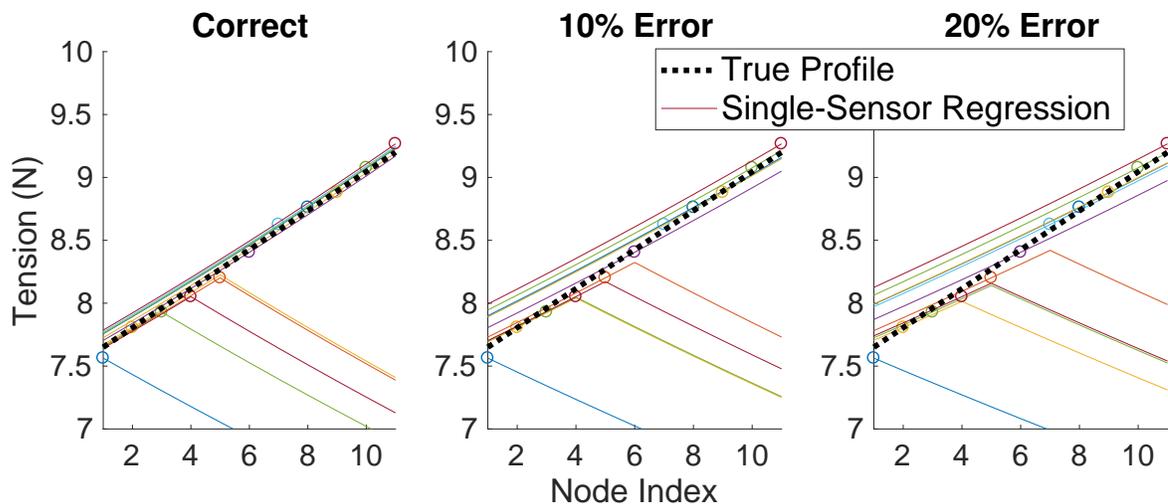


Figure 3.9: As the value of μ used by the sensor fusion model diverges from the correct value, the slope of extrapolated solutions (thin lines) also diverges from the slope of a linear fit of sensor measurements (black). This divergence allows for tuning of μ .

μ is straightforward - as μ increases, so does $|\Delta T|$. Therefore, observing a systematic difference in the slope of the tension-vs-position relationship between measurements and estimations would suggest that the value of μ needs to be adjusted. This can be observed in our multiple-sensor model by intentionally setting an incorrect value for μ , and observing that the slope of estimates diverges both from the slope of the true system and from that of a linear fit of all sensors. Fig. 3.9 demonstrates this phenomenon.

It then follows logically to propose an update law for μ based on this discrepancy such that the system is able to adjust its friction coefficient on-the-fly to reflect changes in frictional forces caused by atmospheric changes, mechanical wear, or other unmodeled effects. While the update law is implicit if an UKF based approach is chosen, simpler approaches might suffice. A first effort might consider representing the value of μ_d as $\mu_{d,0} + k_\mu \text{mean}((T_1 - T_N)_{extrap} - (T_1 - T_N)_{fit})$, or as a similar simple equation.

In the event that $R(x)$ is unknown or uncertain, the same methodology can be

applied to the quantity $\mu/R(x)$. Since Eqn. 2.13 always contains these two quantities together as common scalar multiples, we can apply the same principle used for μ alone to adjust their common value. In this instance, the quantity being estimated varies with axial displacement along the conduit x , such that the slope error used for fitting will also vary as x . This necessitates a higher-order polynomial (or different functional form altogether) be used for fitting of measured data, and calculation of the slope error used in the update law.

Given the array of sensors anticipated in a system contains only one encoder, there exists less opportunity to adjust K_c , since its role in the model is to adjust the *stretch* of the cable in response to tension, as opposed to any singular position. Since we assume only one encoder, at the input, we typically don't have a measurement of the change in the cable's length that isn't produced by the model itself. However, if we restrict estimation of K_c to periods of partial motion, where we assume the node beyond the last moving node is stationary and its position known, then our single encoder is sufficient to measure cable stretch. By observing discrepancies between the expected location of the input⁵ as predicted by the available tension sensors in the partially-moving segment and the measured value, it should be possible to adjust our estimate of K_c so as to drive these quantities toward convergence.

This adjustment would most likely need to be performed deliberately during periods where the output is expected to be stationary and not experiencing any torque (in pull-pull configuration), since the objective of any controller based on this model would be, in part, to bypass partial motion as quickly as possible to avoid a phase lag in tracking of the desired output due to static friction.

⁵ And only the input – it's not safe to assume our system has an encoder at the output to compare against, even though the mathematical system has symmetry here. Weight and size constraints will limit the availability of sensors at the output.

Chapter 4

CONTROLLER DESIGN

Now that we have a dynamic model with desirable properties specific to our use for cable-conduit transmissions, the next practical step is to begin considering the implementation of controllers that use information obtainable with such a model in real-time. There are two modes in which we would like to use the model to ensure desirable system performance:

- To overcome static friction quickly and prevent backlash
- To ensure consistent tracking of desired output tensions during motion

4.1 Stiction Compensation

When beginning motion, it is desirable to compensate for stiction in the system that leads both to initial tracking errors and delay, which in haptic applications can amount to a noticeable distortion of the intended simulated environment. In the absence of a model, a feedback scheme with an integrator will eventually overcome the stiction, but this only addresses the tracking issue—integral control takes time to converge, and while increasing the control gain can decrease this delay, it also tends to introduce undesirable behaviors like oscillations in the feedback variable prior to convergence.

Here, the model can act as an observer for the unmeasurable state that describes what sections of the cable are in motion having overcome static friction, and where frictional forces are still too great to overcome. We can also use the model to estimate how much tension is needed to overcome the friction in those segments which are still stationary by considering Eq. [2.17](#).

We can therefore describe a static friction compensation scheme as follows:

Listing 4.1: Static friction compensation logic

```
1 If LMNs not coincident:
2     Calculate an input torque such that Eq. 2.7
3         satisfied for all non-moving nodes
4     Add required tension to the actuator control signal
```

4.2 Dynamic Friction Compensation

As previously discussed, existing systems using cable conduit transmissions rely on having a high-precision sensing element located at the distal end for feedback control. However, it would be preferable to avoid requiring this sensor, since its inclusion may be impractical for wearable applications. In Chapter 3, an application was discussed in which the model is able to resolve redundant noisy state measurements to a single measurement of the output with improved noise characteristics. This alone is invaluable for a control system, where the model supplied measurement can supply a simple linear controller with an improved or otherwise unavailable measurement.

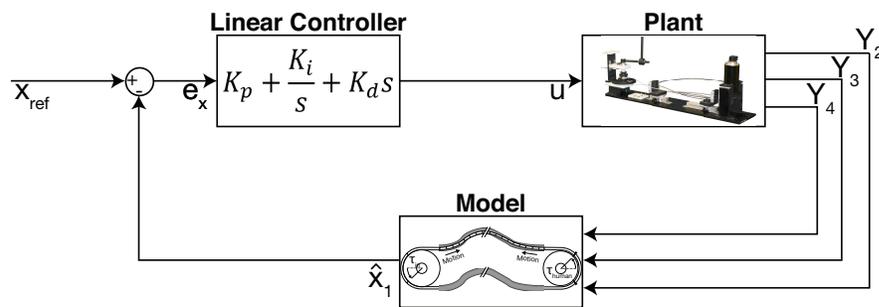


Figure 4.1: A linear controller for a cable-conduit system which utilizes the model to resolve available noisy measurements of cable tension into an estimate of the state of interest, output tension, for feedback.

We can also derive beneficial behavior in the single-sensor case by utilizing the model in a disturbance observer [65], as shown in Fig. 4.2. The disturbance observer formulation uses the measured states to back-predict the system input as a means to estimate disturbances to the system in the presence of sensor noise. This disturbance prediction is combined with the reference tracking controller’s input to compensate for the disturbance while maintaining tracking performance.

Despite relying on measurements to function, this model is suitable for this use due to the assured extra measurement at the motor input to the transmission. A typical application would supply torque commands to the motor, but the presented model will typically utilize the motor’s position sensing capabilities to provide the required measurement of at least one node’s position. After model solution, the calculated motor torque will show a discrepancy with the applied motor torque and provide a disturbance estimate. With model parameters set accurately, this can be used to estimate disturbances in the system.

In the disturbance observer framework, a lowpass filter is typically included to work around issues stemming from the inverse model’s transfer function being non-causal. While our model doesn’t have a causality issue (nor can it be expressed as a transfer function), it is likely that the lowpass filter remains beneficial for rejection of high frequency sensor noise.

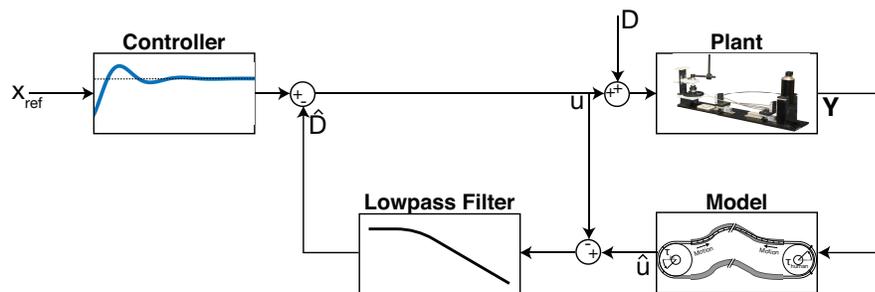


Figure 4.2: A disturbance observer utilizing the model in an inverse capacity to estimate the supplied input from measured outputs. The discrepancy between estimated and applied inputs allows for estimation, and compensation, of disturbance D affecting the system.

The last straightforward form we can consider this model for use in is a feed-forward friction compensation scheme. Given our reference input, we calculate how much tension would be lost to friction across the transmission, and adjust accordingly. Unlike other feed-forward compensation schemes, we must resolve the friction recursively since $F_\mu \propto T$, our frictional losses change as we change input tension. By the form of the tension equations from 2.13, this relationship is $T_{out} = \mathcal{C}(t)T_{in}$. By taking \mathcal{C}^{-1} , we can calculate what tension needs to be applied at the input to create the desired output tension.

Unlike typical feed-forward friction compensations, which express frictional losses in terms of a state (e.g. velocity) known a-priori at a given time, our model has frictional losses proportional to the input variable. This manifests itself in multiplication of the control signal instead of addition to the control signal. For compatibility with other control schemes, it may be beneficial to express this as $u = x_{ref} + x_{ref} \cdot (\frac{T_{in}}{T_{out}} - 1)$ instead of $u = x_{ref} \cdot \frac{T_{in}}{T_{out}}$ as diagrammed in Fig. 4.3 so that other controllers can interact with the feed-through component.

By utilizing one or more of these in conjunction with a high-level controller designed to produce a desired virtual environment, it should be possible to compensate for a large portion of the transmission dynamics and improve system performance to a level not possible for model-free control schemes. The following section of this chapter

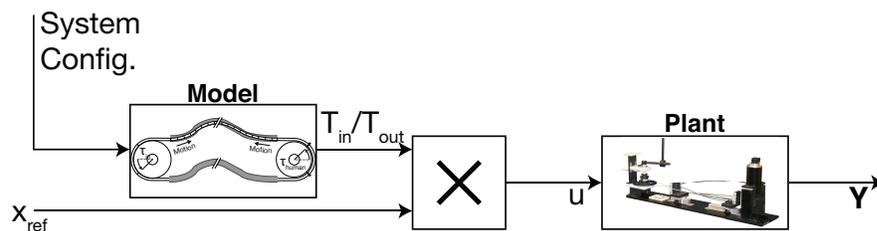


Figure 4.3: Feed-forward friction compensation based on the model. Here, we supply the model with the system configuration only and solve for the relationship between tension at the first and last nodes. This relationship is used to transform the desired output tension into an input tension to be applied.

reports on the implementation of these controllers in our simulation environment, and present a comparison of the results achieved via these controllers.

4.3 Simulated Controller

4.3.1 Simulation 1: One Cable

As an initial verification of the capability of the model to function in a controller, a control system was simulated for a single cable where the control objective is to display an oscillatory tension to a mobile position source at the output. We compare 3 different control systems:

- No friction compensation (Feed-forward only)
- Model-free linear feedback controller
- Model-based feedforward friction compensation

by simulating a single, 10 segment cable with a mobile position source at the output. The control objective is to display a tension at the output that increases with time, and a slope that varies sinusoidally. Pure model-free feed-forward, feedback, and hybrid feed-forward/feedback controllers are compared to a model-based feed-forward/feedback controller. Feedback gains ($K_p = 1$, $K_i = 2$) were constant across all controllers; K_p was chosen to respect passivity limits, and K_i was selected to compromise between model performance and feasibility for use in a physical system without causing oscillations. To demonstrate the robustness of potential control schemes against modeling errors, two error cases are simulated – one where the model has a constant scaling error in the amount of friction, to simulate the effect of errors in parameter estimation; and another where the error in the model varies with time, to simulate the effects of an unmodeled dynamic effect.

Fig. 4.4 shows the result of the simulation. As would be expected, the worst performing controller is to merely supply the desired output tension to the input and ignore all frictional effects. A linear PI feedback controller in cable tension performs substantially better as the integral action takes up compensation of a bulk of the

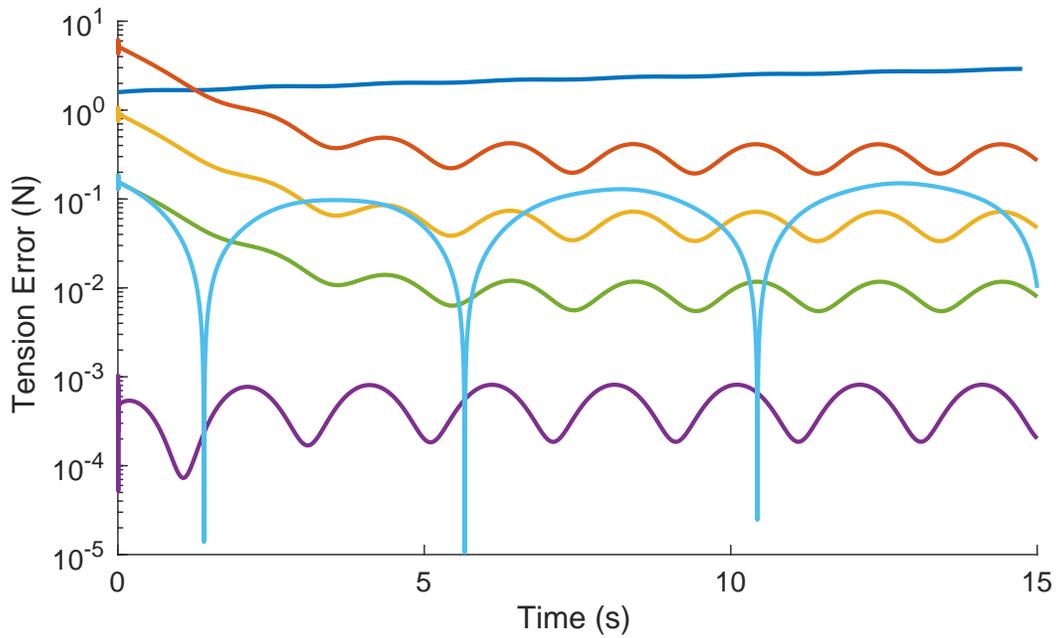
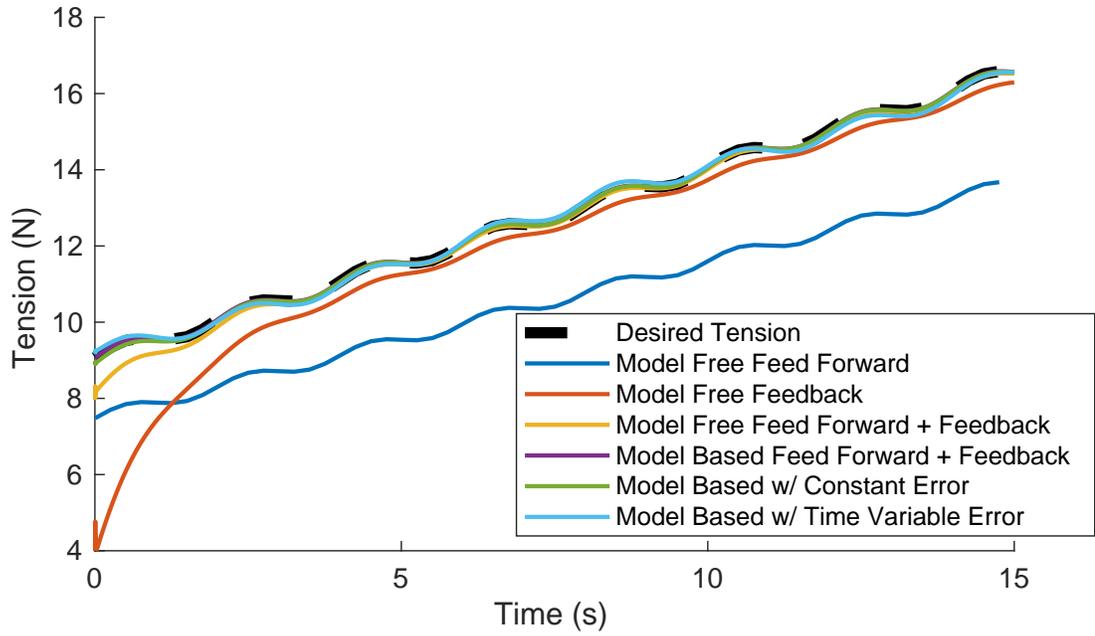


Figure 4.4: Performance of six control schemes at tracking a desired tension profile while the output is in motion. The first three controllers do not utilize the model, and show, in general, worse tracking performance than the model-based controller, even when simulated modeling errors are present.

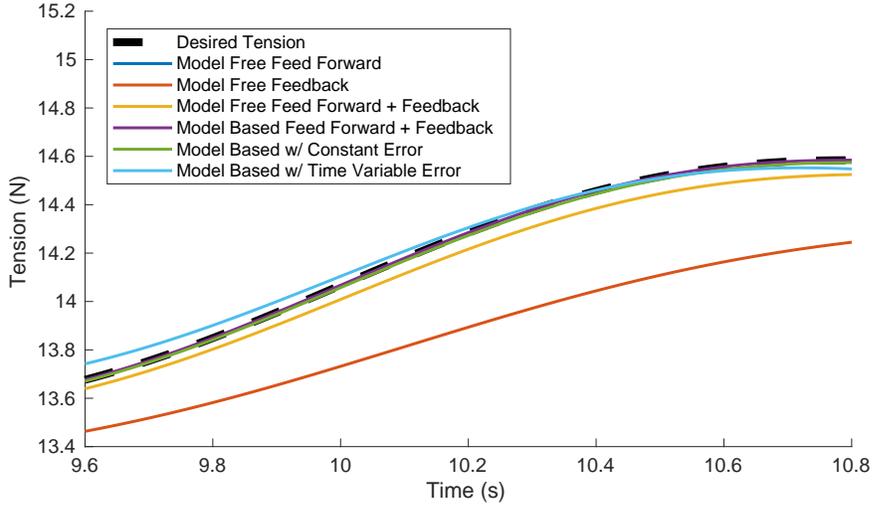


Figure 4.5: A zoomed view of Fig. 4.4 shows that model-free control approaches always lag at the output. The phase of the time-varying error model swings back and forth as the error changes signs. Constant error and error-less model-based controllers are in-phase with the desired output.

friction. However, with a unitary proportional gain K_p and an integral gain of 2, selected such that a physical system using these control parameters would display stable behavior in the presence of noise (and the same gains were used across all feedback controllers), the controller can't keep up with the periodic fluctuations in the desired output, and the output lags the input substantially, as shown in Fig. 4.5. When the feedback action is combined with a feed-forward action, tracking performance is improved by an order of magnitude, and the lag substantially reduced. However, there is still an initial transient where tracking error is as high as 10% as the integrator takes time to ramp up.

The model-based controller resolves this and tracks the output nearly perfectly from controller initialization throughout the entire time period. While a feedback controller exists in parallel, for this test the model matches the plant exactly and the magnitude of the feedback action is orders of magnitude smaller than the feed-forward

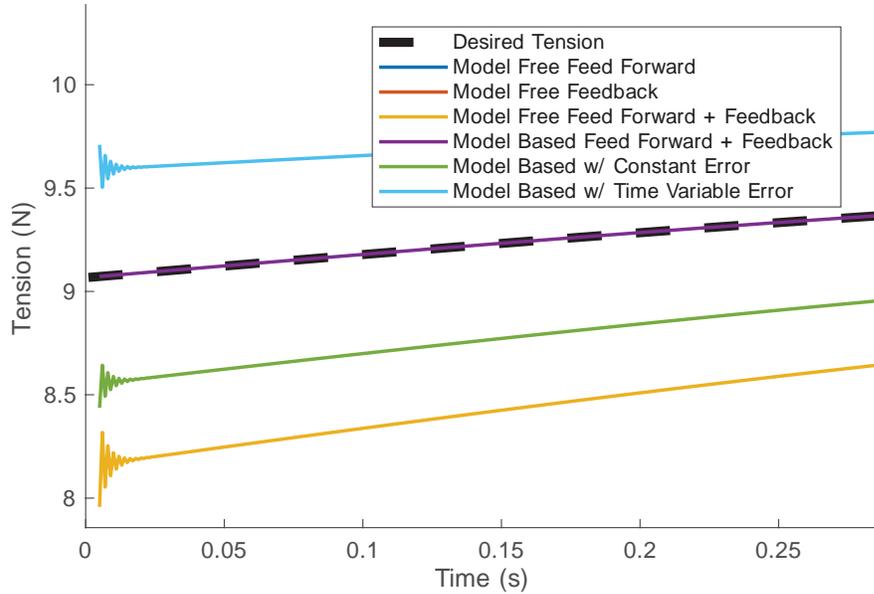


Figure 4.6: Tracking performance of various controllers at time of controller initialization. Since the simulations contain no inertia, controller oscillations settle within milliseconds. In practice, this behavior would need to be considered when selecting controller gains. *Note: Here both model error curves are shown with 10% parameter error to avoid overlap*

action in the absence of sensor noise, model inaccuracy, and actuator dynamics. In fact, the only cause of error is the one sample delay between virtual sensor measurement and application of control to resolve causality in the simulation. When a 3% parameter estimation error is introduced (Green curve in Fig. 4.4), there is an initial error which is quickly taken up by the integrator, resulting in a tracking error an order of magnitude better than the best model-free controller.

The last scenario considered was to introduce a time-varying model error. The specific time variation introduced is a cosine with a period of 3π seconds and an amplitude equal to the 3% used in the constant error case. While this specific time variation is unlikely to be observed in a physical system, it is used here to simulate some model inaccuracy that varies continuously as the system's state changes. In this case, we gain the benefit of good behavior at controller initialization without compromising tracking performance throughout.

An important note about these simulations is that there are no inertias present, since the input is an actuator modeled as an ideal force source, and the output is a person modeled as an ideal position source. This makes it such that in simulation, controller overshoots are reversed in a single timestep. While these are present here, they are observed only in the first several time steps (Fig. 4.6) as they quickly settle in the absence of inertia. In practice, this would present an additional consideration in constructing the controller.

4.3.2 Simulation 2: Two Cables

As a second test, we compare a model-based controller for dynamic friction to a model-free controller in a set of two cables in pull-pull configuration. Here, the radii of input and output pulleys are the same, 10cm, such that the ratio of the transmission is 1:1. The control objective is to display a virtual spring with a stiffness of 1 Nm/radian. The wearer is a position source who moves at constant velocity to an angular deflection of 1.5 radians over 15 seconds, then back to no deflection at 30 seconds. Neither controller has any mechanism for compensation of static friction so that the tracking error during motion can be compared without any confounder factors from stiction compensation. Both controllers utilize the same feedback gains as in the first simulation.

Fig. 4.7 shows the time-domain results for tracking performance. In the absence of stiction compensation, both controllers show comparable performance at motion onset and direction change; however the model-free controller lags the desired trajectory during the entirety of motion, similar to what was seen in the single cable case.

If we instead look in the spring (Torque-angle) domain, as in Fig. 4.8, we observe that the model-based controller renders the spring almost perfectly during most of motion, displaying a small backlash when motion changes due to the lack of compensation, while backlash persists through the entirety of motion for the model free controller.

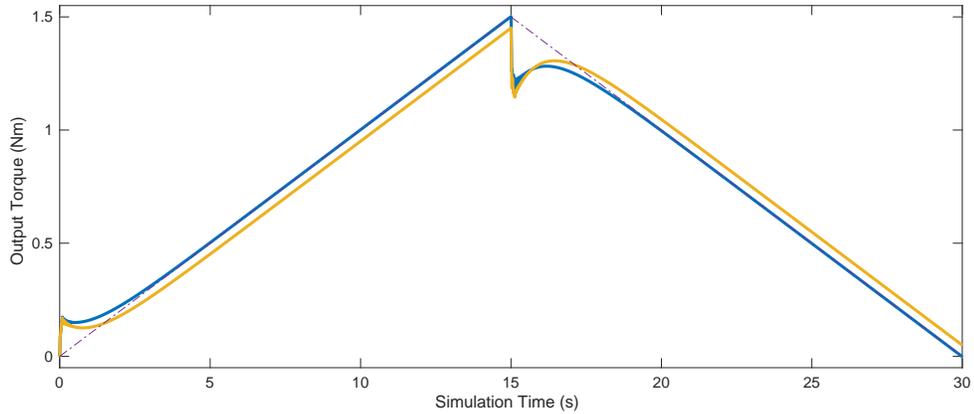


Figure 4.7: Results of a simulation where torque at the input was controlled to display a spring at the output. While neither controller is equipped with static friction compensation, the tracking performance of the model-based controller is far superior to the model-free controller, with no lag or steady state error.

While there is additional work to be done in developing a full controller that includes stiction and disturbance compensation, as well as other features such as those discussed in Chapter 3, these simulations serve as proof-of-concept that the presented model can be integrated into a controller to improve tracking performance at the output of a physical system.

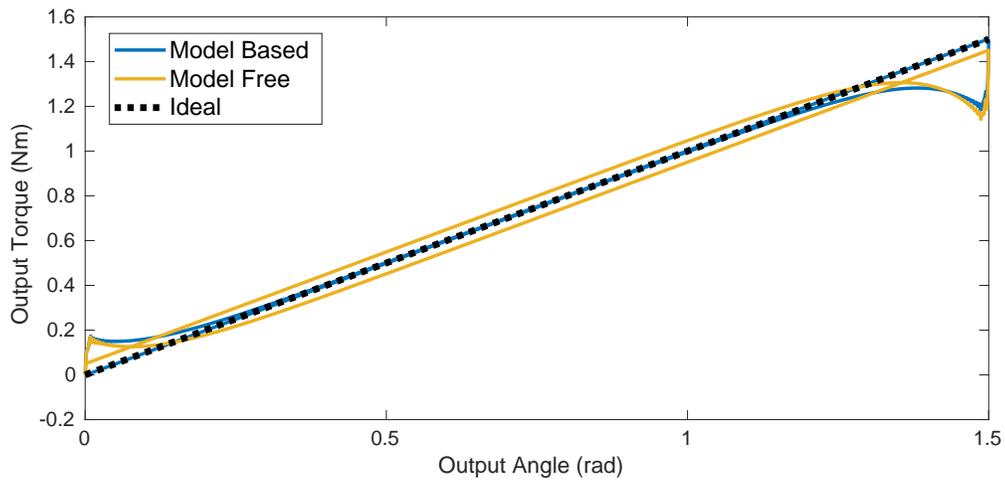


Figure 4.8: Results from the two-cable controller simulation in the Torque-Position domain. Since the desired virtual environment is a spring, the relationship would ideally be linear. While both curves stray from the ideal curve when motion changes direction, the model-based controller lies on the curve during motion, while the model-free controller displays a constant offset.

Chapter 5

DISCUSSION

5.1 Summary of Contributions

This thesis has developed and presented a new model formulation for the study of systems incorporating cable-conduit transmissions. The model was derived directly from mechanical properties of the system, and extends modeling efforts made in previous works by Agrawal [59, 58]. A specific implementation of this model for a typical application was made and validated against a physical system purpose-built for comparison. A novel methodology for solution of the nonlinear system equations was proven and employed to enable computationally efficient use of this model for use in a controller. Avenues for use in state estimation were explored, and while traditional methods, like Kalman filtering, don't work given the algebraic nature of the model, a use case for fusing many unreliable tension sensors into a consistent state estimate was formulated. Methods for parameter estimation during controller use were also considered. Lastly, a discussion of possible controllers this model could be used in was presented alongside simulations of simple controllers in a scenario involving interaction with an active environment.

In aggregate, this thesis sets the foundations for creation of an observer-based control scheme for a cable-conduit transmission to be used for human-interacting robotics. The model developed is sufficiently encompassing that it should be applicable to ongoing research being performed on the topics of on-line cable curvature sensing and in-situ cable strain measurements. The presented solver methodology means that future controller implementations can focus aspects other than the fact that nonlinear models with no closed-form solution are commonly too slow for real-time use.

5.1.1 Limitations

The primary limitation of this research is that for the state estimation and controller work, conclusions are drawn solely from simulation and modeling. In order to truly verify the usability of the schemes presented, they will need to be implemented in hardware. Additionally, validation of the model was performed against a single physical system, and was done primarily to confirm correct behavior of the model moreso than numerical accuracy. A more robust validation would be performed with an array of cable/conduit material pairings to vary frictional constants, and would include a more precise tuning of the underlying frictional model, most likely to a Stribeck curve.

5.2 Research Significance

Cable-conduit transmissions have promise to be the power transmission of choice as the field of rehabilitation robotics moves toward wearable, autonomous systems. Their flexibility, low mass, and safety make them well suited for exoskeleton use, where distal mass is undesirable and the path through which power needs to be transmitted varies as joint angles change. However, their use introduces a large backlash and inefficiency to the system as a consequence of friction between cable and conduit. This makes controlling cable-conduit driven systems a challenge, and poses a substantial barrier to their wide adoption.

Where previous models were built on the premise of a passive environment, we have now created a model designed for an arbitrary active environment. This feature allows for the study of cable-conduit transmissions in human-interacting applications, revealing phenomena such as partial motion in multiple sections of cable and sharing of frictional loads during motion that could not be observed were the transmission represented as a lumped mass. On its own, this model will allow roboticists to determine the effects that changes in configuration during use will have on the transmission's behavior, informing design decisions.

However, for rehabilitation applications, accuracy and reproducibility are paramount to meet the current demand for robots to increase the dosage of therapy that would

be conventionally supplied by a therapist. High-fidelity controllers are needed to accomplish this, and we have shown that a model-based, real-time controller is not only feasible despite the nonlinear nature of the model, but that simple model-based controllers outperform model-free controllers in simulation. Future researchers will be able to adopt these control schemes to allow autonomous exoskeletons to behave like physical therapists.

Once the clinical footwork for autonomous, wearable human-interacting robots has been laid by devices that emulate therapists to increase dosage of therapy, therapists and biomechanists alike can focus efforts on developing new high-level control strategies to improve the walking speed and decrease asymmetry in post-stroke populations. With a model-based low level controller on established devices, testing new controllers requires only implementing the logic for the high-level strategy. Furthermore, human-in-loop optimization strategies will allow for rapid development of assistive controllers by selecting parameters that provide the best outcome measures for a given individual.

In time, it is probable that a model-based low-level controller will be at the core of many wearable robotic devices utilizing cable-conduit actuation. By laying a strong foundation with a model-based controller, this work arms future researchers with a tool they need to develop lightweight, wearable, autonomous exoskeletons possessing the requisite ability to accurately apply torques to their wearer.

5.3 Future Work

The existence of this new model leaves a number of new avenues open for research that fell beyond the scope of this Masters' thesis. While I have attempted to be thorough in ensuring the model is suitable for a wide range of use cases, it was unfeasible to implement every possible use case in hardware. Future work might start out by implementing a cable-bend sensor and including a number of carbon nanotube based strain sensors to allow the model to adjust on-line to changes in cable path, and to estimate its own frictional constants.

The next logical step would be to introduce the self-tuning model into a controller using the paradigms described in Chapter 4. This would require development of a new bench-top device that can change the cable's path during use, either automatically for quantitative assessment, or manually for qualitative assessment. This will reveal the quantitative benefits of using different combinations of model-based control elements, but will require a large amount of effort in implementing the parallelism required to execute both the model solver and control code within the given time window.

Finally, implementation of an autonomous wearable device would close the loop on this research, bringing it back onto the theme of developing robot-mediated rehabilitation paradigms for stroke survivors to improve their mobility, and quality of life.

5.4 Closing Remarks

It is with eager anticipation to see what the future holds for this work that I now leave it for the next wave of researchers. Having completed a large subset of the mathematical and programming legwork in development of this new model, a number of directions lie wide open for improvement of human-interacting robots. The new model is versatile and performs well under the circumstances it is designed for. I trust that the success I have had in development of this dynamic model of cable-conduit transmissions will translate to improved clinical outcomes for individuals undergoing physical therapy in the near future.

BIBLIOGRAPHY

- [1] Sašo Jezernik, Gery Colombo, Thierry Keller, Hansruedi Frueh, and Manfred Morari. Robotic Orthosis Lokomat: A Rehabilitation and Research Tool. *Neuro-modulation: Technology at the Neural Interface*, 6(2):108–115, apr 2003.
- [2] J.F. Veneman, R. Kruidhof, E.E.G. Hekman, R. Ekkelenkamp, E.H.F. Van Asseldonk, and H. van der Kooij. Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(3):379–386, sep 2007.
- [3] U.S. Department of Labor Bureau of Labor Statistics. Physical Therapists, 2018.
- [4] Emelia J Benjamin, Michael J Blaha, Stephanie E Chiuve, Mary Cushman, Sandeep R Das, Rajat Deo, Sarah D de Ferranti, James Floyd, Myriam Fornage, Cathleen Gillespie, Carmen R Isasi, Monik C Jiménez, Lori Chaffin Jordan, Suzanne E Judd, Daniel Lackland, Judith H Lichtman, Lynda Lisabeth, Simin Liu, Chris T Longenecker, Rachel H Mackey, Kunihiro Matsushita, Dariush Mozaffarian, Michael E Mussolino, Khurram Nasir, Robert W Neumar, Latha Palaniappan, Dilip K Pandey, Ravi R Thiagarajan, Mathew J Reeves, Matthew Ritchey, Carlos J Rodriguez, Gregory A Roth, Wayne D Rosamond, Comilla Sasson, Amytis Towfighi, Connie W Tsao, Melanie B Turner, Salim S Virani, Jenifer H Voeks, Joshua Z Willey, John T Wilkins, Jason Hy Wu, Heather M Alger, Sally S Wong, Paul Muntner, and American Heart Association Statistics Committee and Stroke Statistics Subcommittee. Heart Disease and Stroke Statistics-2017 Update: A Report From the American Heart Association. *Circulation*, 135(10):e146–e603, mar 2017.
- [5] Henrik S. Jørgensen, Hirofumi Nakayama, Hans O. Raaschou, and Tom S. Olsen. Recovery of walking function in stroke patients: The copenhagen stroke study. *Archives of Physical Medicine and Rehabilitation*, 76(1):27–32, jan 1995.
- [6] D T Wade, R Langton-Hewer, V A Wood, C E Skilbeck, and H M Ismail. The hemiplegic arm after stroke: measurement and recovery. *Journal of neurology, neurosurgery, and psychiatry*, 46(6):521–4, jun 1983.
- [7] D T Wade and R L Hewer. Functional abilities after stroke: measurement, natural history and prognosis. *Journal of neurology, neurosurgery, and psychiatry*, 50(2):177–82, feb 1987.

- [8] Bart M Demaerschalk, Ha-Mill Hwang, and Grace Leung. US cost burden of ischemic stroke: a systematic literature review. *The American journal of managed care*, 16(7):525–33, jul 2010.
- [9] Fu-Ling Tung, Yea-Ru Yang, Chao-Chung Lee, and Ray-Yau Wang. Balance outcomes after additional sit-to-stand training in subjects with stroke: a randomized controlled trial. *Clinical Rehabilitation*, 24(6):533–542, jun 2010.
- [10] C M Dean and R B Shepherd. Task-related training improves performance of seated reaching tasks after stroke. A randomized controlled trial. *Stroke*, 28(4):722–8, apr 1997.
- [11] R G Lee and P van Donkelaar. Mechanisms underlying functional recovery following stroke. *Can.J.Neurol.Sci.*, 22:257–263, 1995.
- [12] R. J. Nudo, B. M. Wise, F. SiFuentes, and G. W. Milliken. Neural Substrates for the Effects of Rehabilitative Training on Motor Recovery After Ischemic Infarct. *Science*, 272(5269):1791–1794, 1996.
- [13] Giovanni Di Pino, Giovanni Pellegrino, Giovanni Assenza, Fioravante Capone, Florinda Ferreri, Domenico Formica, Federico Ranieri, Mario Tombini, Ulf Ziemann, John C. Rothwell, and Vincenzo Di Lazzaro. Modulation of brain plasticity in stroke: a novel model for neurorehabilitation. *Nature Reviews Neurology*, 10(10):597–608, oct 2014.
- [14] Alex R. Carter, Gordon L. Shulman, and Maurizio Corbetta. Why use a connectivity-based approach to study stroke and recovery of function? *NeuroImage*, 62(4):2271–2280, 2012.
- [15] Assia Jaillard, Chantal Delon Martin, Katia Garambois, Jean François Lebas, and Marc Hommel. Vicarious function within the human primary motor cortex? *Brain*, 128(5):1122–1138, may 2005.
- [16] M Visintin, H Barbeau, N Korner-Bitensky, and N E Mayo. A new approach to retrain gait in stroke patients through body weight support and treadmill stimulation. *Stroke; a journal of cerebral circulation*, 29(6):1122–8, 1998.
- [17] Hugues Barbeau and Martha Visintin. Optimal outcomes obtained with body-Weight support combined with treadmill training in stroke subjects. *Archives of Physical Medicine and Rehabilitation*, 84(10):1458–1465, oct 2003.
- [18] Carolyn Luke, Karen J. Dodd, and Kim Brock. Outcomes of the Bobath concept on upper limb recovery following stroke. *Clinical Rehabilitation*, 18(8):888–898, 2004.

- [19] James C. Grotta, Elizabeth A. Noser, Tony Ro, Corwin Boake, Harvey Levin, Jarek Aronowski, and Timothy Schallert. Constraint-induced movement therapy. *Stroke*, 35(11 SUPPL. 1):2699–2701, 2004.
- [20] R PS Van Peppen, G Kwakkel, S Wood-Dauphinee, H JM Hendriks, Ph J Van der Wees, and J Dekker. The impact of physical therapy on functional outcomes after stroke: what’s the evidence? *Clinical Rehabilitation*, 18(8):833–862, 2004.
- [21] Julien Doyon and Habib Benali. Reorganization and plasticity in the adult brain during learning of motor skills. *Current Opinion in Neurobiology*, 15(2):161–167, apr 2005.
- [22] John W Krakauer. Motor learning: its relevance to stroke recovery and neurorehabilitation. *Current Opinion in Neurology*, 19(1):84–90, feb 2006.
- [23] Pawel Kiper, Andrzej Szczudlik, Annalena Venneri, Joanna Stozek, Carlos Luque-Moreno, Jozef Opara, Alfonc Baba, Michela Agostini, and Andrea Turolla. Computational models and motor learning paradigms: Could they provide insights for neuroplasticity after stroke? An overview. *Journal of the Neurological Sciences*, 369:141–148, oct 2016.
- [24] Erin E Helm and Darcy S Reisman. The Split-Belt Walking Paradigm: Exploring Motor Learning and Spatiotemporal Asymmetry Poststroke. *Physical medicine and rehabilitation clinics of North America*, 26(4):703–13, nov 2015.
- [25] Keith R. Lohse, Catherine E. Lang, and Lara A. Boyd. Is more better? Using metadata to explore dose-response relationships in stroke rehabilitation. *Stroke*, 45(7):2053–2058, 2014.
- [26] S.K. Banala, Seok Hun Kim, S.K. Agrawal, and J.P. Scholz. Robot Assisted Gait Training With Active Leg Exoskeleton (ALEX). *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(1):2–8, feb 2009.
- [27] Joseph Hidler, Wessel Wisman, and Nathan Neckel. Kinematic trajectories while walking within the Lokomat robotic gait-orthosis. *Clinical Biomechanics*, 23(10):1251–1259, dec 2008.
- [28] Andrew Pennycott, Dario Wyss, Heike Vallery, Verena Klamroth-Marganska, and Robert Riener. Towards more effective robotic gait training for stroke rehabilitation: a review. *Journal of NeuroEngineering and Rehabilitation*, 9(1):65, sep 2012.
- [29] C. L. Pollock, L. A. Boyd, M. A. Hunt, and S. J. Garland. Use of the Challenge Point Framework to Guide Motor Learning of Stepping Reactions for Improved Balance Control in People With Stroke: A Case Series. *Physical Therapy*, 94(4):562–570, apr 2014.

- [30] Robert L. McGrath, Margaret Pires-Fernandes, Brian Knarr, Jill S. Higginson, and Fabrizio Sergi. Toward goal-oriented robotic gait training: The effect of gait speed and stride length on lower extremity joint torques. In *2017 International Conference on Rehabilitation Robotics (ICORR)*, pages 270–275. IEEE, jul 2017.
- [31] J.L. Emken and D.J. Reinkensmeyer. Robot-Enhanced Motor Learning: Accelerating Internal Model Formation During Locomotion by Transient Dynamic Amplification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(1):33–39, mar 2005.
- [32] M J Myers and K Steudel. Effect of limb mass and its distribution on the energetic cost of running. *The Journal of experimental biology*, 116:363–373, 1985.
- [33] Alberto Esquenazi, Mukul Talaty, Andrew Packel, and Michael Saulino. The Rewalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury. *American Journal of Physical Medicine and Rehabilitation*, 91(11):911–921, 2012.
- [34] Shirley Ryan. Identify Training Strategies for Progressing Exoskeleton Users Towards Everyday Functional Ambulation, 2017.
- [35] Alan T. Asbeck, Robert J. Dyer, Arnar F. Larusson, and Conor J. Walsh. Biologically-inspired soft exosuit. In *2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR)*, pages 1–8. IEEE, jun 2013.
- [36] Michael Wehner, Brendan Quinlivan, Patrick M. Aubin, Ernesto Martinez-Villalpando, Michael Baumann, Leia Stirling, Kenneth Holt, Robert Wood, and Conor Walsh. A lightweight soft exosuit for gait assistance. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3362–3369, 2013.
- [37] Raymond C. Browning, Jesse R. Modica, Rodger Kram, and Ambarish Goswami. The effects of adding mass to the legs on the energetics and biomechanics of walking. *Medicine and Science in Sports and Exercise*, 39(3):515–525, 2007.
- [38] Steven H. Collins, M. Bruce Wiggin, and Gregory S. Sawicki. Reducing the energy cost of human walking using an unpowered exoskeleton. *Nature*, 522(7555):212–215, jun 2015.
- [39] Juanjuan Zhang, Pieter Fiers, Kirby A Witte, Rachel W Jackson, Katherine L Poggensee, Christopher G Atkeson, and Steven H Collins. Human-in-the-loop optimization of exoskeleton assistance during walking. *Science (New York, N. Y.)*, 356(6344):1280–1284, jun 2017.
- [40] Kirby Ann Witte, Juanjuan Zhang, Rachel W. Jackson, and Steven H. Collins. Design of two lightweight, high-bandwidth torque-controlled ankle exoskeletons. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1223–1228. IEEE, may 2015.

- [41] J. F. Veneman, R. Ekkelenkamp, R. Kruidhof, F. C.T. van der Helm, and H. van der Kooij. A Series Elastic- and Bowden-Cable-Based Actuation System for Use as Torque Actuator in Exoskeleton-Type Robots. *The International Journal of Robotics Research*, 25(3):261–281, mar 2006.
- [42] M. Kaneko, W. Paetsch, and H. Tolle. Input-dependent stability of joint torque control of tendon-driven robot hands. *IEEE Transactions on Industrial Electronics*, 39(2):96–104, apr 1992.
- [43] W. Townsend and Jr. Salisbury, J. The Effect of coulomb friction and stiction on force control. *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, 4:883–889, 1987.
- [44] J. H. Baek, Y. K. Kwak, and S. H. Kim. On the frequency bandwidth change of a servo system with a gear reducer due to backlash and motor input voltage. *Archive of Applied Mechanics*, 73(5-6):367–376, 2003.
- [45] Andre Schiele, Pierre Letier, Richard Van Der Linde, and Frans Van Der Helm. Bowden cable actuator for force-feedback exoskeletons. *IEEE International Conference on Intelligent Robots and Systems*, pages 3599–3604, 2006.
- [46] J.M. Fee and D.J. Quist. A new cable pulling friction measurement method and results. *IEEE Transactions on Power Delivery*, 7(2):681–686, apr 1992.
- [47] M Kaneko, T Yamashita, and K Tanie. Basic Considerations on Transmission Characteristics for Tendon Drive Robots, 1991.
- [48] Makoto Kaneko, M. Wada, H. Maekawa, and Kazuo Tanie. A new consideration on tendon-tension control system of robot hands. *1991 IEEE International Conference on Robotics and Automation (ICRA)*, (April):1028–1033, 1991.
- [49] G. Palli and C. Melchiorri. Model and control of tendon-sheath transmission systems. In *Proceedings - IEEE International Conference on Robotics and Automation*, volume 2006, pages 988–993. IEEE, 2006.
- [50] C. Canudas de Wit, H. Olsson, K.J. Astrom, and P. Lischinsky. A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425, 1995.
- [51] Henrik Olsson, K.J. Astrom, C. Candus De wit, and Gafvert M. Friction Model and Friction Compensation. *Euoropean Journal of Control*, 4(3):176–195, 1998.
- [52] P Dahl. Measurement of solid friction parameters of ball bearings. *Aerospace Corporation El Segundo, CA*, pages REPORT SAMSO–TR–77–132, 1977.

- [53] Thanh Nho Do, Tegoeh Tjahjowidodo, Michael Wai Shing Lau, and Soo Jay Phee. Position control of asymmetric nonlinearities for a cable-conduit mechanism. *IEEE Transactions on Automation Science and Engineering*, PP(99):1515–1523, jul 2015.
- [54] T N Do, T. Tjahjowidodo, M W S Lau, and S. J. Phee. Dynamic Friction-Based Force Feedback for Tendon- Sheath Mechanism in NOTES System. *International Journal of Computer and Electrical Engineering*, 6(3):252–258, 2014.
- [55] T. N. Do, T. Tjahjowidodo, M. W.S. Lau, and S. J. Phee. A new approach of friction model for tendon-sheath actuated surgical systems: Nonlinear modelling and parameter identification. *Mechanism and Machine Theory*, 85:14–24, 2015.
- [56] Qingcong Wu, Xingsong Wang, Lin Chen, and Fengpo Du. Transmission Model and Compensation Control of Double-Tendon-Sheath Actuation System. *IEEE Transactions on Industrial Electronics*, 62(3):1599–1609, mar 2015.
- [57] Yeongtae Jung and Joonbum Bae. Torque control of a double tendon-sheath actuation mechanism in varying sheath configuration. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 1352–1356. IEEE, jul 2017.
- [58] Varun Agrawal, William J. Peine, and Bin Yao. Modeling of Transmission Characteristics Across a Cable-Conduit System. *IEEE Transactions on Robotics*, 26(5):914–924, oct 2010.
- [59] Varun Agrawal, William J. Peine, and Bin Yao. Modeling of a closed loop cable-conduit transmission system. In *2008 IEEE International Conference on Robotics and Automation*, pages 3407–3412. IEEE, may 2008.
- [60] E.T. Thostenson and T.-W. Chou. Carbon Nanotube Networks: Sensing of Distributed Strain and Damage for Life Prediction and Self Healing. *Advanced Materials*, 18(21):2837–2841, nov 2006.
- [61] Brian J. Odelson, Murali R. Rajamani, and James B. Rawlings. A new auto-covariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, 2006.
- [62] Ashok Ramasubramanian and Laura R. Ray. Comparison of EKBF-based and Classical Friction Compensation. *Journal of Dynamic Systems, Measurement, and Control*, 129(2):236, 2007.
- [63] Sarot Srang and Masaki Yamakita. Estimation of discontinuous friction using continuous-discrete unscented Kalman filter for adaptive compensation. *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing, AIM 2013*, pages 429–435, 2013.

- [64] E. A. Wan and R. Van Der Merwe. The unscented Kalman filter for nonlinear estimation. *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium, AS-SPCC 2000*, v:153–158, 2000.
- [65] Wen Hua Chen, Jun Yang, Lei Guo, and Shihua Li. Disturbance-Observer-Based Control and Related Methods - An Overview. *IEEE Transactions on Industrial Electronics*, 63(2):1083–1095, 2016.

Appendix

SEGMENT VELOCITY APPROXIMATION

When they were developing their similar model formulation, Agrawal et. al. [58] chose to represent the velocity that governs a segment's friction by the velocity of the proximal node defining it. Regardless of this decision was made consciously or if it was never thought about, for their choice of model and constraints it works out to be correct. Since their friction only cares about $\text{sign}(\dot{u})$, the incorrect magnitude from taking one node's velocity to be the velocity of the entire segment is inconsequential, and since motion propagates monodirectionally with respect to the spring-like environment, there is no possibility that the sign of the velocity's segment differs from the velocity of its proximal node. As the math works out, choosing the distal node would have also worked mathematically since the partial-motion constraint equation doesn't need the sign of velocity either.

However, as the system grows in complexity, this assumption quickly becomes unusable. The first issue is a straightforward consequence of bi-directional motion propagation: when motion in different directions can originate from each end of the cable, it is possible to find that a segment is being tensioned (or de-tensioned) from both ends. Let's look at one such case in a toy cable with 5 segments and 6 nodes:

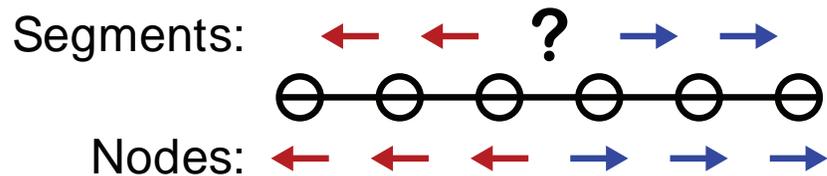


Figure A.1: Sample cable being pulled from both ends.

As the cable is being pulled from both ends, we see that determining the velocity of the segment where motion changes direction is not straightforward given only the signs of velocity at nodes. In fact, with this information provided, we cannot distinguish between cases where

- The segment has net motion toward the left
- The segment has net motion toward the right
- The segment is being stretched uniformly and has no net motion

However, it is critical to the passivity of the model that this be determined accurately. Depending on the value of velocity, the model predicts respectively that

- Frictional forces in the segment are to the right, decreasing tension across the segment
- Frictional forces in the segment are to the left, increasing tension across the segment
- Frictional forces in the segment balance out, and the tension at both ends of the segment is identical

In this example, it is clear that choosing to trust the proximal node exclusively can lead to errors that end up applying frictional forces **in the same direction of motion** for the segment. This would make the frictional model nonpassive, which not only is incorrect and nonphysical, but unacceptable for use in a controller since it suggests that there are circumstances under which the model could be made to diverge.

As it turns out, a first-order approximation of taking the average of both nodes defining the segment is sufficient to ensure velocity selection can't the model to become nonpassive. When we assume that curvature radius is constant through a segment¹,

¹ A required assumption for this to be passive. If $R(x)$ can vary within a segment, the net frictional force could be opposite the mean velocity under some circumstances where a large change in the radius would make tension from one end be affected by friction differently on one side of the segment than the other, given that there exists the possibility of a stationary segment between the two opposite mobile segments

the direction of the net frictional force acting on the segment is always the same as the mean velocity of the segment.

To motivate this, assume a single segment of a cable, whose nodes move with velocities V_1 and V_2 . Since we take the radius of curvature and frictional coefficient to be constant over this segment, the magnitude of frictional forces are constant through the segment. This uniformity enforces that if we were to split this segment in half, to form two segments, the new middle node would have velocity $(V_1 + V_2)/2$. The sign of this velocity matches the sign of at least one of its neighbors². Since we assumed uniformity of the magnitude of friction, let's call this f , the subsegment of matching sign has net frictional force of magnitude equal to the uniform magnitude integrated over its entire length. The subsegment has force magnitude equal to the difference between an integral over a region moving in one direction of length a , and over the remainder of the subcable moving in the other direction. This results in the inequality

$$\left| \int_0^a f \, dx + \int_a^{L/2} -f \, dx \right| < \left| \int_{L/2}^L f \, dx \right| \quad (\text{A.1})$$

which states that the sign of the net friction agrees with the sign of the subsegment agreeing with the mean node. Therefore, assuming the velocity of the segment is that of the mean of its nodes never results in a frictional force in the true direction of motion.

This becomes even more critical if the underlying frictional model is adjusted to include a viscous term as discussed near the end of Chapter 2. Clearly getting the sign wrong is bad when including a viscous term, but simply getting the amplitude wrong would also lead to numerical inaccuracy. In this case, the inaccuracy would be systematic, not random, creating an undesirable bias in the model as a result of trying to improve its accuracy with a more realistic friction model. While the mean velocity may not be the true bulk velocity of a segment, it is a sufficiently accurate approximation to make for both ensuring passivity and maximizing numerical accuracy without increasing computational effort dramatically.

² The case where this mean is 0 is trivial, one subsegment's forces cancel the others exactly, resulting in no net friction—as is computed by a segment velocity of zero.