End Effector Load Force Estimation and Control for a Multi Elastic Link Robot Arm

Master Thesis

submitted to

Institute of Control Theory and Systems Engineering

Faculty of Electrical Engineering and Information Technology

Technische Universität Dortmund

by

Russell Schloss

Philadelphia, Pennsylvania, United States

Date of Submission: September 25, 2014

Responsible Professor:
Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram

Academic Supervisors:
Dipl.-Ing. Jörn Malzahn

"*For my parents*"

Abstract

Elastic link robot manipulators are an exciting field, with many possible future applications. Previous work into force control of elastic link robots has often been of significant mathematical complexity. The goal of this thesis is to present a framework by which standard robot dynamics equations can be used to calculate payload estimates, tip-force estimates, and perform force control tasks with elastic link manipulators. This greatly simplifies the task of applying elastic link manipulators to real world problems.

This thesis begins by building a general framework for translating static torque-referenced robot equations into strain-referenced equations. The experimental performance for predicting strain is evaluated. The need for further dynamic components is shown and added to this model. Joint angle Kalman filtering is applied and shown to additionally improve this model.

Various techniques are applied to the estimation of payload mass for both static and dynamic manipulators. The performance of each is discussed, and a sensor fusion scheme for static payload estimation is demonstrated. Further work estimating payload mass while in motion is presented, and a dual extended Kalman filter is shown to provide accurate results.

Similar approaches are taken to estimate effector tip forces. These estimates, in combination with an existing active vibration damping algorithm, are used to experimentally demonstrate the effectiveness of simple force controllers traditionally used for rigid link manipulators. From the results of this thesis, it can be concluded that, in combination with vibration damping algorithms, standard rigid-link models and force control algorithms can be adapted for use with elastic link manipulators.

Contents

1
Background

1.1. Motivation

Industrial robotics have traditionally been large and heavy, utilizing structurally stiff rigid links to minimize vibrations and deflections. The mass of these rigid links poses safety risks, and in most settings, humans and robots must be strictly separated. Allowing humans to work safely in proximity to robots could increase the opportunities for the application of automation.

Flexible or elastic robot manipulators are a promising avenue to allow humans to inter-operate with robots. While often referred to as elastic links, this thesis will focus on elastically deforming links and will refer to both elastic and elastic links as elastic. Manipulators with elastic links tend to be cheaper and use less material. Reduced link mass can also increase actuation speeds, reducing cycle times in factory settings (Benosman et al. 2004; Lew et al. 1993). By increasing compliance, elastic link robots increase safety during collisions, due to a reduction in impact energy (Garcia et al. 2001).

The reduced weight of elastic manipulators makes them ideal for space applications where the mass of a launch is a primary cost. Link elasticity is a phenomenon that also occurs in existing machinery such as cherry pickers, fire rescue ladders, and cranes. Being able to better deal with vibration in these systems could have immediate real-world benefits. Link elasticity can also be exploited in a biologically inspired ways to provide a tactile sense of obstacles (Malzahn et al. 2014c).

In real-world applications of robot manipulators, it is often necessary to not only control the position but also the force of contact. Tasks such as painting, polishing, grinding, or feeling for randomly placed parts in a workspace are all common tasks that could be performed with an elastic link manipulator.

As will be shown in Section 1.2, the field of elastic link robot force control is an exciting avenue of research due to the benefits listed above. This thesis will develop a method for using standard rigid link robotic dynamics equations with elastic links to predict link strain in static and dynamic conditions. These equations will then be applied to payload and force estimation tasks. Lastly, simple force control tasks will be performed using an elastic link manipulator.

1.2. Related Works

A large body of work exists for rigid link robotic manipulators as well as force control with these rigid link manipulators. A smaller, but still extensive, body of work exists on positioning control and modeling of elastic link manipulators. A survey of position control techniques for elastic link robot manipulators can be found in Benosman et al. (2004). The amount of research combining the topics of elastic link robot manipulators and force control is quite sparse. It is also worth noting that there appear to be limited experimental setups for elastic link manipulators, with many papers relying on only simulations. Most experimental setups are only single-link, and most are horizontal to the ground so as to negate gravitational effects. Wherever possible, this thesis will focus on presenting real results from TUDOR, a 3 degree-of-freedom robot manipulator with two elastic links (described in Section 1.3).

A large category of the control strategies used for problems involving force control of robotic manipulators could be classified as those employing singular perturbation method to divide variously modeled systems into slow and fast subsystems before applying control algorithms to each.

Matsuno et al. (1994) is one of the first to do this with elastic link manipulators, approximating a horizontal, single, elastic link manipulator in contact with b-splines, and subsequently applying singular perturbation method to reduce the model order and find a fast and slow subsystem. Hybrid force/position torque controllers for the slow and fast subsystems are then designed. Matsuno et al. (1994) further notes that the slow subsystem is equivalent to the dynamic equations of a similarly dimensioned rigid manipulator , while the fast subsystem encompasses the vibrational dynamics of the elastic link. This was also noted in Siciliano et al. (1988) for elastic link robots not in contact. This is an important observation that this paper will take advantage of.

Yang et al. (1995) go on to adapt a similar strategy to horizontal manipulators with two elastic links by employing an adaptive hybrid force/position control structure. More recently, Park et al. (2002) applied a similar strategy to an effector with one elastic link, actuated by pneumatic air muscles. A control structure built from a composite scheme based on sliding mode $\mathcal{H}$-infinity controllers and pole placement is used and tested experimentally. Similarly, Lin et al. (2003) apply the singular perturbation method and then use hierarchical fuzzy controllers for a similar task of force/position control with a surface constraint.

Others have developed control strategies that don't rely on the singular perturbation method. Matsuno et al. (1991) derived equations for a horizontal manipulator with one rigid and one elastic link using Hamilton's principle. This system is then simplified by assuming it is quasi-static and used to build a hybrid/position force controller which is tested experimentally.

Lew et al. (1993) modeled multiple-link, redundant, elastic manipulators with multiple points of contact. A control strategy employing bracing against the environment to damp vibrations, while satisfying control constraints, is developed. Dynamic system models are transformed into constrained and unconstrained subspaces using singular value decomposition, reduced in order, and position and force controls designed in these subspaces.

Yim et al. (1993) presents a hybrid position/force control method using an elastic manipulator and constraint surface with non-linear inversions of an input-output map and linear feedback stabilization.

Suzuki et al. (2002) modeled a single, elastic link, horizontal manipulator as a two mass resonance system. A resonance ratio control method, with a reaction torque observer, was then applied. A reaction torque and force feedback controller is then used in combination with disturbance rejection methods. This was applied experimentally to a single elastic link manipulator.

Some papers on force control with elastic link manipulators have covered the various stability and bandwidth controller issues common to this type of system. Chiou et al. (1988) used models based on modal analysis of a one-link elastic manipulator with proportional force controllers. It was demonstrated that link elasticity contributes to control instability for force control. In Li (1990), the inherent bandwidth and performance limitations of force controlled elastic link systems are covered. Looking at single link elastic manipulators, the possibility of adding additional actuation to reduce bandwidth limitations is discussed. Bazaei et al. (2011) used output redefinition with a horizontal, single, elastic link to improve the force control bandwidth.

Another group of papers for force control of elastic link manipulators deal with topics of collision and impact of manipulators with an external environment. Chapnik et al. (1990) built a single elastic link finite-element model with friction and impact modeling. An open loop computed torque controller is built and shown to damp external impacts.

Ueno et al. (1998) developed a method for sensing the position of contact on a single elastic beam by examining the beam's natural oscillation frequencies.

Garcia et al. (2000) developed elastic antennae that can be attached to existing, rigid link, industrial robots for collision detection. Assuming a single vibration mode and simple contact model, 3 different force control schemes are presented for a single elastic link. These include collision detection/impact, direct (using PID) and indirect force control methods.

Ching et al. (2003) focused on the fact that most modal modeling efforts for elastic ignore higher frequency modes. An infinite-dimensional model of a single elastic link colliding with the environment is presented while under control from a proportional derivative force controller.

Malzahn et al. (2011) developed a proportional strain and input shaping damping algorithm for use with flexible link manipulators. This was successfully applied to the 3 degree-of-freedom, 2 link elastic manipulator TUDOR (described in Section 1.3). This damping algorithm will be discussed further in Section 1.3.1. Malzahn et al. (2014c) used model identification to build a linear parameter model of joint torques from strain of TUDOR. Generalized momentum is then used to detect collisions and react with various behaviors including indirect force control. Details of this modeling method is expanded upon in Malzahn et al. (2014a) where a horizontal, analytic, Euler-Bernoulli beam model for elastic links is made linear in parameters and compared in performance to a data-driven approach using an extreme learning machine (ELM) for the tasks of strain and motor current prediction. Chapters 2 and 3 of this thesis extend the linear parameter model, developed in these papers, by allowing the analytic Euler-Bernoulli beam model to account for varying link angles. Additionally, a framework for translating typical rigid link equations for robot joint torque into equations related to strain on elastic link robot arms will be presented in Chapter 2.

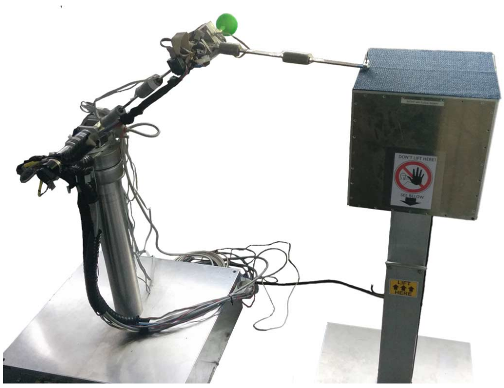


Figure 1.1.: TUDOR touching the contact force cube

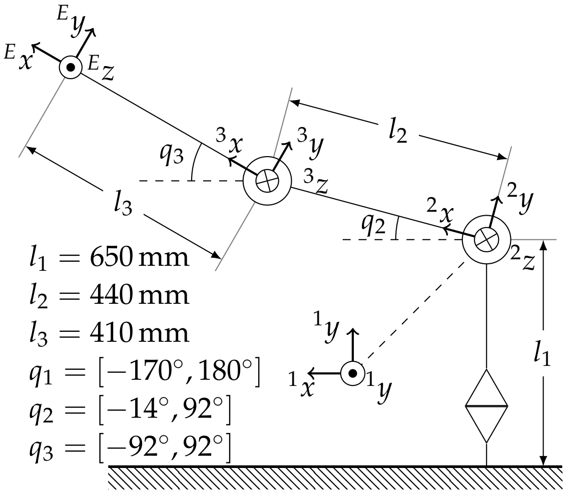


Figure 1.2.: Equivalent rigid body kinematics diagram for TUDOR

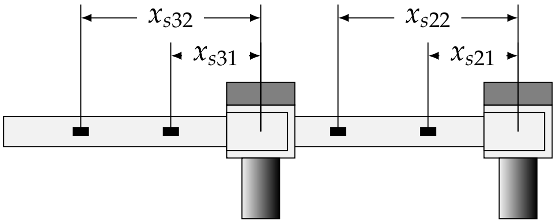


Figure 1.3.: TUDOR top view, showing strain gage placement on links 2 and 3

1.3. Experimental Setup with TUDOR

TUDOR (**T**echnische **U**niversität **D**ortmund **O**mni-elastic **R**obot) is an elastic link robot arm driven by 3 brush-less DC motors with low-backlash planetary gearing. The first joint is located in the base, and the second and third joints drive the two elastic links. This can be seen in the equivalent rigid body kinematics diagram in Figure 1.2. The elastic links are built from clamped steel springs with a height of 4mm and a width of 15mm as described in Malzahn et al. (2014b).

The primary TUDOR sensors used in this paper are joint encoders and 4 strain gage pairs attached to the elastic links. The link 2 strain gages are placed at 46 and 260 mm from joint 2 and the link 3 strain gages are placed at 5 mm and 235 mm from joint 3 as in Figure 1.3. The strain gages are numbered in this paper as 21, 22, 31, and 32, where the first numeral describes the link, and the second is the index of the strain gage. As the clamped steel springs for links 2 and 3 are not ideally clamped, unknown effects may be present in the strain gage behavior.

A contact force measurement cube is also attached to the control systems of TUDOR. This is a pedestal with a 6-axis force-torque sensor mounted inside of a 30 cm diameter metal cube. This can be used for recording collision and contact forces with a resolution of 0.05 N during experiments.

As much previous work with force control and elastic link robots has been in simulation or with horizontal single link experimental setups, this thesis will focus on experimental results with TUDOR. While TUDOR will be used throughout this paper, the same techniques can be applied to serial elastic link robot manipulators with different numbers of links in different configurations.

1.3.1. Damping Algorithm

Depending on the current payload, TUDOR's effector will oscillate roughly 10 cm, with eigenfrequencies ranging from 1 to 13Hz in the x-z plane of the 2nd joint frame (Malzahn et al. 2014c). To damp these oscillations, an active damping scheme composed of strain feedback is used. Before strain feedback can be used, the static strain is removed by using a moving average. The loop is then closed with a proportional strain feedback being fed into the velocity controller. This strain feedback method has been shown to be quite effective for robustly damping vibrations on TUDOR under varying payloads and external stimuli. More details on the vibration control of TUDOR can be found in Malzahn et al. (2011). During the experiments conducted in thesis, this vibration control algorithm remained enabled.

As shown in Section 1.2, a common technique in previous work is to divide an elastic link into slow and fast subsystems using the singular perturbation method for model order reduction. As shown in Siciliano et al. (1988) and Matsuno et al. (1994), for force control problems, using the singular perturbation method can result in a slow subsystem of the same order, and often similar in structure to, the equivalent rigid link robot dynamics. The fast subsystem then uses the slower subsystem's state variables as parameters and usually can be said to characterize the vibrations of the system. This is extremely advantageous as it has been shown that once the fast subsystem controller is designed, typical control strategies used for rigid link robot manipulators can be used for the slow subsystem (Siciliano et al. 1988). The damping algorithm described above will be used throughout this paper and in Chapter 6, when the topic of force control is approached, simple rigid link force control methods will be shown to be effective when using this form of composite control architecture.

1.4. MERIt Dataset

MERIt (**M**ulti-**E**lastic-Link **R**obot **I**dentification Datase**t**) is a freely available dataset focused on providing researchers with data to investigate elastic link robotics (Malzahn et al. 2014b). At the time of this paper, MERIt contains 3 datasets recorded on the TUDOR robot and summarized in Table 1.1. The first, TUD01, referred to in this paper as MERIt01, contains 80 second long runs with payloads between 0 and 400 grams with the robot in motion without active damping. In the second dataset, TUD02, referred to in this paper as MERIt02, contains similar data but with the active damping algorithm designed by Malzahn et al. (2011) activated. The third dataset, TUD03, referred to in this paper as MERIt03, contains 15 seconds of data from the robot at rest in various poses in the working space with weights from 0 to 500 grams. In total, MERIt03 contains 2940 15 second clips of data.

Name	Description
MERIt01	Undamped motion with various payloads
MERIt02	Damped motion with various payloads
MERIt03	Static poses with various payloads

Table 1.1.: Overview of the datasets available in MERIt.

1.5. Overview

In Chapter 2, the Euler-Bernoulli beam theory and process behind building equations for predicting elastic link strain in static conditions is covered. Starting from basic assumptions, a general process for generating equations for predicting strain on multi-link robot arms is developed. This is then used to formulate models that are linear in their parameters, which are trained with linear regression against experimental data.

Chapter 3 starts by justifying the need for further additions for predicting strain in dynamic conditions. It then covers the development of these additional components and details the performance improvement against experimental data. Further performance improvements are also demonstrated through the use of Kalman filtering for joint angle derivatives.

Chapter 4 discusses payload estimation. It begins with a discussion of the problems inherent in payload estimation and moves on to the problem of payload estimation with a robot arm at rest. Sensor fusion schemes are investigated for reducing payload estimation error. Payload estimation of a robot arm in motion is then investigated and various methods are evaluated on experimental data.

Chapter 5 covers the the estimation of strain when in the robot's effector is in contact. In addition, it covers the estimation of contact forces from strain. The collection process for two contact force datasets is discussed. Strain and force estimate equations are then derived and tested against the two datasets. The results and accuracy are discussed, as the effectiveness of these force estimation is relied upon for force control applications.

Chapter 6 covers the topic of force control. A general overview of force control is given and two force controllers are designed. These two force controllers are tested experimentally using TUDOR against 4 force control tasks and the results are discussed.

Chapter 7 finishes by covering various conclusions as well as the outlook of elastic link robots and the possibilities for future work.

2
Static Link Strain Prediction

In this chapter the basics of Euler-Bernoulli beam internal bending moments and strain models are first presented and applied to TUDOR's link 3. Combined with an experimental dataset MERIt, this will yield a model of strain for TUDOR's link 3. It will also provide insights which will lead to the development of a general method for converting standard torque-referenced robot equations to strain-referenced equations. This method will then be used to build equations for TUDOR's second link. A demonstrably accurate model will be derived by making the equations linear with respect to the parameters and applying linear regression. The methods developed in this Chapter will often refer the TUDOR robot, but are designed to be general and applied to other situations and joint configurations.

2.1. Internal Bending Moment

In this paper, robot links at rest will be viewed as a cantilever Euler-Bernoulli beams with various forces and moments applied. This relies on the following assumptions being true (Carrera et al. 2011; Malzahn et al. 2014a):

- Any tip deflections should be small in relation to the beam's length.
- The beam's neutral fiber is treated as having constant length.
- All deflections are elastic and cause no deformation.
- Cross-sections of the beam, a differential distance apart, can be treated as parallel because deflections are small.
- Angular distortion of differential beam elements is small in relation to transverse distortion.
- The beam's material and properties are treated as uniform and homogeneous.

The internal bending moment relates external forces and moments on a beam to internal stresses in the beam (Beer et al. 2012; Gere et al. 2012). For the purposes of this paper, using the principle of superposition, the internal bending moment $M(x)$ of a beam at distance x from the cantilever will be considered to be composed of the bending moment from the beam's weight due to gravity $M_g(x)$, a force applied perpendicular to the beam tip $M_f(x)$, and a possible moment applied at the beam tip $M_M(x)$ as shown in Figure 2.1. The total internal bending moment of the beam is constructed from the sum of these effects (Ruina et al. 2014; Young et al. 2002).

$$M(x) = M_g(x) + M_f(x) + M_M(x) \tag{2.1.1}$$

By successive integration along a length l, internal shear force V, and internal bending moment M can be found for any distribution of forces along a beam $w(x)$ at distance x from the cantilever (Ruina et al. 2014; Young et al. 2002).

$$V(x) = -\int_x^l w(x')dx' \tag{2.1.2}$$

$$M(x) = -\int_x^l V(x')dx' \tag{2.1.3}$$

Applied Tip Force

The generalized forces applied to the beam tip can be viewed as the 6-dimensional vector $\mathcal{F}$ in Equation 2.1.4. This vector is composed of forces in each of the 3 Cartesian axes of the beam tip and moments about those axes. Based on the previous assumptions, only the f_x, f_y and n_z components of the generalized force vector will be considered to effect beam bending.

$$\mathcal{F} = \begin{bmatrix} f_x & f_y & f_z & n_x & n_y & n_z \end{bmatrix}^T \tag{2.1.4}$$

A force at the beam tip f_y, perpendicular to the beam's length l, produces a constant internal shear force with respect to the beam length as in Equation 2.1.5 and an internal bending moment as in Equation 2.1.6.

$$V_f(x) = f_y \tag{2.1.5}$$

$$M_f(x) = f_y(l - x) \tag{2.1.6}$$

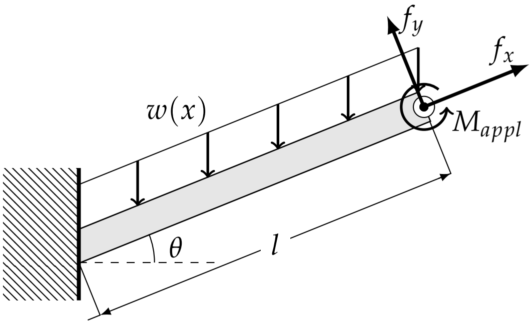


Figure 2.1.: An angled cantilever beam with a distributed force $w(x)$ due to the beam's mass under gravity, tip forces f_x and f_y, and an applied moment about the z-axis $M_{appl} = n_z$.

Beam Distributed Mass

The beam's distributed mass contributes to the internal shear force of the beam and subsequently to the internal bending moment. The force per unit length ρ_b is calculated as in Equation 2.1.7 for a beam of mass m and gravitational constant g. The beam is allowed to rotate about one end (as in Figure 2.1) to approximate a static revolute joint. It is assumed that, under typical loading conditions, the beam will not act as a buckling column. To account for the dependence of the distributed load on the beam angle with respect to horizontal θ, a cosine term can be added as seen in Equation 2.1.8 which carries over in to Equations 2.1.9 and 2.1.10.

$$\rho_b = \frac{mg}{l} \tag{2.1.7}$$

$$w(x) = \rho_b \cos\theta \tag{2.1.8}$$

$$V_g(x) = \rho_b(l - x)\cos\theta \tag{2.1.9}$$

$$M_g(x) = \frac{-\rho_b}{2}(l - x)^2 \cos\theta \tag{2.1.10}$$

Applied Moment

The bending moment applied to the tip of a beam about the z-axis can be considered to distribute evenly throughout the beam's internal bending moment (Young et al. 2002).

$$M_M(x) = M_{appl} \tag{2.1.11}$$

Column Buckling

The assumption that the beam does not experience any column buckling would classify the beam as a long column and means effects from compressive loading can be considered negligible. The lack of column buckling can be justified further by estimating the buckling critical load for the beam. The critical load force F_{xc} can be calculated in Equation 2.1.12 for a beam with Young's Modulus E, area moment of inertia I, length l, and column effective length factor K (Gere et al. 2012). Using the beam in Figure 2.1, this represents the case of a force applied in the F_x direction, while angle θ is 90°.

$$F_{xc} = -\frac{\pi^2 EI}{(Kl)^2} \qquad F_{xc,2} \approx -213.5N \tag{2.1.12}$$

The estimate of the critical column buckling load for TUDOR's second link $F_{xc,2}$ is estimated above in Equation 2.1.12, using a effective length factor of 2, as -213.5 N. To reach this amount of force the link would have to be vertical and a roughly 21.8 kg weight applied. This is clearly outside the scope of the normal operating conditions of a robot of such stature whose typical loads range from 0 to 500 grams.

Beam slenderness ratio λ is used as a method to classify columns (Gere et al. 2012). It can be calculated from the effective length and least radius of gyration as in Equation 2.1.13 with beam cross-sectional area A.

$$\lambda = \frac{Kl}{\sqrt{I/A}} \qquad \lambda_2 \approx 745 \tag{2.1.13}$$

An estimate of TUDOR's second link slenderness ratio λ_2 is 745, much higher than the typical minimum value of 200 required for a steel beam to be classified as a long column (Gere et al. 2012).

2.2. Relating Internal Bending Moment To Strain

The internal bending moment $M(x)$ of a beam at position x can be related to strain $\mathcal{E}(x)$ at that position as in Equation 2.2.1, with Young's modulus E, area moment of inertia I, and distance of beam surface from the neutral fiber y_b (Ruina et al. 2014).

$$M(x) = \frac{EI}{y_b}\mathcal{E}(x) \tag{2.2.1}$$

2.3. Modeling TUDOR Link 3 with Tip Force

The last link of the TUDOR robot (link 3) with a single perpendicular tip force f_y, presents the simplest case for estimating strain from a known tip force.

As this is the last link, with no external moment applied, the internal bending moment of the static case can be represented as in Equation 2.3.1 below with link 3 strain gage position x_{s3}.

$$M_3(x_{s3}) = M_f(x_{s3}) + M_g(x_{s3}) \tag{2.3.1}$$

Additionally, the static behavior of link 3 with a perpendicular tip force f_y is nearly identical to its behavior with a payload mass m_4. As previously mentioned in Section 2.1, compression of the beam by a payload mass is negligible and not detectable by the strain gage configuration on TUDOR.

The bending moment can then be related to strain using Equation 2.2.1 as seen in Equation 2.3.2. As E, I, and y_b are all assumed to be constant, the strain can be related to the internal bending moment by a proportional constant. The ability to pull out constants and treat them as proportional constants will be used extensively in the following sections.

$$\mathcal{E}_3(x_{s3}) = \frac{EI}{y_b}(M_f(x_{s3}) + M_g(x_{s3})) \tag{2.3.2}$$

2.4. Linear Relationship Between Payload and Link 3 Strain

The MERIt03 dataset (presented in Section 1.4) of the TUDOR robot with various payloads in various static configurations is examined. In Figures 2.2 and 2.3 the strains from link 3 are plotted with respect to link 3 angle to the horizontal (joint 2 and 3 angles added together). Each payload is plotted as a separate line series. It can be clearly seen that there is a trigonometric relationship between the link angle and strain. Additionally, for a given link angle, the payload increases the magnitude of strain. The plots for strain gages 31 and 32 are almost identical plots, with the exception of the magnitude. The values of strain from strain gage 32 in Figure 2.3 are a less than than half of the strain values from strain gage 31 in Figure 2.2.

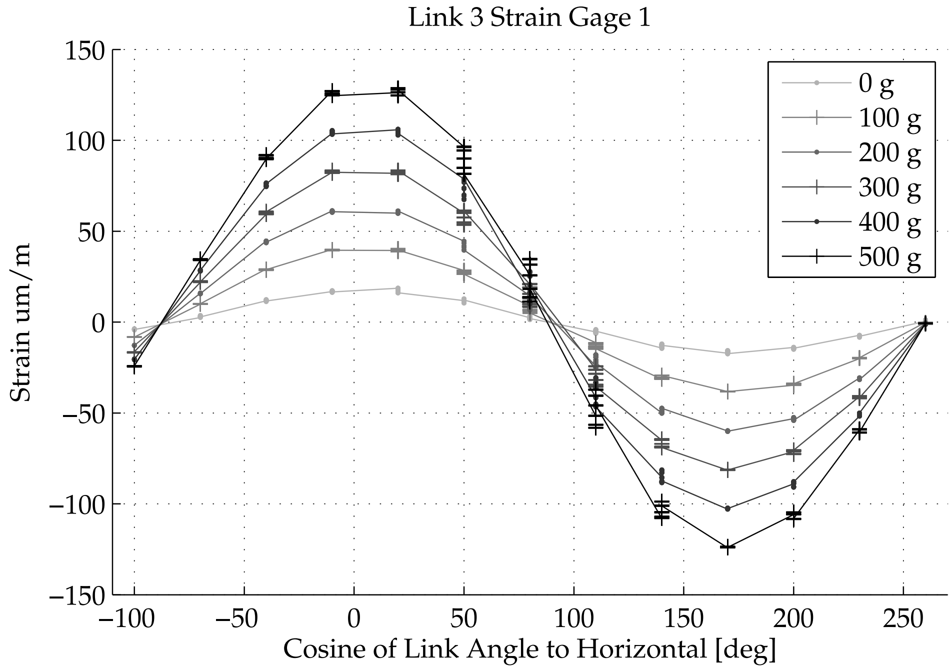


Figure 2.2.: Link 3 strain gage 1 strains from varying payloads.

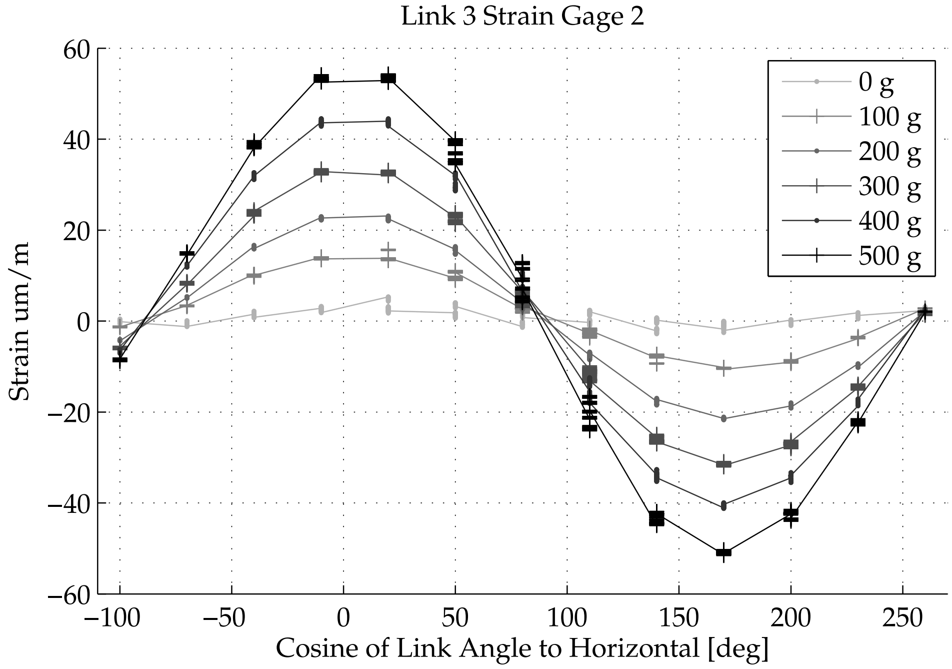


Figure 2.3.: Link 3 strain gage 2 strains from varying payloads.

In Figure 2.4 the same graph is transformed as the x-axis becomes the cosine of the link angle with respect to horizontal. As this removes the trigonometric effects seen in Figure 2.2 it is clear this can be represented by a cosine. The resulting data can now be seen to have a clear linear relationship between strain and payload.

From link 3 the component for the internal bending moment of the applied perpendicular tip force components in Equation 2.3.2 can be found in Equation 2.4.1. This

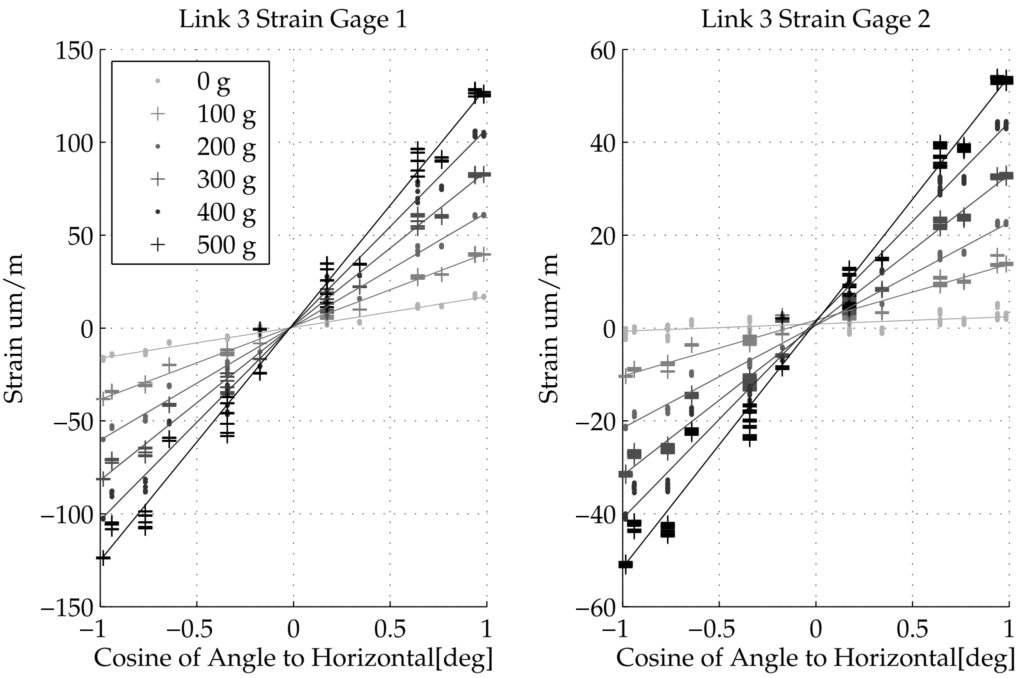


Figure 2.4.: Link 3 strains graphed against the cosine of the link angle with respect to horizontal. Linear regression lines of fit for each payload are also shown.

can then be converted to a component of strain of link 3 resulting from the applied force as in Equation 2.4.2.

$$M_f(x_{s3}) = f_y(l - x_{s3}) \tag{2.4.1}$$

$$\mathcal{E}_f(x_{s3}) = \frac{y_b}{EI} f_y(l - x_{s3}) \tag{2.4.2}$$

$$f_y = -m_4 g \cos\theta \tag{2.4.3}$$

By substituting Equation 2.4.3 into Equations 2.4.1 and 2.4.2, it can be seen that at any given joint angle configuration the internal bending moment due to the payload mass can be found. As a result, in a given angle configuration, the change from nominal link strain (with no mass) due to an added payload is proportional to the payload mass applied.

Further substitutions of the beam mass internal bending moment can produce an equation for a strain gage in link 3 with a fixed payload. This can be made linear in the parameters by putting it in the form of Equation 2.4.4. It should be noted that from this point on, equations for TUDOR will begin using the notation for the TUDOR robot as opposed to the notation used previously for a single beam model. Joint 2 and 3 angles q_2 and q_3 and link 3 length a_3 are substituted for beam angle θ and length l. A reference for symbols and notation used in each chapter of this thesis can be found in Appendix A.

$$\mathcal{E}_3 = \begin{bmatrix} 1 & cos(q_2 + q_3) \end{bmatrix} \begin{bmatrix} \frac{f_y y_b (a_3 - x_3)}{EI} \\ \frac{-g\, m_3 y_b (a_3 - x_3)^2}{2EIa_3} \end{bmatrix} = \begin{bmatrix} 1 & \cos(q_2 + q_3) \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} \tag{2.4.4a}$$

$$= \beta_1 + \beta_2 \cos(q_2 + q_3) \tag{2.4.4b}$$

Lumping the various fixed parameters together gives the added benefits of both simplifying the relationships between and reducing the number of parameters that need to be found. This is discussed further in Section 3.3.

To demonstrate the validity of this model, using the MERIt03 static dataset a linear regression was performed for each payload to find the lumped parameters β_1 and β_2 seen in Equation2.4.4b. The results for the linear regression can be seen in Table 2.1 and the lines of fit in Figure 2.4. Overall, the fit for both strain gauges on TUDOR's link 3 is very good for all payloads, with the exception of 0 grams on strain gage 32. This is the result of the lower signal to noise ratio of the second strain gage. With no payload, as the link becomes perpendicular to the ground, the magnitude of strain due to gravity becomes less than the magnitude of strain gage sensor noise.

	Strain Gage 31			Strain Gage 32		
Payload [g]	β_1	β_2	R^2	β_1	β_2	R^2
0	0.3371	16.5780	0.988	0.8763	1.5579	0.464
100	0.9446	39.3558	0.994	1.7129	12.0486	0.981
200	0.9005	61.1981	0.994	0.5759	22.1149	0.990
300	1.3640	83.2644	0.992	0.6615	32.3906	0.990
400	1.8959	105.3615	0.991	1.6466	42.6422	0.989
500	2.3647	127.4683	0.989	1.2852	52.9138	0.987

Table 2.1.: Linear regression models for estimating link 3 strain at various angles with a fixed payload with their R^2 coefficient of determination values.

2.5. Modeling Link 3 Strain for Varying Payloads

The models built in Section 2.4 work well, but only for a fixed payload. It would be beneficial to be able to model link 3 strain for any potentially varying tip force or payload.

If Equation 2.4.4a is examined, it can be seen that the perpendicular tip force f_y only appears in the first parameter term and is simply multiplied by the other parameters. It can therefore be moved algebraically into the regressor terms as in Equation 2.5.1.

$$\mathcal{E}(x_{s3}) = \begin{bmatrix} f_y & cos(q_2 + q_3) \end{bmatrix} \begin{bmatrix} \dfrac{y_b(a_3 - x_{s3})}{EI} \\ \dfrac{-\,g\, m_3 y_b (a_3 - x_{s3})^2}{2EIa_3} \end{bmatrix} \tag{2.5.1}$$

The perpendicular tip force f_y can be calculated as a component coming from a payload mass m_4 as well as a possible extra force applied by the environment F_{ly} as in Equation 2.5.2

$$f_y = -g\, m_4 \cos\left(q_2 + q_3\right) + F_{ly} \tag{2.5.2}$$

Equation 2.5.2 is then substituted into Equation 2.5.1 and rearranged. In addition, a constant term R_1 can be added to compensate for any offset or modeling error that may be present before performing a linear regression to produce Equation 2.5.3.

$$\mathcal{E}(x_{s3}) = \begin{bmatrix}1 & g\,m_4\cos(q_2+q_3) - F_{ly} & g\cos(q_2+q_3)\end{bmatrix} \begin{bmatrix} R_1 \\ -\dfrac{y_b(a_3 - x_{s3})}{EI} \\ -\dfrac{m_3 y_b (a_3 - x_{s3})^2}{2EIa_3} \end{bmatrix} \tag{2.5.3}$$

The linear regression can be re-run for all payloads, and now can be represented in Equation 2.5.4, with parameters R_1, R_2 and R_3 for a fixed strain gage position.

$$\mathcal{E}(x_{s3}) = R_1 + R_2(g\,m_4\cos(q_2+q_3) - F_l y) + R_3\,g\cos(q_2+q_3) \tag{2.5.4}$$

The linear regression (with a 60-40 training and test split) across the the MERIt03 dataset for all payloads yields very accurate estimates of strain. Results and parameters can be seen in Tables 2.2 and 2.3. Both strain gages show high R^2 values above 0.98 indicating that these models perform quite well for predicting link 3 strains while at rest.

	Training Performance			Test Performance		
Strain Gage	R^2	RMSE [μm/m]	NRMSE	R^2	RMSE [μm/m]	NRMSE
31	0.991	5.49	0.0217	0.990	5.64	0.0223
32	0.988	2.56	0.0242	0.987	2.58	0.0244

Table 2.2.: Performance of a linear regression for estimating link 3 strain in various link configurations using Equation 2.5.4. NRMSE is normalized by the range of the dataset strain measurements.

	Parameters		
Strain Gage	R_1	R_2	R_3
31	1.2087	22.5658	1.7145
32	1.0742	10.4553	0.1634

Table 2.3.: Parameters of the linear regression for estimating static link 3 strain in various link configurations.

2.6. Generalized Analytic Equations for Elastic Link Robots

Link 2 presents a much more complicated case than link 3. A tip force applied at the robot's effector applies only a force to link 3, but that same force can apply a force as well as a moment to link 2. This makes interpreting and predicting a strain readings from link 2 more complex. Additionally, the mass of link 3 can apply an additional force and a moment to the tip of link 2. If a payload at the effector is considered, this too can apply both a force and moment to the tip of link 2.

This more general case will be considered and link 2 equations will be formulated for both a perpendicularly applied effector force and an effector payload. This method could then be used to find analytic strain equations for other elastic link manipulators with alternate link configurations.

2.6.1. Relating Joint Torque To Strain

Joint torque is frequently found in equations related to robotics. The internal bending moment at distance $x = 0$ from the beam cantilever as pictured in Figure 2.1 is equivalent to joint torque τ. As strain gages are common on flexible link robots as they aid in tip position estimation, being able to translate between joint torque and strain referenced equations would allow the application of many existing tools currently used in robotics.

For example, the recursive Newton-Euler formulation can be used to build a static joint torque model for an n joint robot in generalized coordinates as in Equation 2.6.1, with vector of joint torques $\boldsymbol{\mathcal{T}}$, generalized joint coordinate $\boldsymbol{q}$ dependent gravity model $\mathbf{g}(\mathbf{q})$, Jacobian $\mathbf{J}(\mathbf{q})$ and generalized effector force $\boldsymbol{\mathcal{F}}$ (Spong et al. 2005).

$$\boldsymbol{\mathcal{T}} = \mathbf{g}(\mathbf{q}) + \mathbf{J}(\mathbf{q})^T \boldsymbol{\mathcal{F}} \tag{2.6.1}$$

The difference between joint torques $\boldsymbol{\mathcal{T}}$ (which is equivalent to $\mathbf{M}(\mathbf{x} = 0)$) and the internal bending moment at any point in the beam $\mathbf{M}(\mathbf{x})$ can be defined by the vector of differences $\boldsymbol{\Delta\tau}(\mathbf{x})$ as shown in Equation 2.6.2 where $\boldsymbol{x_s}$ is the vector of strain gage positions in link beams. Using Equation 2.2.1, the internal bending moments of the beams can then be converted to the equivalent strains in Equation 2.6.3.

$$\boldsymbol{M}(\boldsymbol{x_s}) = \boldsymbol{\mathcal{T}} - \boldsymbol{\Delta\tau}(\boldsymbol{x_s}) \tag{2.6.2}$$

$$\boldsymbol{\mathcal{E}}(\boldsymbol{x}) = \frac{y_b}{EI}(\boldsymbol{M}(\boldsymbol{x_s})) = \frac{y_b}{EI}(\boldsymbol{\mathcal{T}} - \boldsymbol{\Delta\tau}(\boldsymbol{x_s})) \tag{2.6.3}$$

The $\boldsymbol{\Delta\tau}(\boldsymbol{x_s})$ difference term vector is further split into components $\boldsymbol{\Delta\tau_g}(\boldsymbol{x_s})$, $\boldsymbol{\Delta\tau_f}(\boldsymbol{x_s})$, and $\boldsymbol{\Delta\tau_M}(\boldsymbol{x_s})$ in Equation 2.6.4 corresponding to the internal bending moments from the beams' weight due to gravity, the vector of perpendicular beam tip forces for each beam $\boldsymbol{f_y}$, and applied moments respectively (from Equation 2.1.1).

$$\boldsymbol{\Delta\tau}(\boldsymbol{x_s}) = \boldsymbol{\Delta\tau_g}(\boldsymbol{x_s}) + \boldsymbol{\Delta\tau_f}(\boldsymbol{x_s}) + \boldsymbol{\Delta\tau_M}(\boldsymbol{x_s}) \tag{2.6.4}$$

The individual difference terms are defined below in Equations 2.6.5, 2.6.6 and 2.6.7. For $\boldsymbol{\Delta\tau_g}$ in Equation 2.6.5, $\boldsymbol{\rho_b}$ is the vector of gravitational force per unit length of each link when horizontal, $\boldsymbol{l}$ is the vector of link lengths, and $\boldsymbol{\theta}$ is the vector of beam angles with respect to horizontal.

$$\boldsymbol{\Delta\tau_g}(\boldsymbol{x_s}) = \frac{1}{2}\,\mathrm{diag}(\boldsymbol{\rho_b})\,\mathrm{diag}(\boldsymbol{x_s})\,\mathrm{diag}(\cos\boldsymbol{\theta})(\boldsymbol{x_s} - 2\boldsymbol{l}) \tag{2.6.5}$$

For $\boldsymbol{\Delta\tau_f}$ in Equation 2.6.6, $\boldsymbol{f_y}$ is the vector of perpendicular tip forces applied to each link and $\boldsymbol{x}$ is the vector of strain gage positions on each link (zero being closest to the joint).

$$\boldsymbol{\Delta\tau_f}(\boldsymbol{x}) = \boldsymbol{f_y}\,\boldsymbol{x_s} \tag{2.6.6}$$

It is worth noting that $\boldsymbol{\tau_M}(\boldsymbol{x_s})$ in Equation 2.6.7 is zero due to an applied moment being constant in the internal bending moment of the beam as in Equation 2.1.11.

$$\boldsymbol{\Delta\tau}_M(\boldsymbol{x}) = \mathbf{0} \tag{2.6.7}$$

Together these equations will serve as a starting part for connecting equations in terms of joint torque with strain measurements.

The torque seen at a joint n resulting from the recursive Newton-Euler formulation will be referred to as $\mathcal{T}_n$. To build a strain model for link n, 2 difference terms must be applied so that the internal bending moment in link n can be calculated as in Equation 2.6.8.

$$M_n(x_{sn}) = \mathcal{T}_n - \Delta\tau_{f,n}(x_{sn}) - \Delta\tau_{g,n}(x_{sn}) \tag{2.6.8}$$

The difference term $\Delta\tau_{g,n}$ accounts for link n's distributed mass in Equation 2.6.9 with link mass m_n, link n strain gage position x_{sn}, link length a_n, and angle to horizontal θ_n. The difference term $\Delta\tau_{f,n}$ in Equation 2.6.10 accounts for perpendicular forces applied at the tip of link n .

$$\Delta\tau_{g,n}(x_{sn}) = \frac{m_n x_{sn}(x_{sn} - 2a_n)}{2a_n} \cos\theta \tag{2.6.9}$$

$$\Delta\tau_{f,n}(x_{sn}) = f_{y,n} x_{sn} \tag{2.6.10}$$

Where $f_{y,n}$ is the sum of the perpendicular force applied to the tip of link n by successive links and any applied external effector forces. To calculate this, a method to determine the effect a 6-dimensional force or wrench on the effector has on each subsequent link is needed.

2.6.2. Relating Wrenches to Joint Torques

The transpose of the robot Jacobian $\boldsymbol{J}(\boldsymbol{q})$ allows a generalized force $\boldsymbol{\mathcal{F}}$ (as seen before in Equation 2.1.4) applied to the effector to be transformed into the vector of joint torques required to balance it as in Equation 2.6.11. (Corke 2011). This is easy to use when joint torque can be directly sensed (Craig 2004, p. 158). However, when using only strain gages, joint torques cannot be directly measured as various forces acting on the beam distribute differently along the length.

$$\boldsymbol{\mathcal{T}} = \boldsymbol{J}(\boldsymbol{q})^T \boldsymbol{\mathcal{F}} \tag{2.6.11}$$

A force applied perpendicular to the effector will result in an equivalent moment and force applied to each link. The moment and force resulting from a given force depend on the link of interest and the current pose of the robot. For the link closest to the effector, the models presented previously in Section 2.5 for TUDOR's link 3 can be used. For other links a means to determine the equivalent moment and perpendicular force applied to each link is needed to be able to use the beam models developed so far.

2.6.3. Force Moment Transform

The force-moment transform allows for the transformation of a 6-dimensional generalized static force $\boldsymbol{\mathcal{F}}$ in one frame to its equivalent in another frame. Simply rewriting the components of the force in the other frames' coordinates would not be sufficient as the point of application of the force would have to change for the work to be equivalent. The force-moment transform allows this mapping while keeping the point of application of the force the same (Murray et al. 1994).

The equations for the mapping of a generalized force or wrench $\boldsymbol{\mathcal{F}}$ in frame A to frame B can be seen in Equation 2.6.12, with vector offset ${}^{B}\boldsymbol{P}_{AORG}$ between frames A and B, 3 component force vector ${}^{k}\boldsymbol{f}$ in frame k, 3 component moment vector ${}^{k}\boldsymbol{n}$ in frame k, and rotation transform between frames A and B ${}^{B}_{A}\boldsymbol{R}$ (Craig 2004, p. 158) (Murray et al. 1994, p. 62).

$$\begin{bmatrix} {}^{B}\boldsymbol{f} \\ {}^{B}\boldsymbol{n} \end{bmatrix} = \begin{bmatrix} {}^{B}_{A}\boldsymbol{R} & \boldsymbol{0} \\ {}^{B}\boldsymbol{P}_{AORG} \times {}^{B}_{A}\boldsymbol{R} & {}^{B}_{A}\boldsymbol{R} \end{bmatrix} \begin{bmatrix} {}^{A}\boldsymbol{f} \\ {}^{A}\boldsymbol{n} \end{bmatrix} \tag{2.6.12a}$$

$${}^{B}\boldsymbol{\mathcal{F}} = {}^{B}_{A}\boldsymbol{T}_{f}\, {}^{A}\boldsymbol{\mathcal{F}} \tag{2.6.12b}$$

2.6.4. Method for Determining Forces and Moments Seen by a Link

To convert a standard torque-referenced robot model to a strain referenced one the difference terms found in Section 2.6.3 must be computed for each link of interest. To compute the difference terms, the force perpendicular to the link tip, as well as the link mass and length must be known. As the basic link properties are known, only the tip force must be found. This is done by using the force-moment transform equation in Section 2.6.3. The generalized effector force is transformed and the equivalent at each link tip is found.

To shorten notation, the quantity $m_{s,n}$ is defined in Equation 2.6.13, for a robot with number of links p, and link mass m_n for link n. For a given link number n, this gives the sum of the mass of any following links in the manipulator chain. For example for TUDOR, $m_{s,2} = m_3$. It should also be noted that for the last link, this quantity will always be zero, so for TUDOR's link 3 $m_{s,3} = 0$.

$$m_{s,n} = \sum_{k=n}^{p-1} m_{k+1} \tag{2.6.13}$$

The rows of the robot Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$ are notated as per Equation 2.6.14 where ${}^{n}\boldsymbol{j}_k(\boldsymbol{q})$ is the kth row of the Jacobian with respect to link n.

$${}^{n}\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} {}^{n}\boldsymbol{j}_1(\boldsymbol{q}) \\ \vdots \\ {}^{n}\boldsymbol{j}_k(\boldsymbol{q}) \end{bmatrix} \tag{2.6.14}$$

The unit vector $\hat{\mathbf{j}}^{*}_{\mathbf{n}}$ is defined as the unit vector in the y-axis for link n, augmented with 3 additional zero elements to make a 6 element vector as per Equation 2.6.15. The subscript 0 on vector $\hat{\mathbf{j}}^{*}_{\mathbf{0}}$ refers to the global coordinate frame.

$$\hat{\mathbf{j}}_{\mathbf{n}}^{*} = \begin{bmatrix} 0 & \hat{\boldsymbol{j}}_{\boldsymbol{n}} & 0 & 0 & 0 & 0 \end{bmatrix}^{T} \tag{2.6.15}$$

The equation relating a static robot gravity torque model $\boldsymbol{g}(\boldsymbol{q})$ and internal bending moment for all links of an n link robot can be found in Equation 2.6.16. Where ${}^{B}_{A}\boldsymbol{R}$ is the rotation transform from frame A to B, x_n is the strain gage position on link n, and $\cdot$ is the dot product, m_l is the payload mass, ${}^{n}_{E}\boldsymbol{T}$ is the force-moment transform from Section 2.6.3 from effector frame E to the frame of link n. The difference terms for link mass $\Delta\tau_{g,n}$ can be calculated using the Equation 2.6.9, for each link n.

$$\boldsymbol{M}(\boldsymbol{x_s}) = \boldsymbol{\mathcal{T}} - \boldsymbol{\Delta\tau}(\boldsymbol{x_s}) \tag{2.6.16a}$$

$$\boldsymbol{M}(\boldsymbol{x_s}) = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^{T}\boldsymbol{\mathcal{F}} - \boldsymbol{\Delta\tau}_{f}(\boldsymbol{x_s}) - \boldsymbol{\Delta\tau}_{g}(\boldsymbol{x_s}) \tag{2.6.16b}$$

$$\begin{bmatrix} M_1(x_{s1}) \\ \vdots \\ M_n(x_{sn}) \end{bmatrix} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^{T}\boldsymbol{\mathcal{F}} - \begin{bmatrix} x_{s1}(({}^{1}\boldsymbol{j}_{1}^{T}\ {}^{1}_{E}\boldsymbol{T}\,\boldsymbol{\mathcal{F}} + {}^{1}_{0}\boldsymbol{R}(-(m_l + m_{s,1})\,g\,\hat{\mathbf{j}}_{\mathbf{0}}^{*})) \cdot \hat{\mathbf{j}}_{\mathbf{1}}^{*}) \\ \vdots \\ x_{sn}(({}^{n}\boldsymbol{j}_{n}^{T}\ {}^{n}_{E}\boldsymbol{T}\,\boldsymbol{\mathcal{F}} + {}^{n}_{0}\boldsymbol{R}(-(m_l + m_{s,n})\,g\,\hat{\mathbf{j}}_{\mathbf{0}}^{*})) \cdot \hat{\mathbf{j}}_{\mathbf{n}}^{*}) \end{bmatrix} + \dots$$
$$- \begin{bmatrix} \Delta\tau_{g,n} \\ \vdots \\ \Delta\tau_{g,1} \end{bmatrix} \tag{2.6.16c}$$

This equation shows that the perpendicular force needed for the difference term can be calculated as a result of a force-moment transform of the effector force, as well as the force from the mass of following links. The force-moment transformed effector force is then converted to a joint torque using the row of the Jacobian for that link's joint (with the Jacobian with respect to that link's frame).

2.7. Formulating Internal Bending Moments of TUDOR

Equation 2.6.16 can be applied to calculate the internal bending moments of link 2 and 3 of TUDOR. Effector tip forces F_{lx} and F_{ly} as well as a payload mass m_4 are considered and the generalized effector force is represented as in Equation 2.7.1 where θ_n is defined as the link n's angle with respect to horizontal.

$$\boldsymbol{\mathcal{F}} = \begin{bmatrix} f_x & f_y & 0 & 0 & 0 & 0 \end{bmatrix}^{T} \tag{2.7.1a}$$

$$f_x = -g\,m_4 \sin\theta_n + F_{lx} \tag{2.7.1b}$$

$$f_y = -g\,m_4 \cos\theta_n + F_{ly} \tag{2.7.1c}$$

The difference terms $\Delta\tau_{f,n}$ for links n are calculated using the method described in Section 2.6.4 and can be found in Equation 2.7.2 with $q_2 + q_3$ substituted for θ_n for TUDOR.

$$\Delta\tau_{f,2}(x_{s2}) = x_{s2}(F_{ly}\cos q_3 + F_{lx}\sin q_3 - g\,m_3\cos q_2 - g\,m_l\cos q_2) \tag{2.7.2a}$$

$$\Delta\tau_{f,3}(x_{s3}) = x_{s3}(F_{ly} - g\,m_4\cos{(q_2 + q_3)}) \tag{2.7.2b}$$

The difference terms $\Delta\tau_{g,n}$ are calculated using Equation 2.6.9 and found below as in Equation 2.7.3. For a reference of symbols used, see Appendix A.

$$\Delta\tau_{g,2}(x_{s2}) = -\frac{g\, m_2 x_{s2} \cos q2(2a_2 - x_{s2})}{2a_2} \tag{2.7.3a}$$

$$\Delta\tau_{g,3}(x_{s3}) = -\frac{g\, m_3 x_{s3} \cos(q_2 + q_3)(2a_3 - x_{s3})}{2a_3} \tag{2.7.3b}$$

The gravity model $\boldsymbol{g}(\boldsymbol{q})$ for the links 2 and 3 is built using the recursive Newton-Euler model and is found in Equation 2.7.4 Craig 2004. This gravity model includes the effects of payload m_4 as an additional link with zero length to aid in equation generation.

$$\begin{aligned} g_2(\boldsymbol{q}) = {} & \frac{a_3\, g\, m_3 \sin q_2 \sin q_3}{2} - \frac{a_2\, g\, m_2 \cos q_2}{2} - a_2\, g\, m_3 \cos q_2 - a_2\, g\, m_4 \cos q_2 \\ & - \frac{a_3\, g\, m_3 \cos q_2 \cos q_3}{2} - a_3\, g\, m_4 \cos q_2 \cos q_3 + a_3\, g\, m_4 \sin q_2 \sin q3 \end{aligned} \tag{2.7.4a}$$

$$g_3(\boldsymbol{q}) = -\frac{a_3\, g\, m_3 \cos(q_2 + q_3)}{2} - a_3\, g\, m_4 \cos(q_2 + q_3) \tag{2.7.4b}$$

The Jacobian $\boldsymbol{J}(\boldsymbol{q})$ in Equation 2.7.5 below can be used to calculate the torques resulting from the applied effector force. The payload mass m_4 is omitted as this is already captured in the gravity model $\boldsymbol{g}(\boldsymbol{q})$ in Equation 2.7.4.

$$\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} F_{ly}(a_3 + a_2 \cos q_3) + F_{lx} a_2 \sin q_3 \\ F_{ly} a_3 \end{bmatrix} \tag{2.7.5}$$

These equations can be assembled to build the standard rigid-link torque-referenced model for TUDOR's 2nd and 3rd joints as in Equation 2.7.6.

$$\begin{aligned} \mathcal{T}_2(\boldsymbol{q}) = {} & F_{ly}(a_3 + a_2 \cos q_3) + F_{lx} a_2 \sin q_3 - \frac{a_3\, g\, m_3 \cos(q_2 + q_3)}{2} - a_3\, g\, m_4 \cos(q_2 + q_3) \\ & - \frac{a_2\, g\, m_2 \cos q_2}{2} - a_2\, g\, m_3 \cos q_2 - a_2\, g\, m_4 \cos q_2 \end{aligned} \tag{2.7.6a}$$

$$\mathcal{T}_2(\boldsymbol{q}) = -\frac{a_3(g\, m_3 \cos(q_2 + q_3) - 2F_{ly} + 2\, g\, m_4 \cos(q_2 + q_3))}{2} \tag{2.7.6b}$$

These equations can also be assembled to form the equations for the internal bending moment of TUDOR's links 2 and 3 as in Equation 2.7.7. For brevity, abbreviations for the sine and cosine functions of the generalized link angles $\boldsymbol{q}$ have been used such that $\mathcal{C}_{23}$ represents $\cos(q_2 + q_3)$, and $\mathcal{S}_2$ represents $\sin q_2$.

$$\begin{aligned} M_2(x_{s2}) = {} & F_{ly}(a_3 + a_2\mathcal{C}_3) - x_{s2}(F_{ly}\mathcal{C}_3 + F_{lx}\mathcal{S}_3 - g\, m_3\mathcal{C}_2 - g\, m_4\mathcal{C}_2) + F_{lx} a_2 \mathcal{S}_3 - \frac{a_3\, g\, m_3 \mathcal{C}_{23}}{2} \\ & - a_3\, g\, m_4 \mathcal{C}_{23} - \frac{a_2\, g\, m_2 \mathcal{C}_2}{2} - a_2\, g\, m_3 \mathcal{C}_2 - a_2\, g\, m_4 \mathcal{C}_2 + \frac{g\, m_2 x_{s2} \mathcal{C}_2 (2a_2 - x_{s2})}{2a_2} \end{aligned} \tag{2.7.7a}$$

$$M_3(x_{s3}) = \frac{(a_3 - x_{s3})(2F_{ly} a_3 - a_3\, g\, m_3 \mathcal{C}_{23} - 2a_3\, g\, m_4 \mathcal{C}_{23} + g\, m_3 x_{s3} \mathcal{C}_{23})}{2a_3} \tag{2.7.7b}$$

It should be noted that when the strain positions x_{sn} are zero the internal bending moment equations will be the same as the standard rigid-link model equations. Performing this check on Equations 2.7.6 and 2.7.7 shows that this is true for the calculated equations for TUDOR.

The strain model can now be built from the internal bending moment using Equation 2.2.1. As seen previously in Section 2.4, these equations can be made linear in the parameters. While lumping the parameters, it can be seen that some parameter terms will be quite similar and that keeping the gravitational constant g out of the parameter terms will allow some to be lumped further. This reduces the number of parameters for link 2 from 6 to 4, and for link 3 from 6 to 2. The results can be seen in Equation 2.7.8 with additional offset parameters R_1 added.

$$\mathcal{E}_2(x_{s2}) = \begin{bmatrix} 1 \\ g\, m_4\mathcal{C}_{23} - F_{ly} \\ g\,\mathcal{C}_{23} \\ g\, m_4\mathcal{C}_2 - F_{lx}\mathcal{S}_3 - F_{ly}\mathcal{C}_3 \\ g\,\mathcal{C}_2 \end{bmatrix}^T \begin{bmatrix} R_1 \\ -\dfrac{a_3 m_3 y_b}{2EI} \\ -\dfrac{a_3 m_3 y_b}{2EI} \\ -\dfrac{y_b(a_2 - x_{s2})}{EI} \\ -\dfrac{y_b(a_2 - x_{s2})(a_2m_2 + 2a_2m_3 - m_2x_{s2})}{2EIa_2} \end{bmatrix} \tag{2.7.8a}$$

$$\mathcal{E}_3(x_{s3}) = \begin{bmatrix} 1 \\ g\, m_4\mathcal{C}_{23} - F_{ly} \\ g\,\mathcal{C}_{23} \end{bmatrix}^T \begin{bmatrix} R_1 \\ -\frac{y_b(a_3-x_{s3})}{EI} \\ -\frac{m_3y_b(a_3-x_{s3})^2}{2EIa_3} \end{bmatrix} \tag{2.7.8b}$$

This additional lumping of parameters has the added benefit of moving terms containing F_{ly} or F_{lx} into terms containing m_4. This will later prove highly beneficial later as it will allow any models trained on payload data to also be used in the presence of effector forces. Reliably applying perpendicular tip forces in a reproducible fashion is much more difficult than applying fixed payloads; being able to use payloads to train for tip forces is greatly beneficial.

It should be noted that following this method, the analytic equations for link 3 found in equation 2.5.3 have been reproduced identically in Equation 2.7.8b. This means that the results of the linear regression for link 3 found in Section 2.5 will be identical as well.

2.8. TUDOR Link 2 Linear Regression Results

The analytic equation derived above (Equation 2.7.8a), shows that it can expected that the strain from a payload and tip force will be a linear combination of $g\, m_4 \cos(q_2 + q_3) - F_{ly}$, $g \cos(q_2 + q_3)$, $g\, m_4 \cos q_2 - F_{lx} \sin q_3 - F_{ly} \cos q_3$, and $g \cos q_2$. As done previously with the link 3 linear regression in Section 2.5, a constant offset parameter is added as well.

This can be demonstrated by performing a linear regression using the data from the MERIt03 dataset. The effector tip forces F_{lx} and F_{ly} are set to zero and the payload m_4 is set. A 60-40 training and test split was used with the MERIt03 static dataset. The performance and parameters of the linear regression can be found in Tables 2.4 and 2.5 for each strain gage.

	Training Performance			Test Performance		
Strain Gage	R^2	RMSE [μm/m]	NRMSE	R^2	RMSE [μm/m]	NRMSE
21	0.997	17.29	0.0152	0.997	17.69	0.0155
22	0.997	9.25	0.0138	0.997	9.31	0.0139

Table 2.4.: Performance of linear regression for estimating link 2 strain in various link configurations. The following equation $\mathcal{E}_2 = R_1 + R_2\, g\, m_4 \cos(q_2 + q_3) + R_3\, g \cos(q_2 + q_3) + R_4\, g\, m_4 \cos q_2 + R_5\, g \cos q_2$, which is linear in the parameters R_i is used for the regression (and tip forces F_{lx} and F_{ly} were neglected for this dataset). NRMSE is normalized by the range of the dataset strain measurements.

	Parameters				
Strain Gage	R_1	R_2	R_3	R_4	R_5
21	-19.3697	25.8985	2.6468	27.2239	31.4224
22	-1.4358	25.7291	2.4479	12.4543	13.8884

Table 2.5.: Parameters from the linear regression for estimating static link 2 strain in various link configurations from strain gages 21 and 22.

From the error metrics for link 2 in Table 2.4 and link 3 in Table 2.1, it is clear that the method for building a strain model from a standard torque model and fitting the lumped parameters with a linear regression has worked very well. This will allow accurate predictions of strain while the robot is at rest to be made and could be easily adapted to other elastic link robot manipulators with differing joint configurations.

3
Dynamic Link Strain Prediction

This chapter develops models for strain of elastic link robot manipulators in motion. It begins by applying the static equations to dynamic data to demonstrate the need for additional dynamic modeling. A dynamic model is built, applied and performance is demonstrated against dynamic data. The dynamic model is further improved with the application of Kalman filtering for the joint angle derivatives.

The use of the rigid link torque-referenced manipulator dynamics equations in this chapter can be justified by the assumptions made in Section 1.3.1. When the elastic link dynamics are split into slow and fast subsystems, and the fast subsystem is sufficiently damped, the slow system will be sufficiently similar to the rigid link torque-referenced robot dynamics. This chapter will also experimentally demonstrate the validity of this approach by providing sufficiently accurate strain predictions for a flexible link manipulator in motion.

3.1. Dynamic Strain Estimation Using Static Equations

In Chapter 2, static strain estimation equations were built, and then trained using linear regression against the MERIt03 dataset. This performed well, but as the MERIt03 dataset only contains data from the robot at rest, it does not give an idea of how these same equations would perform while in motion. To test this, the performance of the static equations and parameters from Chapter 2 against the MERIt02 dataset is summarized in Table 3.1. As described in Section 1.4, MERIt02 contains runs of data with TUDOR in motion with various payloads.

Strain Gage	R^2	RMSE [μm/m]	NRMSE
21	0.948	74.57	0.0347
22	0.852	69.33	0.0595
31	0.883	19.00	0.0335
32	0.848	8.83	0.0358

Table 3.1.: Performance of the static strain equations and parameters against dynamic data from the MERIt02 dataset. NRMSE is normalized by the range of the dataset strain measurements.

The performance of the static equations and parameters is considerably worse against dynamic data than the static data (results of which can be found in Tables 2.4 and 2.2). Performance for strain estimates in dynamic conditions (from MERIt02) compared to measured strains with a 300 gram payload can be seen in Figures 3.1 and 3.2.

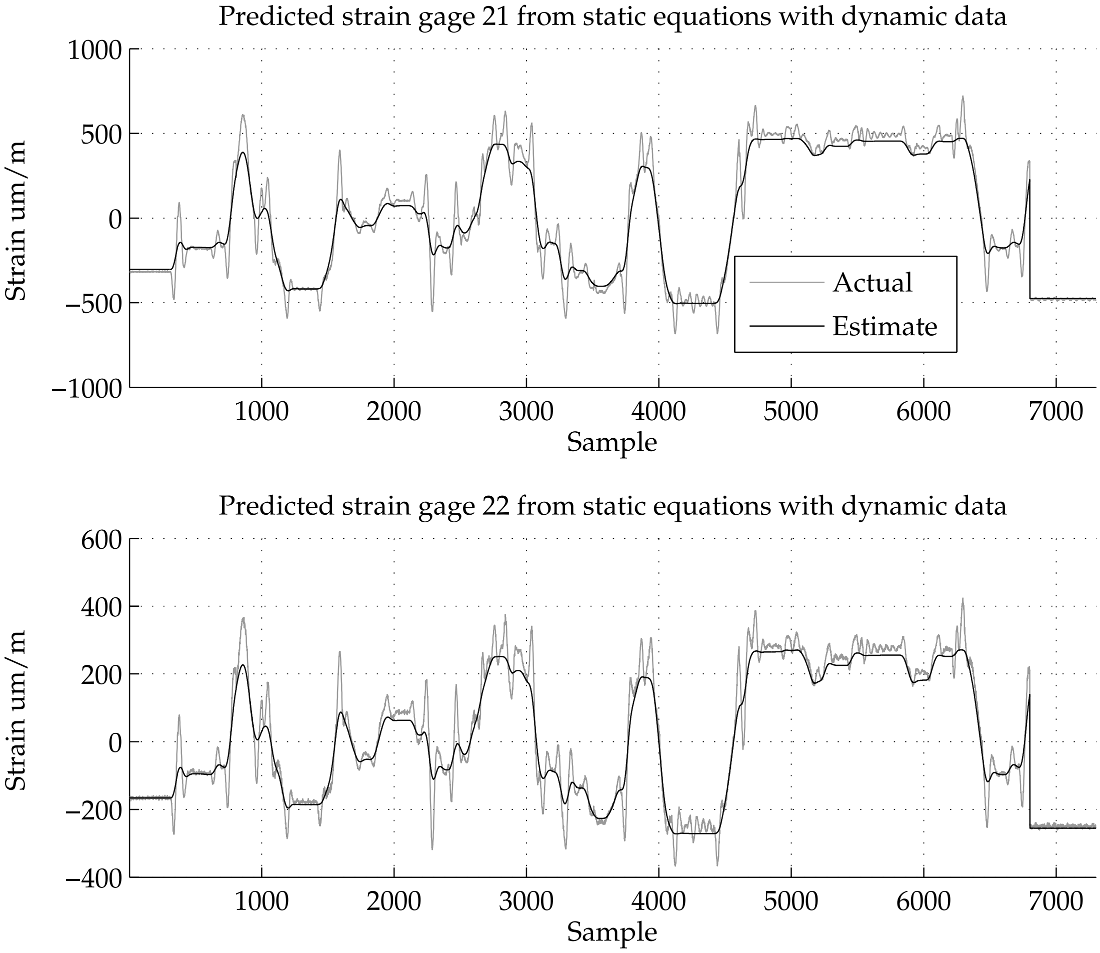
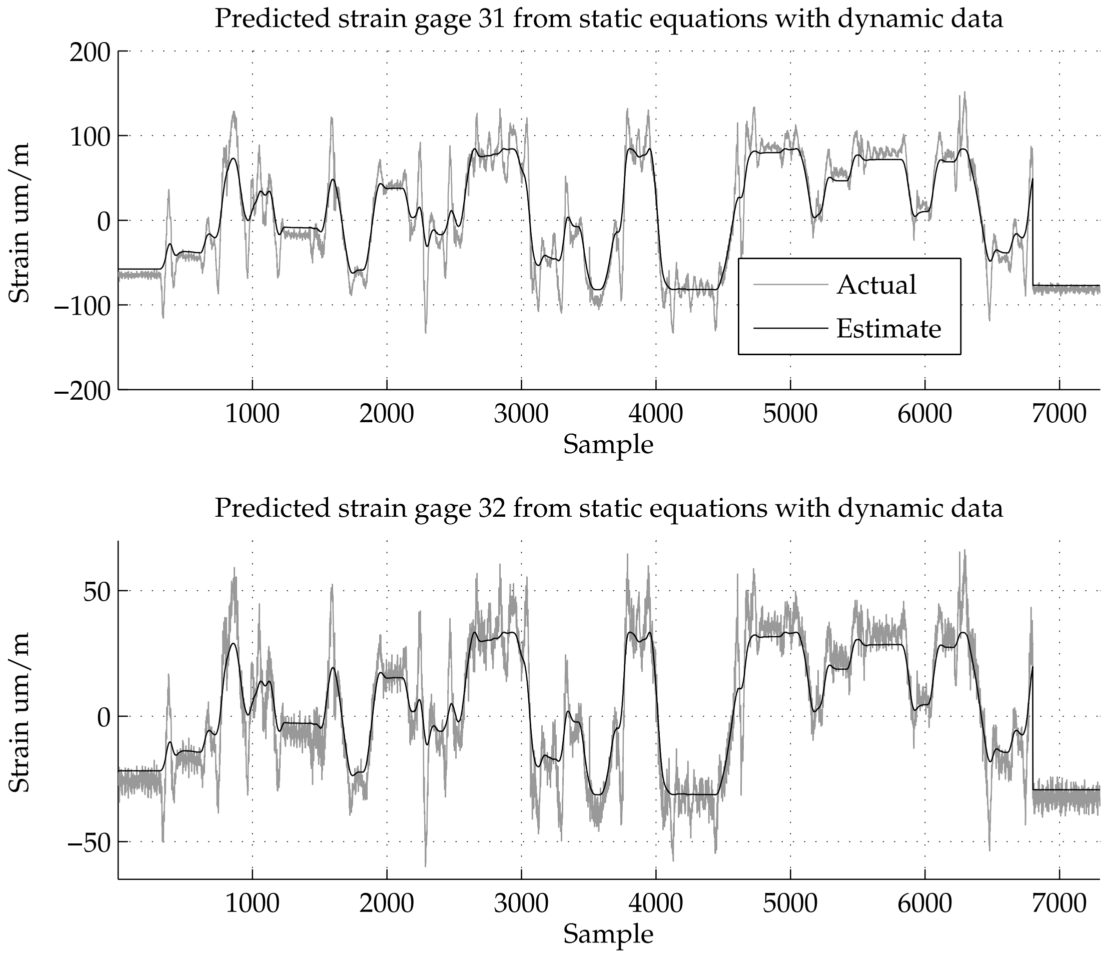


Figure 3.1.: Actual strain from dynamic data with 300g payload, and predicted values using static equations trained on static data for strain gages 21 and 22.

When comparing the static MERIt03 and dynamic MERIt02 dataset, an offset between data can sometimes be seen for identical payloads and joint angles when at rest. This can be linked to the method used for strain calibration offsets when data is collected. Noise can sometimes cause a different offset to be applied during calibration. This appears more frequently in strain gage 22.

Table 3.1 clearly shows that performance for dynamic cases is substantially reduced. Any attempt to use the static equations to predict payload mass would likely be very problematic, as it will be later shown in Chapters 4 and 5 that small errors in strain predictions can lead to large errors in mass and force estimates.


Predicted strain gage 31 from static equations with dynamic data

Predicted strain gage 32 from static equations with dynamic data

Figure 3.2.: Actual strain from dynamic data with 300g payload, and predicted values using static equations trained on static data for strain gages 31 and 32.

3.2. Training Static Equations With Dynamic Data

While the static dataset has a limited number of joint angles with very low noise, the dynamic dataset offers much more data with more varied joint angles. Retraining the static equations on the dynamic dataset could possibly provide a better fit for dynamic situations.

The static model linear regression is redone using the dynamic MERIt02 dataset with a 60-40 training and test split. The results of this can be seen in Tables 3.2.

Additionally, this model is then re-tested against the static dataset to determine how the performance in static situations has been effected by training against dynamic data. The results of this can be seen in Table 3.3.

The performance against the dynamic data improves some for strain gages 21 and 22, but only marginally for 31 and 32. The performance against the static dataset has deteriorated, with RMSE increasing for all strain gages. The improvement against dynamic data is not significant enough to warrant the loss of static performance. Training a static model on dynamic data is not a viable option for predicting strain in dynamic situations.

	Training Performance			Test Performance		
Strain Gage	R^2	RMSE [μm/m]	NRMSE	R^2	RMSE [μm/m]	NRMSE
21	0.957	67.46	0.0314	0.957	67.70	0.0318
22	0.891	59.47	0.0520	0.890	59.71	0.0512
31	0.888	18.59	0.0330	0.888	18.61	0.0340
32	0.860	8.47	0.0349	0.859	8.48	0.0344

Table 3.2.: Static strain prediction linear regression trained and tested on dynamic data. NRMSE is normalized by the range of the dataset strain measurements.

	Static Performance			
Strain Gage	R^2	RMSE [μm/m]	NRMSE	RMSE Increase
21	0.985	39.72	0.0348	2.24×
22	0.947	39.73	0.0592	4.27×
31	0.985	6.93	0.0274	1.23×
32	0.975	3.62	0.0341	1.40×

Table 3.3.: Static strain prediction linear regression trained on dynamic data and tested against the static dataset. The increase in RMSE from the original static test is also included to illustrate the increase in error. NRMSE is normalized by the range of the dataset strain measurements.

3.3. Adding Dynamic Components

In order to capture the dynamic behavior of the robot more accurately, additional terms will need to be added to the equations used for linear regression. The typical state-space form of torque-referenced robot manipulator dynamics is found in Equation 3.3.1 (Lewis et al. 2003).

$$\boldsymbol{\mathcal{T}} = \boldsymbol{I}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\mathcal{F}} \tag{3.3.1}$$

The dynamics of a robot arm are typically built using either the Euler-Lagrange formulation or the recursive Newton-Euler formulation. Both formulations result in the same equations, however for the Newton-Euler method can be applied recursively, making computer implementation simpler. Both formulations are typically put into the form seen above in Equation 3.3.1. The matrix of equations $\boldsymbol{I}(\boldsymbol{q})$ captures effects related to inertia and is often termed the inertia matrix. The matrix of equations $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ captures centrifugal and Coriolis effects and is often termed the Coriolis matrix (Spong et al. 2005). As seen before in Equation 2.6.1, $\boldsymbol{g}(\boldsymbol{q})$ represents the gravity model of the robot, and $\boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\mathcal{F}}$ the torques from an applied static generalized effector force or wrench.

The missing components in previous static equations are the Inertia and Coriolis terms. Luckily these terms can be viewed as applying moments to the link beams. As seen before in Section 2.6.1, this means that no difference terms will need to be added to convert Equation 3.3.1 from a torque-referenced model to a strain referenced model for use with elastic link robots. The Inertia and Coriolis terms need only be computed and added to the existing models from Chapter 2. The general form for strain-referenced robot dynamics equations built from standard torque-referenced robot dynamics can be seen below in Equations 3.3.2, with vector of strain gage positions $\boldsymbol{x_s}$, and vector of generalized robot coordinates $\boldsymbol{q}$.

$$\boldsymbol{\mathcal{E}}(\boldsymbol{x_s}) = \frac{y_b}{EI}\boldsymbol{M}(\boldsymbol{x_s}) = \frac{y_b}{EI}(\boldsymbol{\mathcal{T}} - \boldsymbol{\Delta\tau}(\boldsymbol{x_s})) \tag{3.3.2a}$$

$$\boldsymbol{\mathcal{E}}(\boldsymbol{x_s}) = \frac{y_b}{EI}(\boldsymbol{I}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\mathcal{F}} - \boldsymbol{\Delta\tau}(\boldsymbol{x_s})) \tag{3.3.2b}$$

The Coriolis and inertia matrices for the TUDOR robot are generated using the recursive Newton-Euler algorithm as implemented in the Robotics Toolbox for MATLAB (Corke 2011). In addition, the friction and motor inertia terms that were neglected in the static gravitational models due to a lack of motion are now included. It is worth noting that the current version of the robotics toolbox uses a single static friction model for all joints of a serial link robot, so only one friction coefficient is used in these models.

The dynamic equations for the link 2 and 3 strain of TUDOR contain a total of 37 and 27 constant parameters (listed in Equation 3.3.3 and 3.3.4). This is a dramatic increase from 4 and 2 parameters in the static strain equations of link 2 and 3. The robot is physically limited by its kinematics, making the identification of each of these parameters not just difficult, but in fact impossible. Parameters defining the robot dynamics for torque-referenced models are often grouped as fully identifiable, identifiable only in linear combination, and completely unidentifiable (Verdonck 2004). Those parameters that are completely unidentifiable are ones which do not effect the torques (and hence the strains) of the model and can be safely ignored (Atkeson et al. 1986). These same grouping can be applied to the parameters in the strain referenced robot models.

$$\boldsymbol{\theta}_{all2} = \begin{array}{lllllllllll} \big[E & I & I_{xx2} & I_{xx3} & I_{xx4} & I_{xy2} & I_{xy3} & I_{xy4} & I_{xz2} & I_{xz3} & I_{xz4} \\ I_{yy2} & I_{yy3} & I_{yy4} & I_{yz2} & I_{yz3} & I_{yz4} & I_{zz2} & I_{zz3} & I_{zz4} & J_{m2} & T_c \\ a_2 & a_3 & g_{R1} & g_{R2} & g_{R3} & m_2 & m3 & r_{x2} & r_{x3} & r_{y2} & r_{y3} \\ r_{z2} & r_{z3} & x_2 & y_b\big]^T \end{array} \tag{3.3.3}$$

$$\boldsymbol{\theta}_{all3} = \begin{array}{lllllllllll} \big[E & I & I_{xx3} & I_{xx4} & I_{xy3} & I_{xy4} & I_{xz3} & I_{xz4} & I_{yy3} & I_{yy4} & I_{yz3} \\ I_{yz4} & I_{zz3} & I_{zz4} & J_{m3} & T_c & a_2 & a_3 & g_{R1} & g_{R2} & g_{R3} & m_3 \\ r_{x3} & r_{y3} & r_{z3} & x_3 & y_b\big]^T \end{array} \tag{3.3.4}$$

Following the same procedure as with the static equations in Section 2.5, the robot strain equations can be made linear in the parameters. This is a technique employed frequently with the standard torque-referenced robot dynamics (Swevers et al. 2007). In Equation 3.3.5 this can be seen, where the torque dynamics model is represented as a regressor matrix $\boldsymbol{\Phi}_\tau$ and a parameter vector $\boldsymbol{\theta}_{\tau,linear}$. As the strain referenced model is mathematically very similar, these same tools can be easily applied as in Equation 3.3.6.

$$\begin{aligned} \boldsymbol{\mathcal{T}} &= \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\mathcal{F}} \\ &= \boldsymbol{\Phi}_\tau(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\theta}_{\tau,linear} \end{aligned} \tag{3.3.5}$$

$$\begin{aligned} \boldsymbol{\mathcal{E}}(\boldsymbol{x}) &= \frac{y_b}{EI}(\boldsymbol{\mathcal{T}} - \boldsymbol{\Delta\tau}(\boldsymbol{x})) \\ &= \boldsymbol{\Phi}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\theta}_{linear} \end{aligned} \tag{3.3.6}$$

The procedure used is the same as in (Malzahn et al. 2014a) in which the generalized link coordinates and trigonometric functions of link coordinates are factored from the equations to produce a minimal set of identifiable parameters for each link. As before in Section 2.5, the applied effector force, payload mass and gravity constant is factored out in addition to the joint angles and joint angle derivatives. This allows the same model to be used for varying effector forces and masses. The model can also be trained on mass data and then used to estimate effector forces.

Following this process, the parameter vector $\boldsymbol{\theta_{linear}}$ for links 2 and 3 can be found. The number of lumped linear parameters is greatly reduced from the original amount in equations 3.3.3 and 3.3.4. For link 2 37 parameters are reduced to 24 and for link 3 27 parameters are reduced to 15. The contents of the regressor matrix $\boldsymbol{\Phi}$ for links 2 and 3 can be seen in Tables 3.4 and 3.5.

1	1
2	$\dot{q_1}^2(\cos(2q_2) - 3\cos(2q_3) + \cos(2q_2 + 2q_3) - 3)$
3	$m_4(8\ddot{q_3} + \dot{q_1}^2\sin(2q_2) - 3\dot{q_1}^2\sin(2q_3) + \dot{q_1}^2\sin(2q_2 + 2q_3))$
4	$\dot{q_1}^2(\sin(2q_2) - 3\sin(2q_3) + \sin(2q_2 + 2q_3))$
5	$\dot{q_1}^2\cos(2q_2 + q_3) - 4\dot{q_3}^2\cos q_3 - 3\dot{q_1}^2\cos q_3 - 2\ddot{q_2}\sin q_3 - 4\ddot{q_3}\sin q_3 - 2\dot{q_2}\dot{q_3}\cos q_3$
6	$-m_4(3\dot{q_1}^2\sin q_3 + 4\dot{q_3}^2\sin q_3 - \dot{q_1}^2\sin(2q_2 + q_3) - 2\ddot{q_2}\cos q_3 - 4\ddot{q_3}\cos q_3 + 2\dot{q_2}\dot{q_3}\sin q_3)$
7	$\dot{q_1}^2\sin(2q_2 + q_3) - 4\dot{q_3}^2\sin q_3 - 3\dot{q_1}^2\sin q_3 + 2\ddot{q_2}\cos q_3 + 4\ddot{q_3}\cos q_3 - 2\dot{q_2}\dot{q_3}\sin q_3$
8	$\dot{q_1}^2(\cos q_2^2 - 2)$
9	$\ddot{q_1}\cos(q_2 + q_3)$
10	$gm_4\cos(q_2 + q_3) - F_{ly}$
11	$g\cos(q_2 + q_3)$
12	$m_4\dot{q_1}^2\sin(2q_2)$
13	$\dot{q_1}^2\sin(2q_2)$
14	$\ddot{q_1}\sin(q_2 + q_3)$
15	$g\sin(q_2 + q_3)$
16	$\ddot{q_1}\cos q_2$
17	$gm_4\cos q_2 - F_{lx}\sin q_3 - F_{ly}\cos q_3$
18	$g\cos q_2$
19	$\ddot{q_1}\sin q_2$
20	$g\sin q_2$
21	$\ddot{q_1}$
22	$m_4\ddot{q_2}$
23	$\ddot{q_2}$
24	$\ddot{q_3}$

Table 3.4.: TUDOR link 2 dynamic regressors

If generalized joint coordinate derivatives $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are set to zero in the above dynamic regressors in Tables 3.4 and 3.5 it can be verified that these will reduce to the

1	1
2	$-\dot{q_1}^2(3\cos{(2q_2)}+3\cos{(2q_3)}-\cos{(2q_2+2q_3)}-9)$
3	$m_4(8\ddot{q_2}+8\ddot{q_3}-3\dot{q_1}^2\sin{(2q_2)}-3\dot{q_1}^2\sin{(2q_3)}+\dot{q_1}^2\sin{(2q_2+2q_3)})$
4	$-\dot{q_1}^2(3\sin{(2q_2)}+3\sin{(2q_3)}-\sin{(2q_2+2q_3)})$
5	$\dot{q_1}^2\cos{q_3}+2\dot{q_2}^2\cos{q_3}+\dot{q_1}^2\cos{(2q_2+q_3)}-2\ddot{q_2}\sin{q_3}$
6	$m_4(\dot{q_1}^2\sin{q_3}+2\dot{q_2}^2\sin{q_3}+\dot{q_1}^2\sin{(2q_2+q_3)}+2\ddot{q_2}\cos{q_3})$
7	$\dot{q_1}^2\sin{q_3}+2\dot{q_2}^2\sin{q_3}+\dot{q_1}^2\sin{(2q_2+q_3)}+2\ddot{q_2}\cos{q_3}$
8	$\ddot{q_1}\cos{(q_2+q_3)}$
9	$gm_4\cos{(q_2+q_3)}-F_{ly}$
10	$g\cos{(q_2+q_3)}$
11	$\ddot{q_1}\sin{(q_2+q_3)}$
12	$g\sin{(q_2+q_3)}$
13	$\ddot{q_1}$
14	$\ddot{q_2}$
15	$\ddot{q_3}$

Table 3.5.: TUDOR link 3 dynamic regressors

same regressors found in Sections 2.5 and 2.6.4 for the static strain equations for links 2 and 3.

3.4. Dynamic Strain Linear Regression and Results

The quality of excitation signals in parameter identification can greatly effect the accuracy of the parameters identified (Armstrong 1989). The MERIt02 dataset contains the TUDOR robot with an amplitude-modulated pseudo-random binary signal (APRBS) applied as excitation. For parameter identification in non-linear systems, having a signal that varies not just in frequency, but in amplitude such as an APRBS is often necessary to properly excite the system dynamics (Isermann et al. 2011). This makes the signals in MERIt02 well suited for robot dynamics parameter identification (Malzahn et al. 2014c).

The linear regression is performed on the MERIt02 dataset with the active damping described in Section 1.3.1 activated. A 60-40 split of training and test data was used. The performance of the dynamic strain equations against dynamic data is better than with only static equations (Section 3.1) and with static equations trained against dynamic data (Section 3.2). The performance can be seen summarized in Table 3.6. When compared to the identical test using the static equations on the same dynamic data (Section 3.1), there is a 17-19% reduction in RMSE for link 2 strain gages and 9-10% for link 3 strain gages.

Looking at estimates of strain as compared to the actual measurements in Figures 3.3 and 3.4 we can see that performance is greatly increased as compared to static strain reproduction in dynamic cases as in Figures 3.1 and 3.2. Whereas the static equations tend to underestimate the strain heavily during changes in velocity, the dynamic equations represent the peaks quite well. Large spikes can be seen in the estimates that do not match the actual strain data. These can be seen in strain

Strain Gage	Training Performance RMSE [μm/m]	Training Performance NRMSE	Test Performance RMSE [μm/m]	Test Performance NRMSE	RMSE % Decrease from Static
21	60.62	0.0282	60.68	0.0285	18.63
22	57.13	0.0500	57.27	0.0491	17.39
31	17.19	0.0305	17.14	0.0313	9.80
32	7.93	0.0326	7.91	0.0321	10.41

Table 3.6.: Strain prediction error using dynamic equations tested against dynamic data. The percentage of error reduced from the similar test using the static equations in Table 3.1 is shown. NRMSE is normalized by the range of the dataset strain measurements.

predictions for all 4 strain gages and are caused by the angular acceleration signal being very noisy due to being numerically differentiated twice from the joint angle sensor measurements.

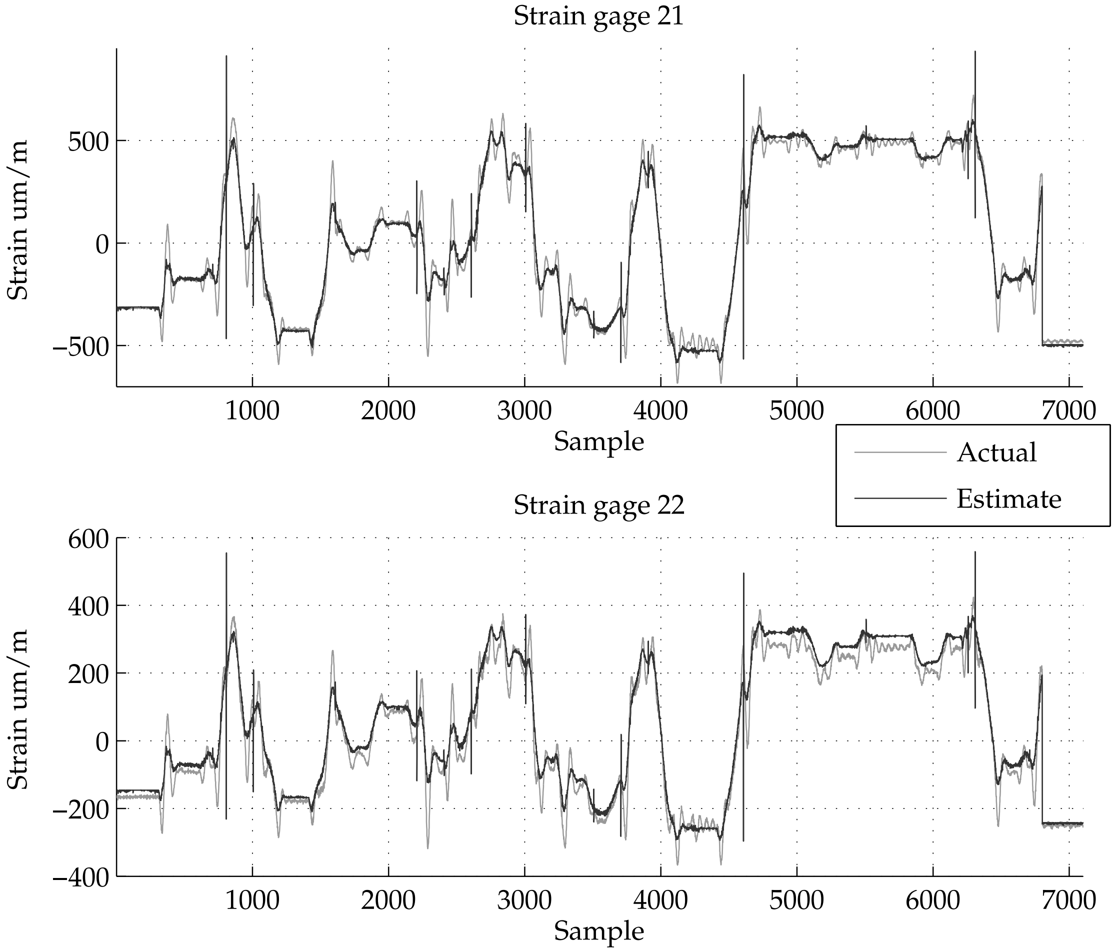


Figure 3.3.: Actual strain from dynamic data with 300g payload, and predicted values using dynamic equations trained on static data for strain gages 21 and 22

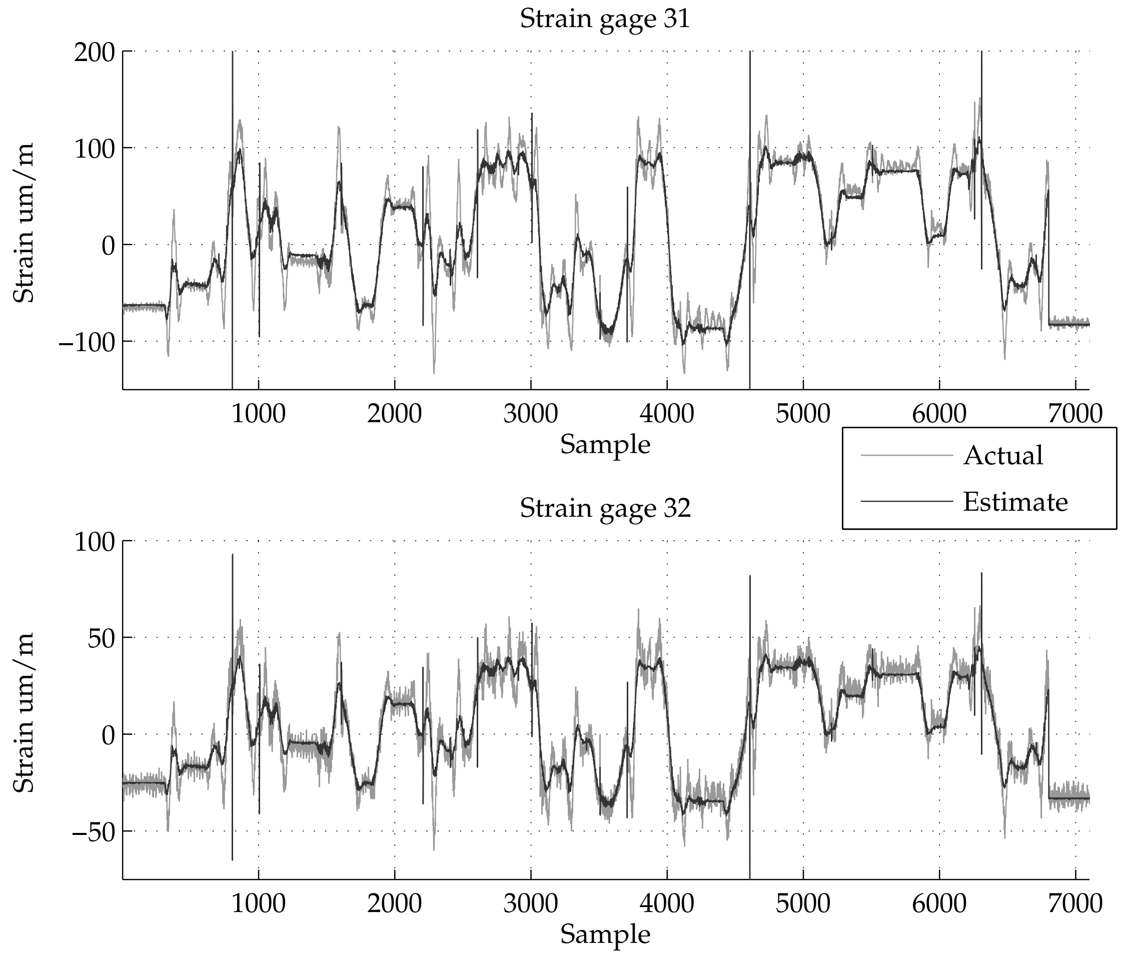


Figure 3.4.: Actual strain from dynamic data with 300g payload, and predicted values using dynamic equations trained on static data for strain gages 31 and 32

3.5. Improving Numerical Differentiation for Velocity and Acceleration

The joint sensors only provide angle measurements and the angular velocity and acceleration must be calculated in some way. The angular velocity and acceleration in the MERIt datasets comes from the method of finite differences. This is quite noisy and results in the singularities occurring in the strain estimates in Figures 3.3 and 3.4. Any noise in the angle measurement is typically greatly amplified by discrete differentiation methods such as this.

3.5.1. Link Angle Kalman Filter and Results

To provide an estimate of the angular velocity and angular acceleration, an observer such as a Kalman filter can be used. A continuous Wiener process acceleration (CWPA) or white noise jerk model model was chosen. This makes the assumption that acceleration is constant, but influenced by a jerk that is Gaussian distributed noise. This is a common model for this type of system (Bar-Shalom et al. 2001). It should be noted that due to various friction effects there are in reality quite large spikes in angular acceleration for this robot, but for the purposes of dynamic strain prediction, it helps greatly to neglect this.

As an example with a single joint, the state space vector is chosen as in Equation 3.5.1 below.

$$\boldsymbol{x} = \begin{bmatrix} q_1 & \dot{q}_1 & \ddot{q}_1 \end{bmatrix}^T \tag{3.5.1}$$

The system is modeled as a linear time-invariant system as shown in Equation 3.5.2.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{v}(t) \tag{3.5.2a}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \tag{3.5.2b}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{D} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{3.5.2c}$$

When designing an observer it is routine to check the observability. The rank of the observability matrix is found to be always 3, meaning the system is observable. This can be seen in Equation 3.5.3.

$$\boldsymbol{\mathcal{O}} = \left[\begin{array}{c} \boldsymbol{C} \\ \hdashline \boldsymbol{CA} \\ \hdashline \boldsymbol{CA}^2 \end{array}\right] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad rank(\boldsymbol{\mathcal{O}}) = 3 \tag{3.5.3}$$

This continuous system is extended to 3 joints and then discretized for use with the Kalman filter. For Kalman filtering related functions a library from Alto University's Biomedical Engineering and Computational Science department was used (Särkkä et al. 2011). When using a Kalman filter, the tuning of process noise variance $\boldsymbol{Q}$ and measurement noise variance $\boldsymbol{R}$ is a main concern. To deal with this, the RMSE of the dynamic measurements was treated as a fitness and minimized using global optimization against the $\boldsymbol{Q}$ and $\boldsymbol{R}$ parameters of interest.

For a global optimization algorithm Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen et al. 2001) was chosen and a library from Universitè Paris-Sud was used (Hansen 2012). A 60-20-20 percent split of the MERIt02 dynamic data was used for training, cross-validation, and testing. The parameters for measurement and process noise in Equation 3.5.4 were the best found after multiple runs of the search.

$$\boldsymbol{R} = \begin{bmatrix} 2.0592 \times 10^{-08} & 0 & 0 \\ 0 & 2.0592 \times 10^{-08} & 0 \\ 0 & 0 & 9.0389 \times 10^{-11} \end{bmatrix} \qquad \boldsymbol{P} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix} \tag{3.5.4}$$

The element of $\boldsymbol{R}$ corresponding to joint 1 was set to the same value as joint 2 due to a lack of data to examine for joint 1 motion in the MERIt02 dynamic dataset.

Predicted strain with and without Kalman filtered angles can be seen in Figure 3.5. The singularities seen previously are almost completely eliminated by the addition of the Kalman filter. In addition, the tracking of the strain estimates is significantly improved with respect to measured strain. This can be seen quantitatively in Table 3.7,

Strain Gage	Unfiltered RMSE [μm/m]	Unfiltered NRMSE	Kalman Filtered RMSE [μm/m]	Kalman Filtered NRMSE	RMSE % Decrease
21	60.68	0.0285	42.80	0.0233	29.46
22	57.27	0.0491	37.91	0.0337	33.81
31	17.14	0.0313	12.07	0.0273	29.56
32	7.91	0.0321	5.91	0.0307	25.29

Table 3.7.: Improvements of dynamic strain prediction using Kalman filtered joint angles. A percentage reduction of RMSE with Kalman filtering enabled is shown. NRMSE is normalized by the range of the dataset strain measurements.

where the RMSE has been reduced by 25 to 33 percent as a result of the addition of the Kalman filter. It can also be seen qualitatively as improved tracking of strain estimates to actual measured values as seen in Figure 3.6 for strain gage 32.

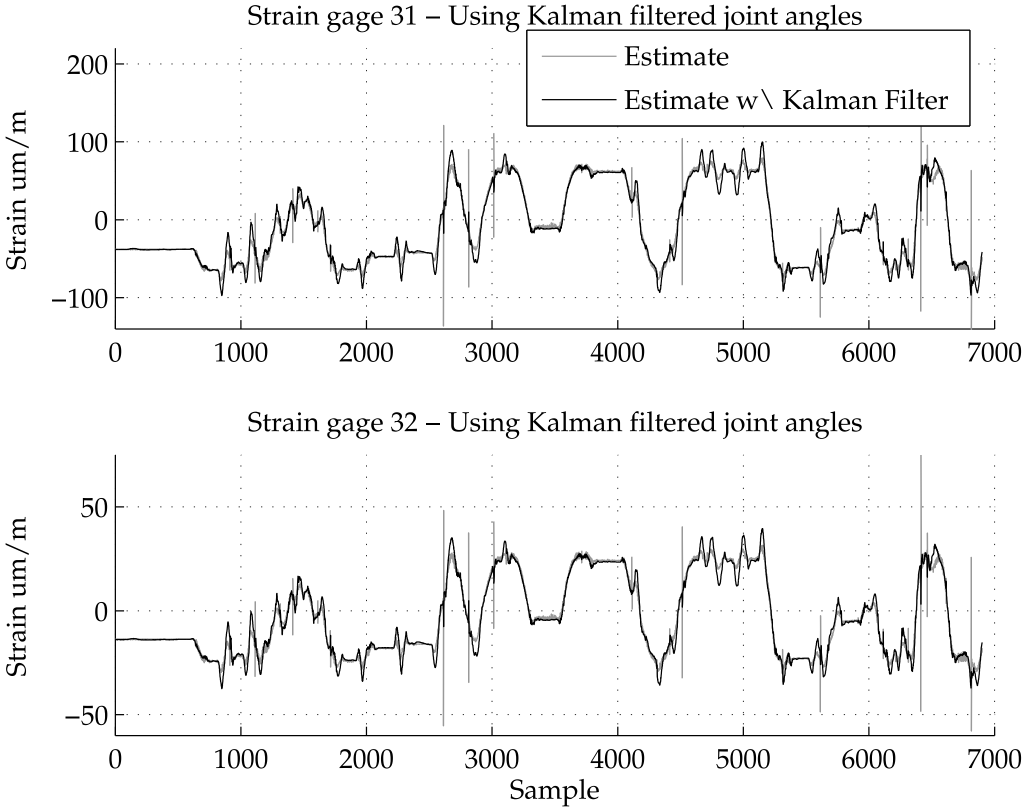


Figure 3.5.: Predicted strain values with 300g payload using dynamic equations trained on dynamic data for strain gages 31 and 32 and angle variables found by finite differences. The same values using Kalman filtered joint angles are shown.

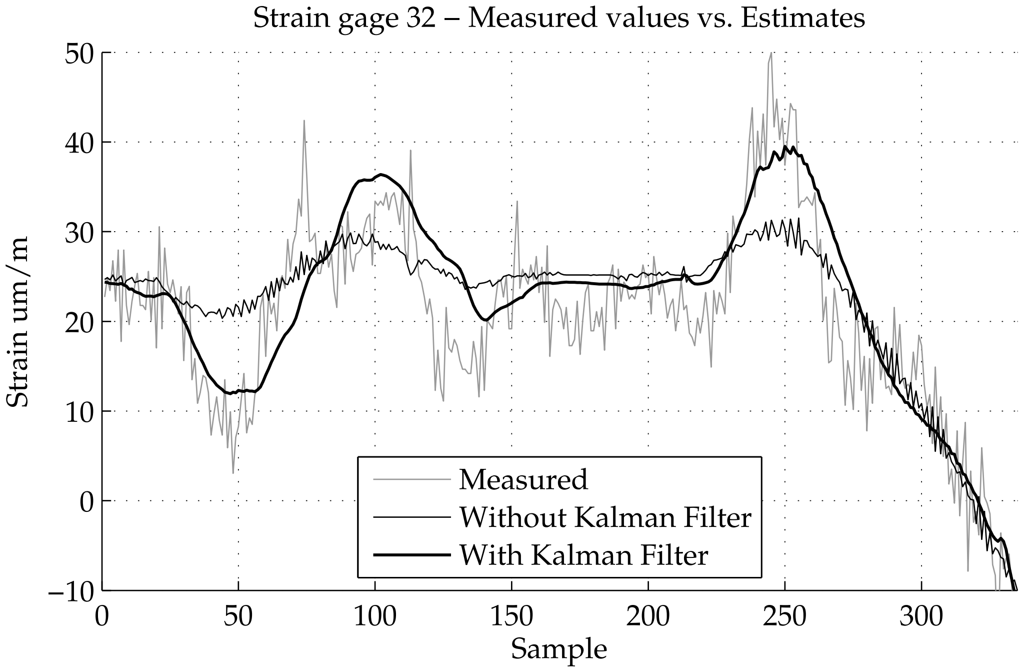


Figure 3.6.: Predicted strain values with 200g payload using dynamic equations trained on dynamic data for strain gage 32. Strain prediction values can be seen to track the measured value more closely using Kalman filtered joint angles.

4

Elastic Link Payload Estimation

4.1. Static Payload Estimation

The equations used in the linear regressions of static strain in Chapter 2 can be rearranged so that for a given strain, joint angles, and effector force, the payload can be estimated. This is synonymous with accounting for a given strain by a nominal strain from the current robot geometry, and accounting for any remainder by a tip force and payload. The assumption however must be made that both the strain and link angles are perfect measurements. Any noise present in the measurements will be accounted for by a tip force or payload which may or may not be physically plausible.

4.1.1. Link 3 Application and Experimental Results

Solving Equation 2.5.4 algebraically for payload m_4 and assuming that the additional effector perpendicular force F_{ly} is known, Equation 4.1.1 can be found. This equation uses the linear regression coefficients found in Section 2.5.

$$m_4 = -\frac{R_1 - F_{ly}R_2 - \mathcal{E}_3(x_{s3}) + g\,R_3\cos(q_2 + q_3)}{g\,R_2\cos(q_2 + q_3)} \tag{4.1.1}$$

Predictions of payload mass were made using Equation 4.1.1, the joint angles and link 3 strain readings for the complete MERIt03 dataset containing static poses as described in Section 1.4. The results can be found in Table 4.1. The errors are split by payload and a total value for all payloads is also given. It can be seen that the error metrics increase in magnitude as the payload increases. NRMSE is normalized by the range of payloads in the dataset (0 to 500 grams).

Looking at equation 4.1.1, it can be seen that there is a $\cos(q_2 + q_3)$ term in the denominator. This equation attempts to explain strain that is different from a nominal model with an added payload mass. If there is any noise present in measurements, this noise will be explained by a payload mass. As link 3 approaches vertical, the strain gauges become less sensitive to an added payload and so any noise must be explained by an increasingly larger payload mass. This dependence of the payload estimation noise can be seen in Figure 4.1. NRMSE by joint angle is shown to approach asymptotes at -90, +90 and +270 degrees (indicated by the dashed lines in the figure), the cosines of which are zero. This leads to the relatively poor overall performance seen in Table 4.1 as high errors at certain angles drastically increase the error. This

	Strain Gage 31		Strain Gage 32	
Payload [g]	RMSE [g]	NRMSE	RMSE [g]	NRMSE
0	24.61	0.0492	41.48	0.0830
100	32.60	0.0652	37.98	0.0760
200	53.86	0.1077	59.99	0.1198
300	76.77	0.1535	75.78	0.1516
400	104.63	0.2093	97.16	0.1943
500	137.79	0.2756	129.76	0.2595
All	82.05	0.1641	80.39	0.1608

Table 4.1.: Payload identification error using link 3 static strain equations. NRMSE is normalized by the range of payloads in the dataset.

graph in Figure 4.1 was created using a kinematics identification dataset created in (Malzahn et al. 2014a). This dataset offers a much more diverse set of link 3 angles to horizontal, providing a better picture of the angle-related effects occurring than MERIt03 static payload estimates.

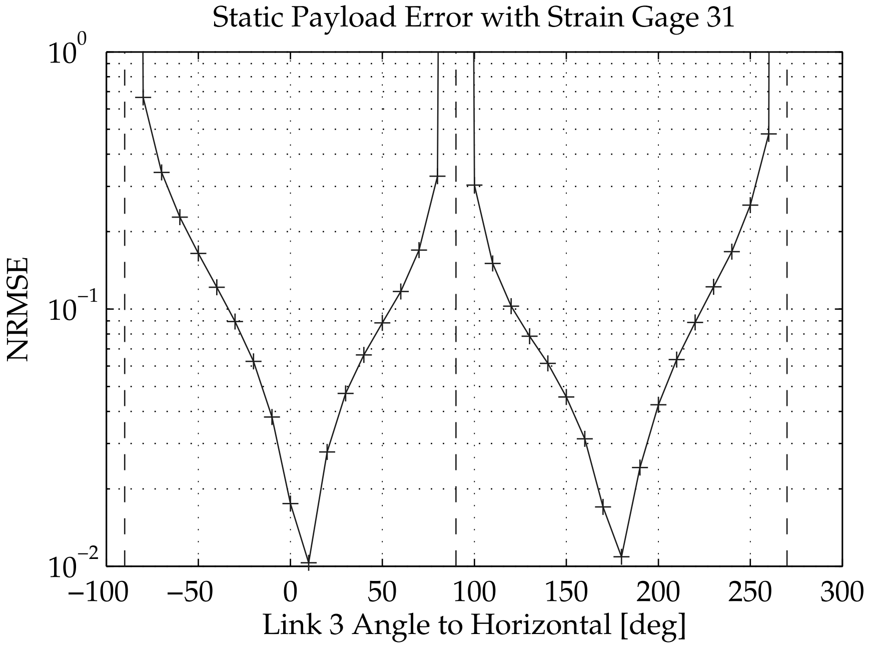


Figure 4.1.: Payload estimation error using strain gage 31. NRMSE is graphed against link 3 angle to horizontal. NRMSE is normalized by the range of payloads in the dataset. This was produced using a kinematics identification dataset similar to MERIt03, but with greater diversity of angles. The results are similar to those found on MERIt03, but easier to visualize with this dataset. Apparent asymptotes are indicated with dashed lines.

Performance at angles where the cosine of the link 3 angle to horizontal is one (0 and 180 degrees) have very low NRMSE. It can be considered when estimating the payload that it would be feasible to constrain the robot arm to certain poses to improve performance. If link 3 is maintained within +/− 20 % from horizontal (irrespective of individual joint angles), the payload estimation performance can be increased markedly as seen in Table 4.2. Constraining the link angle before estimating the error reduced the RMSE of the estimates by 85 and 82.5 percent for strain gages 31 and 32 respectively. This is a very sizable improvement and allows quite accurate estimates of payloads given certain conditions.

	Strain Gage 31			Strain Gage 32		
Payload [g]	RMSE [g]	NRMSE	% RMSE Reduced	RMSE [g]	NRMSE	% RMSE Reduced
0	4.27	0.0085	82.64	8.22	0.0165	80.17
100	4.89	0.0098	85.01	11.76	0.0235	69.04
200	5.03	0.0101	90.65	6.47	0.0129	89.21
300	9.89	0.0198	87.11	10.13	0.0203	86.63
400	17.00	0.0340	83.75	19.66	0.0393	79.76
500	21.27	0.0426	84.56	22.12	0.0443	82.95
All	12.25	0.0245	85.07	14.28	0.0286	82.24

Table 4.2.: Payload identification error using static link 3 strain prediction equations with limited link 3 angle with respect to horizontal. Errors are shown grouped by payload and in total. The percent RMSE is reduced from predictions in Table 4.1 with no link angle constraints is shown. NRMSE is normalized by the range of payloads in the dataset.

4.1.2. Link 2 Application and Experimental Results

As with link 3, Equation 2.5.4 for link 2 is solved algebraically for payload m_4 as shown in Equation 4.1.2a. Assuming that the additional effector perpendicular forces F_{lx} and F_{ly}, link angles, and strain are known, the mass can be found. The numerator and denominator can be represented as functions N_2 and D_2 as in Equation 4.1.2b and will be used later in this section.

$$m_4 = \frac{\mathcal{E}_2(x_{s2}) - R_1 + F_{ly}R_2 - g\left(R_5 \cos q_2 + R_3 \cos\left(q_2 + q_3\right)\right) + R_4(F_{ly}\cos q_3 + F_{lx}\sin q_3)}{g(R_4 \cos q_2 + R_2 \cos\left(q_2 + q_3\right))} \tag{4.1.2a}$$

$$= \frac{N_2(F_{lx}, F_{ly}, q_2, q_3, x_2)}{D_2(q_2, q_3)} \tag{4.1.2b}$$

This is then used to calculate the payload for given strain and angle measurements from the MERIt03 dataset. Results are shown in Table 4.3. These payload estimates have considerably more error as compared to the link 3 estimates seen in Table 4.1. With NRMSE values over 1, the estimates from strain gage 22 are very bad. Looking at Equation 4.1.2a above, the denominator approaches zero for certain angle combinations, causing a singularity. These angle combinations are more complex than those encountered previously with link 3. The regression parameters weight the cosine of q_2 and the cosine of link 3 angle to horizontal. As the regression parameters are different for each strain gage, this means that different angles will cause the mass estimate to become singular for each strain gage. Strain gage placement on link 2 can be said to change the position of payload estimate singularities in joint space.

	Strain Gage 21		Strain Gage 22	
Payload [g]	RMSE [g]	NRMSE	RMSE [g]	NRMSE
0	110.73	0.2215	414.18	0.8284
100	102.19	0.2044	575.91	1.1518
200	175.45	0.3509	455.18	0.9104
300	124.35	0.2487	334.42	0.6688
400	175.70	0.3514	496.71	0.9934
500	175.37	0.3508	560.41	1.1208
All	147.52	0.2950	480.11	0.9602

Table 4.3.: Payload identification error using link 2 strain prediction equations. Error is split by payload and shown in total. NRMSE is normalized by the range of payloads in the dataset.

In Equation 4.1.2b the mass estimate equation for link 2 is split into numerator and denominator functions N_2 and D_2. The absolute value of the denominator function D_2 can be examined as a metric for how singular the mass estimate is for any given pose. This is identical to the technique when using Singular Value Decomposition to examine the singular values of a matrix to avoid singularities in the robot Jacobian (Maciejewski et al. 1989).

Using the MERIt03 dataset and limiting the angles to cases where the denominator is over a certain cutoff value can be shown to reduce the payload estimate errors. This is a sort of trade-off: more accurate payload estimates can be made, but only by increasingly limiting the joint angles at which estimates can be made. This is visualized in Figure 4.2. The percentage of removed values in this graph refer to the percent of tests that had to be removed from MERIt03 due to the denominator cutoff. As MERIt03 spans the overall working space of joints 2 and 3, this can be treated as a metric for loosely evaluating how limited the joint space of the robot will be.

As an example, denominator cutoff values of 250 and 175 are chosen for payload estimation for strain gages 21 and 22. These cutoff values remove only 34.7% of the available angles from MERIt for both strain estimates for each strain gage. If the removed sets of angles for both strain gages are combined they account for 40.8% of available angles. The performance under these cutoffs can be seen in Table 4.4 and the angles allowed are visualized in Figure 4.3. The RMSE using these limiting coefficients is 66.69% and 89.77% of the RMSE without (in Table 4.3) for strain gages 21 and 22. This is a dramatic increase and shows that this method can be used to improve static payload estimates without constraining the workspace extensively.

4.2. Static Sensor Fusion for Payload Estimation

While Section 4.1 focused on determining when payloads estimates were most accurate for each strain gage, a large variety of techniques referred to as sensor fusion can be used to combine measurements from multiple sensors. This can sometimes provides more accurate, less noisy, and more stable results. This section will focus on the application of sensor fusion techniques to static payload estimation as an alternative to limiting the allowed joint angles as in Section 4.1.

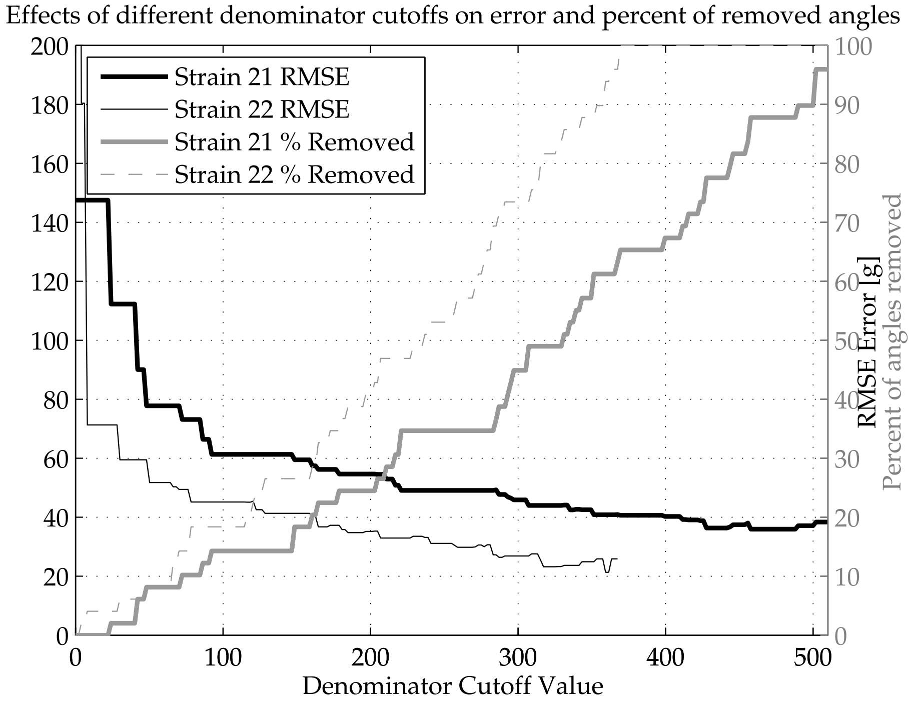


Figure 4.2.: Effects of various denominator cutoffs on payload RMSE and percent of angles removed from the dataset. Left Y-Axis shows RMSE in grams, and the right Y-Axis shows the percent of angles removed.

	Strain Gage 21			Strain Gage 22		
Payload [g]	RMSE [g]	NRMSE	% RMSE Reduced	RMSE [g]	NRMSE	% RMSE Reduced
0	31.56	0.0631	71.50	17.58	0.0352	95.75
100	36.44	0.0729	64.34	25.70	0.0514	95.54
200	43.64	0.0873	75.13	26.16	0.0523	94.25
300	40.35	0.0807	67.55	35.21	0.0704	89.47
400	63.31	0.1266	63.97	45.55	0.0911	90.83
500	67.97	0.1359	61.24	58.13	0.1163	89.63
All	49.13	0.0544	66.69	37.30	0.0746	92.23

Table 4.4.: Payload identification error using static link 2 strain prediction equations with mass estimate denominator cutoffs of 250 and 175 for strain gages 21 and 22 respectively. Error is shown grouped by payload and in total. The percent RMSE is reduced from predictions in Table 4.3 with no link angle constraints is shown. NRMSE is normalized by the range of payloads in the dataset.

4.2.1. Effects of Strain Gage Noise on Payload Estimates

A random variable X, with linear function f applied can be demonstrated to produce Y, which can also be treated as a random variable. If the linear function f can be represented as in Equation 4.2.1a, then the expected value of X can be related to Y as in Equation 4.2.1b and the variance of measurement X can be related to the variance

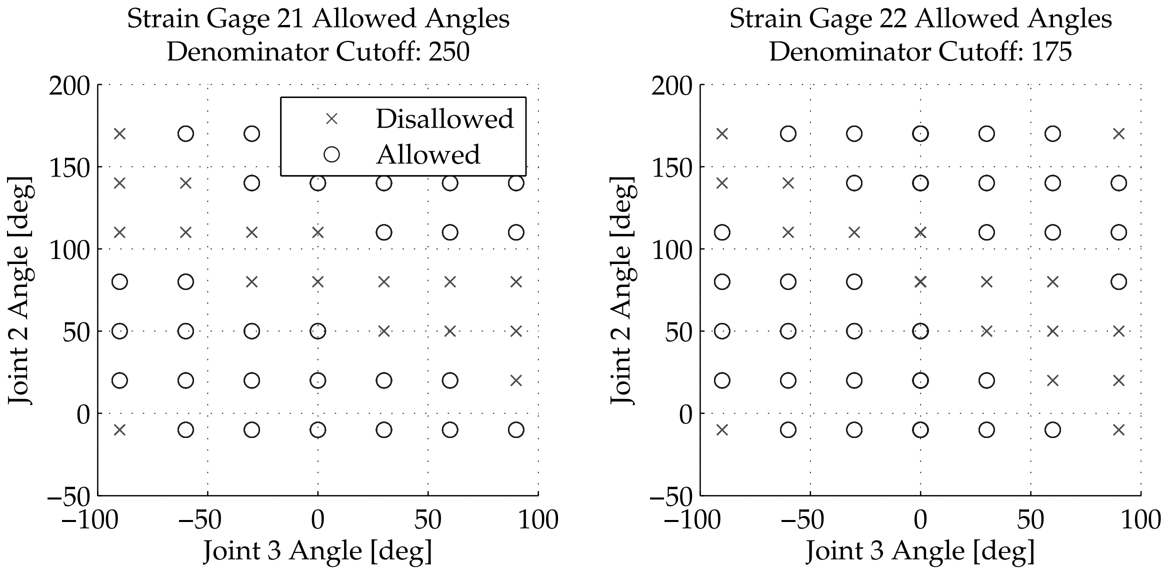


Figure 4.3.: Disallowed angles by choice of denominator cutoffs for payload estimation from strain gages 21 and 22

of Y as in Equation 4.2.1c (Montgomery et al. 2010, p. 182)(Lohr 1999).

$$Y = f(X) = aX + b \tag{4.2.1a}$$

$$E[Y] = E[aX + b] = E[ax] + c = aE[x] + c \tag{4.2.1b}$$

$$Var[Y] = Var[aX + b] = a^2 Var[X] \tag{4.2.1c}$$

For more complicated cases, a nonlinear function f with respect to n uncorrelated random variables can be represented as in Equation 4.2.2a. Using a first order Taylor series expansion the variance can be estimated as in Equation 4.2.2b (Lohr 1999).

$$y = f(x_1, x_2, ..., x_n) \tag{4.2.2a}$$

$$\sigma_y^2 = \left(\frac{\partial f}{\partial x_1}\right)^2 \sigma_{x_1}^2 + \left(\frac{\partial f}{\partial x_2}\right)^2 \sigma_{x_2}^2 + ... + \left(\frac{\partial f}{\partial x_n}\right)^2 \sigma_{x_n}^2 \tag{4.2.2b}$$

Strain Gage Variance

Using the MERIt03 dataset of the robot holding various static poses, the variance of strain measurements can be found. The MERIt03 dataset contains 2940 15 second recordings of data spanning the workspace of TUDOR. The last 5 seconds of 100Hz data recorded at each of these 2940 poses is used. As a result, the variance approximations for each sensor can be made from 1.47 million points. The average strain of each static pose is subtracted from the measurements and collected in a table where the variances across all measurements are found. The standard deviation can then be calculated to give more physically meaningful values. Both these variances and standard deviations are listed in Table 4.5 for each of the 4 strain gages on TUDOR.

Strain Gage	Variance $[\mu m^2/m^2]$	Standard Deviation $[\mu m/m]$
21	86.645	9.308
22	37.597	6.132
31	12.921	3.595
32	4.339	2.083

Table 4.5.: Approximations of variance and standard deviation of strain measurements with TUDOR in static poses.

Link 3

Equation 4.1.1 for estimating the payload from static link 3 strain can be rearranged into the same form as equation 4.2.1c as shown in Equation 4.2.3. Given the variance of the strain gage and the assumption that link angles are measured perfectly, the variance of the mass estimate can be found as in Equation 4.2.4

$$\begin{aligned} \hat{m}_4 &= \frac{1}{gR_2\cos(q_2+q_3)}\mathcal{E}(x_3) - \frac{R_1 - F_{ly}R_2 + gR_3\cos(q_2+q_3)}{gR_2\cos(q_2+q_3)} && (4.2.3a) \\ &= a\mathcal{E}(x_3) + b && (4.2.3b) \end{aligned}$$

$$\begin{aligned} Var[m_4] &= a^2 Var[\mathcal{E}(x_3)] && (4.2.4a) \\ &= \frac{1}{g^2R_2^2\cos^2(q_2+q_3)} Var[\mathcal{E}(x_3)] && (4.2.4b) \end{aligned}$$

Previously in Sections 4.1.2 and 4.1.1, the conclusion had been made that at certain joint angles, the amount strain added from a payload approached zero. As this occurs, in the presence of sensor noise, this leads to any unexplained strain needing to be explained by an increasingly large payload mass. This can be seen in the Equation 4.2.4 above as well. As the cosine of the angle of link 3 with respect to horizontal approaches zero, the variance of the mass estimate will approach infinity.

This can be visualized in Figure 4.4 using the linear regression parameters determined in Section 2.5 and strain gage variances from Table 4.5. It should be noted that Figure 4.4 is quite similar to Figure 4.1 calculated from the NRMSE of payload identification against actual data with varying angles. Additionally, the standard deviation values in this chart are quite similar to values seen previously in mass estimation in Section 4.1.

The increased variance of mass estimates from strain gage 32, as compared to strain gage 31, can be explained by looking into the analytic equations for the parameter R_2 in Equation 4.2.5

$$R_2 \approx -\frac{y_b(a_3 - x_3)}{EI} \qquad (4.2.5)$$

As the strain gage position becomes farther from the joint, R_2 will become smaller, and the variance of the mass estimate becomes larger. This difference in the R_2 pa-

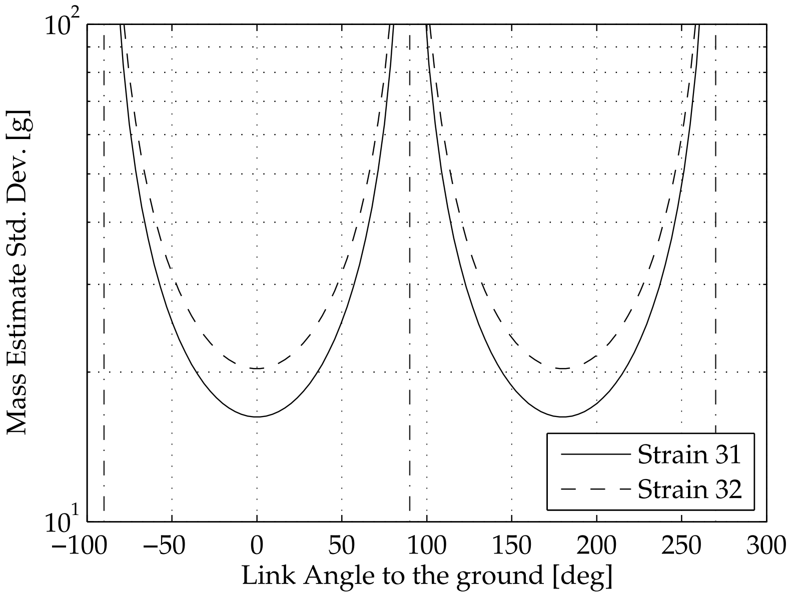


Figure 4.4.: Estimate of standard deviation of payload estimates using link 3 strain gages with variance estimates from Table 4.5. Singularities at the angles -90, 90, and 270 are shown.

rameter for strain gages with different positions appears in the results of the linear regression as well in Table 2.3.

Link 2

The same procedure can be followed for link 2, rearranging the equation for link 2 strain into the same form of equation 4.2.1c as shown in Equation 4.2.6. It can be seen in Equation 4.2.6c below, that the mass estimate variance depends on both the angle of joint 2 and the angle of link 3 to horizontal.

$$\hat{m}_4 = \frac{1}{g(R_4 \cos q_2 + R_2 \cos(q_2 + q_3))} \mathcal{E}_2(x_{s2}) \tag{4.2.6a}$$

$$- \frac{R_1 - F_{ly}R_2 + gR_5 \cos q_2 + gR_3 \cos(q_2 + q_3) - F_{ly}R_4 \cos q_3 - F_{lx}R_4 \sin q_3}{g(R_4 \cos q_2 + R_2 \cos(q_2 + q_3))} \tag{4.2.6b}$$

$$Var[m_4] = a^2 Var[\mathcal{E}_s(x_{s2})] = \frac{1}{(gR_4 \cos q_2 + gR_2 \cos(q_2 + q_3))^2} Var[\mathcal{E}_2(x_{s2})] \tag{4.2.6c}$$

This can be visualized in Figure 4.5 for strain gage 21. The similarity between the disallowed link angles from Figure 4.3 can be seen to match with the singularity shown in this graph.

4.2.2. Maximum Likelihood Estimation

As there are 2 strain gages on each link it may be possible to combine measurements from each to obtain a more accurate estimate of the payload. Ideally, it would be good to integrate knowledge about the variance of the measurements and how this changes

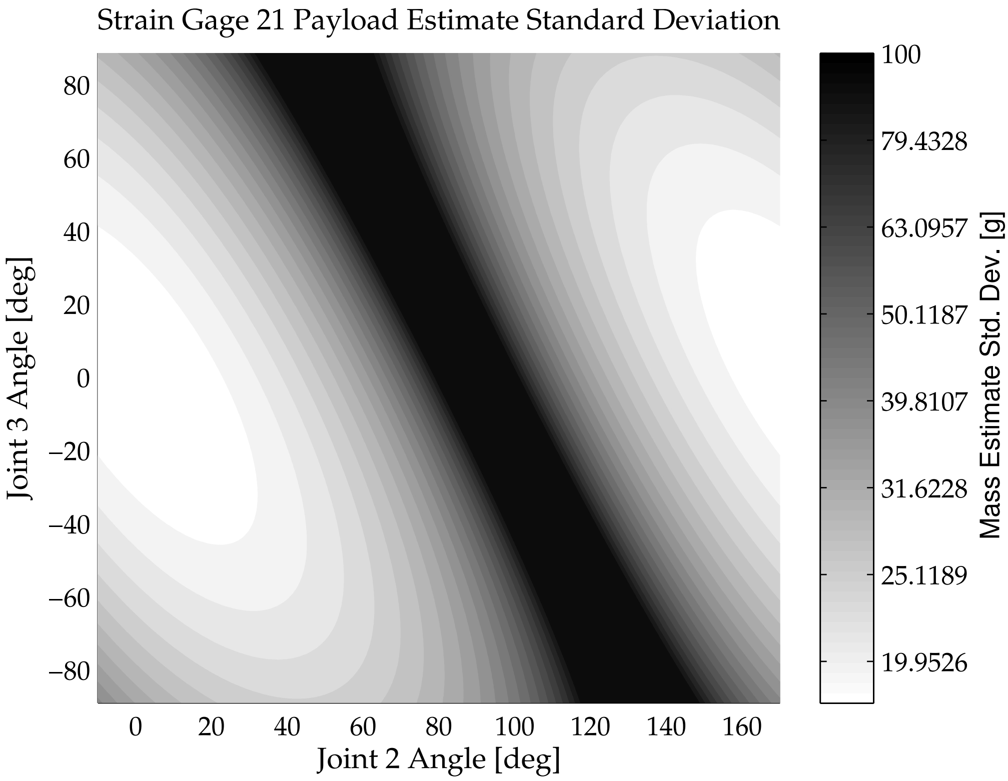


Figure 4.5.: Payload estimate standard deviation for link 2 strain gage 21. A logarithmic scale for the color is used and the maximum is clipped at 100 grams.

as payload is estimated. If it is assumed that the sensors are independent and behave in a Gaussian fashion, Maximum Likelihood Estimation (MLE) can be used as a tool to accomplish this (Zisserman 2007).

It can be said that the likelihood $\mathcal{L}$ defines the likelihood of recording measurements z_i of the true value we are trying to measure x as in Equation 4.2.7 (Zisserman 2007).

$$\mathcal{L}(x) = P(z|x) \tag{4.2.7}$$

Given multiple observations, the most likely estimate of x can be defined as a maximization of the likelihood, or more usefully, when using Gaussian distributions, as a minimization of the negative log likelihood as in Equation 4.2.8 (Zisserman 2007).

$$\hat{x} = \arg\min_x \{-\ln \mathcal{L}(x)\} \tag{4.2.8}$$

With two independent, Gaussian sensors each making measurement z_n of some true value x, the likelihood function can be defined by Equation 4.2.9 (Zisserman 2007).

$$\mathcal{L}(x) = P(z_1, z_2|x) = P(z_1|x)P(z_2|x) \tag{4.2.9}$$

Substituting the Gaussians representing each sensor, the likelihood function can be found to be proportional to Equation 4.2.10 (normalization constants are ignored).

$$\mathcal{L}(x) \propto e^{-\frac{(z_1 - x)^2}{2\sigma_1^2}} e^{-\frac{(z_2 - x)^2}{2\sigma_2^2}} \tag{4.2.10}$$

Using the assumption that the distributions for both sensors are Gaussian, this can then be simplified to give the maximum likelihood estimate of x as well as the variance of the estimate as shown in Equations 4.2.11. The final estimate can be seen as a linear combination of the measurements with weights w_n. Identical result can be found by applying the central-limit theorem. (Zisserman 2007).

$$\hat{x}_{MLE} = \frac{\sigma_1^{-2}}{\sigma_1^{-2} + \sigma_2^{-2}} z_1 + \frac{\sigma_2^{-2}}{\sigma_1^{-2} + \sigma_2^{-2}} z_1 \tag{4.2.11a}$$

$$= w_1 z_1 + w_2 z_2 \tag{4.2.11b}$$

$$\sigma_x^{-2} = \sigma_1^{-2} + \sigma_2^{-2} \tag{4.2.11c}$$

MLE Results

An overview of the results of the MLE sensor fusion performed on all payloads for all angles in the MERIt02 dataset can be seen in Table 4.6. The fused error for link 2 and link 3 estimates are compared to the original mass estimates from Table 4.1 and 4.3 without any methods to avoid singularities.

	RMSE [g]	NRMSE
Strain Gage 21	147.52	0.2950
Strain Gage 22	480.11	0.9602
Strain Gage 31	84.91	0.1698
Strain Gage 32	83.33	0.1667
Link 2 Fusion	74.78	0.1496
Link 3 Fusion	82.38	0.1648
All Strains Fusion	43.13	0.0863

Table 4.6.: Error in payload estimations using individual strain gages and MLE sensor fusion with all link angles. NRMSE is normalized by the range of payloads in the dataset.

It can be seen that for link 3 there is a small improvement, while for link 2 there is quite a large improvement. This can be explained by looking at the nature of the singularities experienced for link 2 and link 3. Looking at the estimates of payload estimate standard deviations of link 3 sensors in Figure 4.4, it can be seen that both strain gages approach singularities in the same place. In fact, the two curves are always the same distance for 2 link 3 strain gages and so the MLE algorithm chooses 2 fixed weights for strain gages 31 and 32 (0.61 and 0.39 respectively). For link 2 payload estimates, the singularities are positioned in joint space differently for different strain gage positions. This has been visualized in Figure 4.5 where the payload estimate standard deviations for strain gages 21 and 22 have been added and plotted with a logarithmically scaled shading. The two singularities can be clearly seen.

Due to the orientation of the singularities, the MLE algorithm is sometimes able to favor one sensor over the other. This results in the much improved mass estimation in Table 4.6. Additionally, this accounts for the improved results when all 4 strain gages are fused together.

From these results, it can be concluded that MLE is a viable alternative to limiting the link angles for static payload estimates. Additional tests were made combining

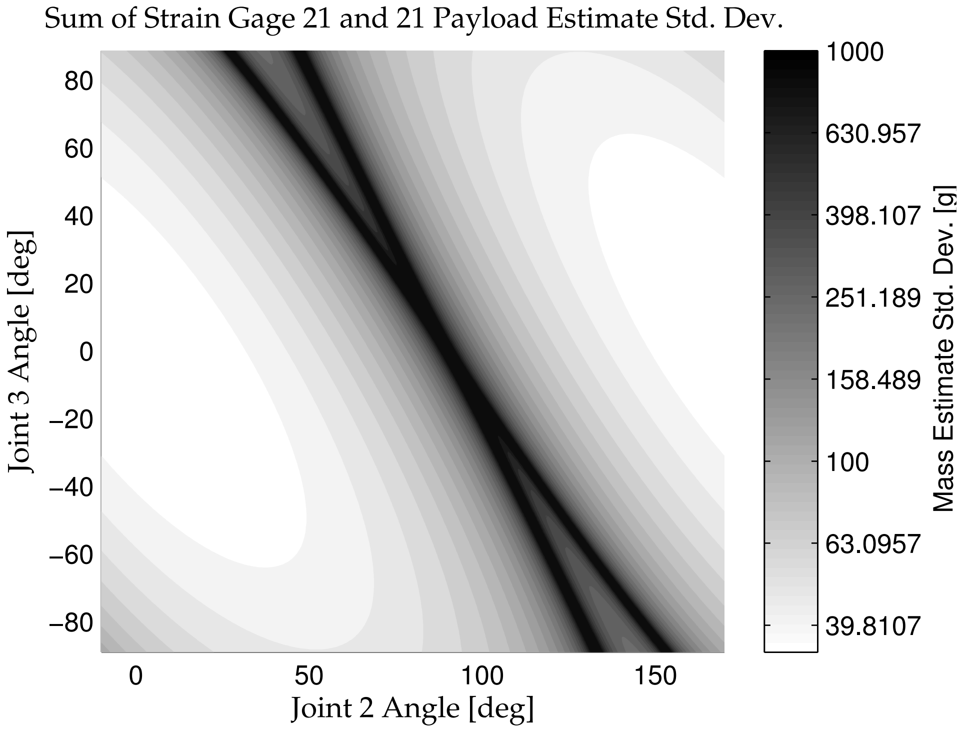


Figure 4.6.: The sum of payload estimate standard deviation for link 2 strain gages 21 and 22. This visualizes the different singularities in the joint space of the robot for link 2 strain gages. A logarithmic scaling of the color shading is used and the maximum is clipped at 1000 grams.

limits to link angles and MLE. Simply limiting joint angles was found to provide slightly better performance in terms of the error metrics above. Using sensor fusion provides the added benefit of being able to account for sensor noise. If a robot can perform a startup calibration it could use MLE to account for sensor noise changing as the sensors age. Also, in the event of a failure of one of the sensors, others could be automatically favored, adding a level of robustness that purely limiting link angles does not provide. The ability to use multiple strain gages to reduce the effects of singularities also should be considered when designing elastic link robots.

4.2.3. Maximum a posteriori Estimation

Maximum a posteriori estimation (MAP) can combine posterior information about the payload to improve estimates. For example, if the robot were weighing fruit, a prior for the payload estimation could be used to make more accurate estimations of the fruit mass. As the payload masses used in the datasets in this paper are uniformly distributed this does not provide any gains for this dataset, however depending on the application, it should be considered.

4.3. Dynamic Payload Estimation

While the static payload estimation is quite useful, the ability to measure while in motion would allow the robot to operate normally without having to stop to assess the current payload. In a manufacturing setting this could result in greatly increased cycle times. Besides accuracy, there are certain properties that are desirable for a payload estimation algorithm used while the robot is in motion. First, the algorithm should, no matter the motion of the robot, be able to continue operating. This implies that the algorithm should be stable. Second, the algorithm should be able to operate while in motion without diverging too much from the correct value. Third, the algorithm should also be able to reach an estimate value as quickly as possible, despite being in motion.

4.3.1. Ordinary Least Squares of APRBS Signals

The MERIt02 dataset contains the TUDOR robot with various amplitude modulated pseudo-random binary sequences (APRBS) of motion organized into runs (see Section 3.4 for more details). This also makes them very useful in the context of payload identification while in motion. They contain a 80 seconds of 100Hz data in widely varied poses and speeds with different known payloads. In terms of the data that will likely be encountered, this represents the most ideal case that could be hoped for.

The equations for dynamic strain prediction can be reorganized into the form $\boldsymbol{y} = \boldsymbol{X\beta}$, where the parameter vector $\boldsymbol{\beta}$ is the constant payload mass m_4. Substituting generalized robot coordinates as well as measured strain, the vectors $\boldsymbol{y}$ and $\boldsymbol{X}$ can be calculated for the entire run of data and the constant m_4 can then be estimated using linear least-squares as in Equation 4.3.1.

$$\boldsymbol{y} = \boldsymbol{X\beta} = \boldsymbol{X}m_4 \tag{4.3.1a}$$

$$(\boldsymbol{X}^T\boldsymbol{X})m_4 = \boldsymbol{X}^T\boldsymbol{y} \tag{4.3.1b}$$

$$\hat{m}_4 = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{4.3.1c}$$

Least squares estimation of the payload for each run in MERIt02 was performed with and without Kalman filtering of joint angles as described in Section 3.5.1. The comparison between the two can be found in Table 4.7 and details of the results using Kalman filtering, grouped by payload, can be found in Table 4.8.

	Unfiltered		Kalman Filtered		
Strain Gage	RMSE [g]	NRMSE	RMSE [g]	NRMSE	RMSE % Decrease
21	13.39	0.0335	8.55	0.0214	36.17
22	66.19	0.1655	58.09	0.1452	12.24
31	8.73	0.0218	6.83	0.0171	21.76
32	8.44	0.0211	6.72	0.0168	20.39

Table 4.7.: Payload estimation error from ordinary least squares for each strain gage with and without Kalman filtered joint angles, as well as percent reduction in RMSE. NRMSE is normalized by the range of payloads in the dataset.

Again, as before with dynamic strain prediction in Section 3.5.1, the Kalman filtered joint angles provide a substantial reduction in error, keeping the RMSE below 9 grams for at least one sensor on each link. Even without Kalman filtering, for each link, there

Payload	Payload RMSE per strain gage [g]				Payload NRMSE per strain gage			
[g]	21	22	31	32	21	22	31	32
0	6.01	68.41	2.15	2.17	0.0150	0.1710	0.0054	0.0054
100	4.74	101.85	3.46	4.66	0.0118	0.2546	0.0087	0.0116
200	9.55	30.43	7.64	7.42	0.0239	0.0761	0.0191	0.0186
300	9.19	16.10	9.17	8.39	0.0230	0.0402	0.0229	0.0210
400	11.44	25.17	8.62	8.60	0.0286	0.0629	0.0215	0.0215
All	8.56	58.09	6.83	6.72	0.0214	0.1452	0.0171	0.0168

Table 4.8.: Payload estimation RMSE on entire runs of data in MERIt02 using ordinary least squares for each strain gage. Data is grouped by payload and strain gage. NRMSE is normalized by the range of payloads in the dataset.

is at least one strain gage with an RMSE below 14 grams, and an NRMSE below 0.04 on each link. The large error for strain gage 22 result from the poorer dynamic model of this strain. As mentioned in Section 3.1, this is likely caused by the calibration procedure for the strain gages applying a different offset due to sensor noise. This model error is further exaggerated when the mass estimation is performed. Under ideal circumstances, it is possible to to determine payload mass with reasonable accuracy. It should also be noted that the RMSE can be seen to usually increase with larger payloads.

When random variables have a variance that changes with respect to each other or time, they are referred to as heteroscedatic (Gujarati et al. 2002). This concept is more frequently encountered in the field of econometrics and so much of the literature comes from these areas. As was shown in Section 4.2.1, payload estimation is clearly heteroscedatic when the robot is in motion as the variance of the payload estimate will change as the joint angles change.

Ordinary least squares is generally made under the assumption of constant variance or homoscedasticity. The presence of heteroscedatic error does not add a bias or inconsistency to the estimation if given a sufficient amount of data. It does however mean that the ordinary least squares is no longer considered the most optimal estimation (minimum variance unbiased estimator) (Gujarati et al. 2002, p. 394). Despite this, from the results in Tables 4.7 and 4.8, it can be seen that the ordinary least squares is performing quite well. When applying this method, the computation time needed to compute the matrix inversion in Equations 4.3.1 should be considered carefully.

4.3.2. Moving Window Least Squares

In real world applications, collecting 8000 data points over 80 seconds before making a single payload estimation is usually not practical (or computationally feasible in embedded applications). The least squares technique can instead be applied to a small moving window of data. When a new data point is received it is added into the window and the oldest data point is subsequently removed.

Window sizes from 1 to 1000 samples (at 100Hz) were tested against the MERIt02 dataset. These are shown in Figure 4.7 and summarized in Table 4.9. For an increasing window size, the error in estimates can be seen to decrease. With an appropriate choice of window size, the RMSE can be brought below 50 grams, which may be low enough for some applications. It should be noted that as this method uses a matrix inversion, the window size does increase the computation power needed. Various algorithms exist to compute an inverse or pseudo-inverse. As an example, the Williams inverse algorithm for a square matrix of size n can be computed in $\mathcal{O}(n^{2.373})$ (Williams 2011).

	Strain Gage 21		Strain Gage 31	
Window Size	RMSE [g]	NRMSE	RMSE [g]	NRMSE
1	179.52	0.3590	118.11	0.2362
50	114.10	0.2282	88.35	0.1767
100	101.48	0.2030	81.27	0.1625
250	89.17	0.1783	71.52	0.1430
500	75.28	0.1506	54.21	0.1084
750	63.39	0.1268	31.10	0.0622
1000	47.72	0.0954	22.08	0.0442

Table 4.9.: Moving window payload estimation error against the MERIt02 dataset with various window sizes. NRMSE is normalized by the range of payloads in the dataset.

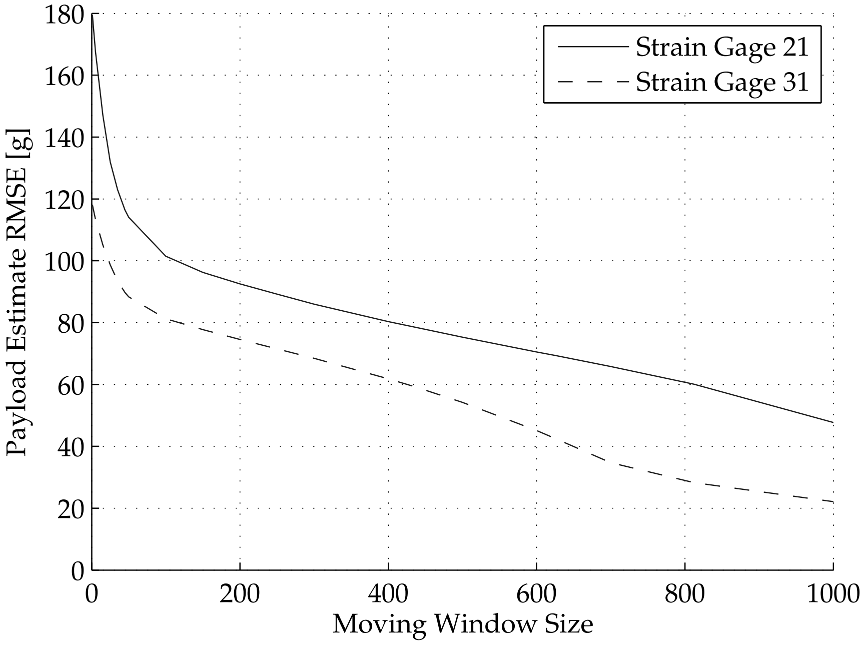


Figure 4.7.: Moving window payload estimation error for strain gages 21 and 31 against the size of the moving window.

4.3.3. Extended Kalman Filter

Extended Kalman filters are often applied as in (Gourdeau et al. 1991) to estimating derivatives of link coordinates as well as parameters of rigid-link robot manipulators. They have also been successfully applied to elastic link, rigid joint robots and rigid link, elastic joint robots for tip position estimation as in (Lertpiriyasuwat et al. 2000) and (Lightcap et al. 2010).

The nonlinear discrete state space system in Equation 4.3.2 is used as a starting point to represent a system with process noise $\boldsymbol{q}_k$, measurement noise $\boldsymbol{r}_k$, states $\boldsymbol{x}_k$, measurements $\boldsymbol{y}_k$, inputs $\boldsymbol{u}_k$ and parameters $\boldsymbol{w}$.

$$\boldsymbol{x}_k = \boldsymbol{F}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}) + \boldsymbol{q}_k \tag{4.3.2a}$$

$$\boldsymbol{y}_k = \boldsymbol{H}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}) + \boldsymbol{r}_k \tag{4.3.2b}$$

The matrix $\boldsymbol{F}$ is composed of the equations from Section 3.5.1 for the joint angles. A constant state is added to represent the payload as well. The measurement equations for strain are built from the robot dynamics linear regression in Section 3.4 and used to form $\boldsymbol{H}$. The state vector, composed of link angles, derivatives and the added payload m_4 can be seen in Equation 4.3.3.

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} & \dot{\boldsymbol{q}} & \ddot{\boldsymbol{q}} & m_4 \end{bmatrix}^T \tag{4.3.3}$$

As in Section 3.5.1, global optimization in the form of CMA-ES was used to optimize the parameters of this extended Kalman filter. Unfortunately, no set of process and measurement noise parameters were stable when moving through singularities in the MERIt02 dataset. The best set of parameters found with CMA-ES is illustrated in Figure 4.8 approaching the correct value while in motion and then at sample 590, as the joint angles approach a singularity, the mass estimate becomes unstable and begins approaching 1×10^{10} kilograms of error. It is likely that the linearization used by the extended Kalman filter is insufficient to approximate the system behavior near singularities as illustrated in the presence of numerical problems often occurring in the matrix operations during these tests.

This approach is clearly not sufficient to meet the desired characteristics for estimating payloads as presented in Section 4.3. This could possibly fixed with the application of higher order Taylor series expansions as shown in (Einicke 2012). Section 4.3.4 however provides a simpler solution.

4.3.4. Dual Extended Kalman Filter

The process of estimating payload while simultaneously estimating joint angles and derivatives is a type of problem often referred to as a dual estimation problem. A dual estimation problem is one in which both system states and system parameters are estimated simultaneously. This is often performed using two Kalman filters (Haykin 2001). The system of joint angles and parameters are represented as a linear (from Section 3.5.1) and nonlinear discrete state space systems using in Equation 4.3.2 as a starting point.

The joint angle Kalman filter and parameters designed in Section 3.5.1 are used to estimate the joint angles and subsequent derivatives. These are then presented as inputs to an extended Kalman filter, along with strain measurements, to estimate the payload mass m_4 as a state for each discrete time step. The extended Kalman filter can be further simplified as the system dynamics for payload estimation can be represented as linear function. This leaves only a nonlinear measurement equation which

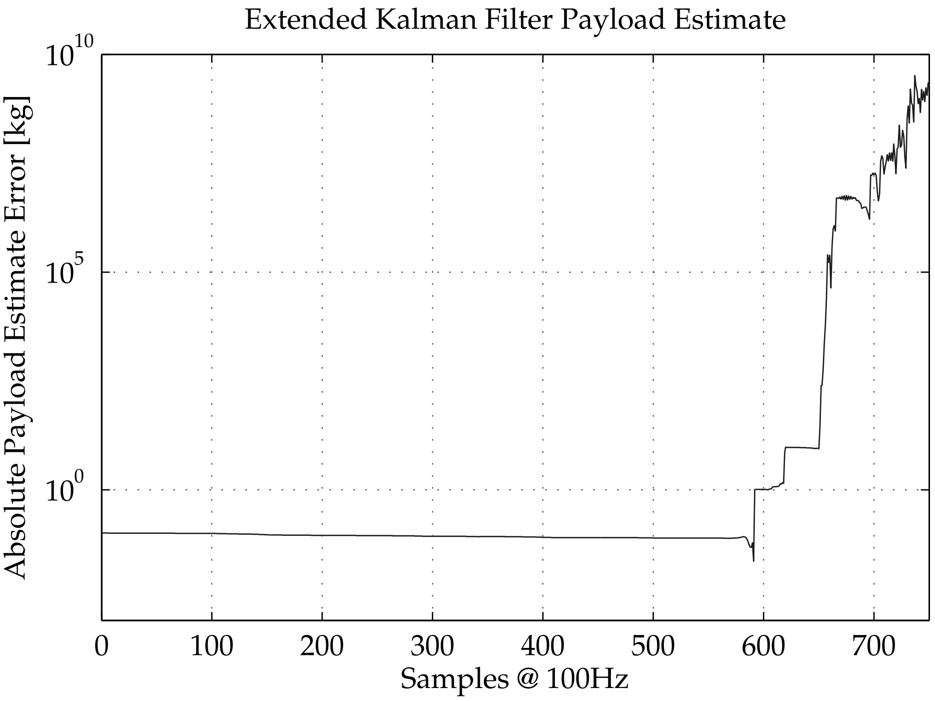


Figure 4.8.: Extended Kalman filter payload estimate error on a MERIt02 run of data with 400g payload. The estimate error can be seen to explode at roughly sample 590 (note the logarithmic scale). Robot joint angles and derivatives experience similar explosion, as well as unstable oscillatory behavior.

is more accurately approximated with a first order Taylor expansion as compared to Section 4.3.3. The measurement equations and the Jacobian are generated using the MATLAB symbolic toolbox. The the flow of measurements and estimates between the two Kalman filters through time is visualized in Figure 4.9.

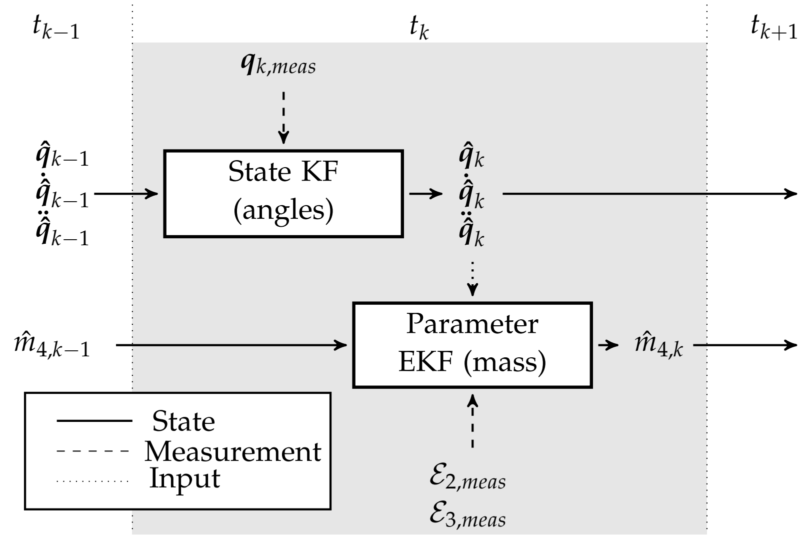


Figure 4.9.: Structure of a dual Kalman filter for joint angle and payload mass estimation. Current time step t_k, previous time step t_{k-1} and next time step t_{k+1} are shown.

As before in Sections 3.5.1 and 4.3.3, CMA-ES was used as a global optimization algorithm to find parameters for the extended Kalman payload estimator. As discussed in Section 4.3, it is desirable to have an estimator that can track through singularities, remain stable, and quickly reach a correct value. To optimize for this behavior a fitness function for use with the CMA-ES algorithm is chosen.

The fitness function divides each 80 second run of the MERIt02 dataset into 10 second chunks. The dual Kalman filter is then run across each chunk separately as well as against the entire run. The root-mean-square error for all is then evaluated as the fitness to be minimized by the CMA-ES optimization algorithm. The optimization was run for several hours with multiple restarts to find a local minimum. The MERIt02 dataset is also divided into test and training subsets with a 60-40 split. All parameter optimization was done against the training set, while the graphs and performance metrics below are against the test set. The measurement and process noise parameters found by CMA-ES for the mass estimation extended Kalman filter can be found in Equation 4.3.4.

$$\boldsymbol{P}_{mass} = \begin{bmatrix} 2.6759 \times 10^{-07} \end{bmatrix} \qquad \boldsymbol{R}_{mass} = \begin{bmatrix} 4.4930 \times 10^{5} & 0 \\ 0 & 3.0404 \times 10^{4} \end{bmatrix} \tag{4.3.4}$$

An example of the dual Kalman filter performing on a 80 second run and tracking through singularities with a 200 gram payload can be seen in Figure 4.10.

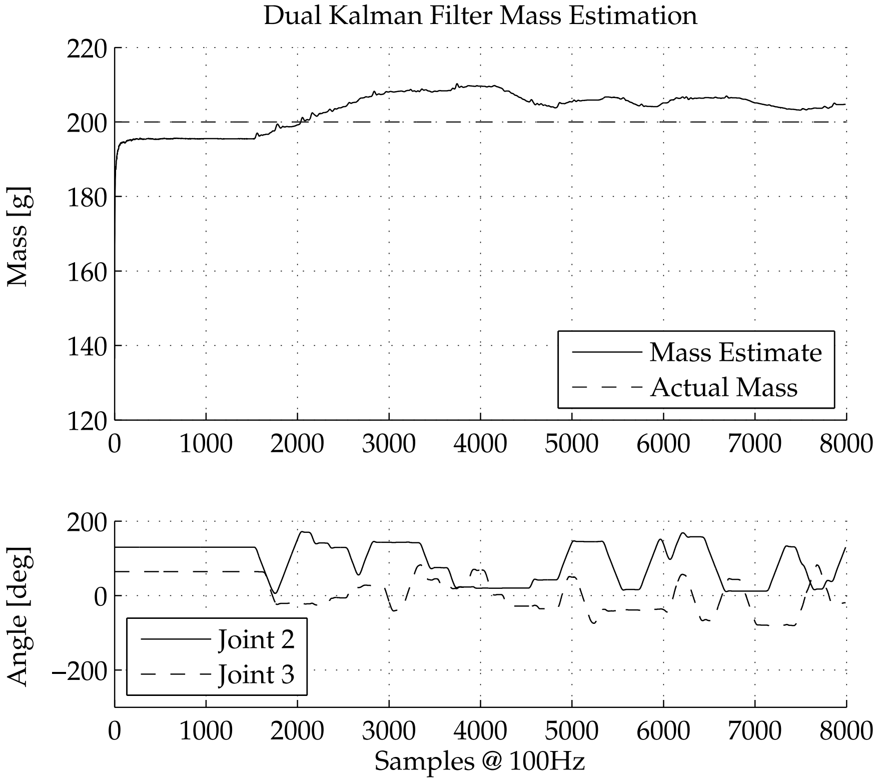


Figure 4.10.: Dual Kalman filter mass estimation with TUDOR in motion. This test was performed on an 80 second run from the MERIt02 dataset with a 200 g payload. The filter tracks the payload through singular link configurations. Joint angles during this are shown to demonstrate the robot is in motion.

The performance against the 10 second chunks can give insight as to how quickly the filter can provide an estimate of a payload, even when starting in very undesirable link configurations. The same run as shown in Figure 4.10 is shown divided into chunks in Figure 4.11 with the filter restarted for each chunk.

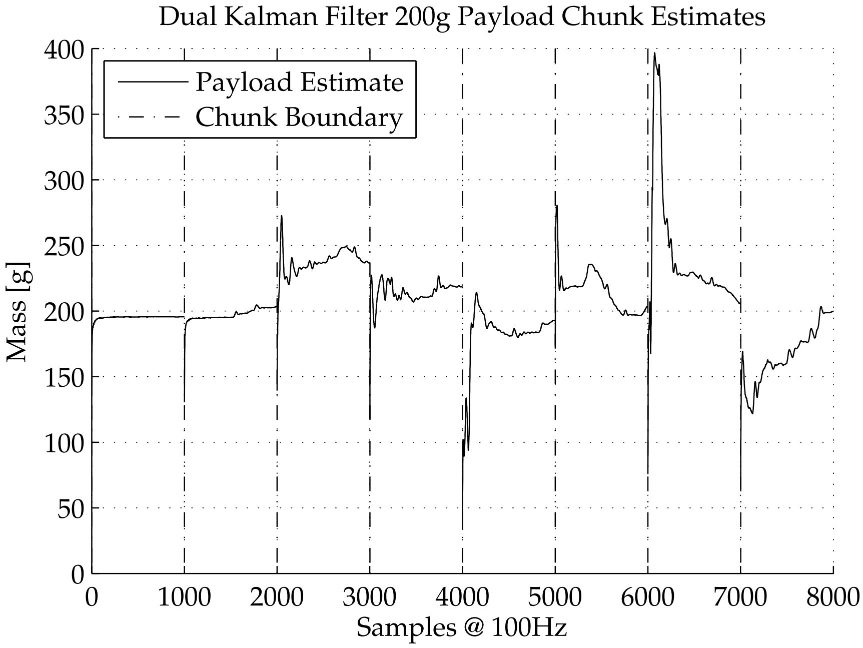


Figure 4.11.: Dual Kalman filter mass estimation with TUDOR in motion. This test was performed on 8 consecutive 10 second chunks from a run from the MERIt02 dataset with a 200 g payload. The dual Kalman filter is reset at the beginning of each chunk.

With an initial payload guess of 0 grams, the subsequent payload estimate can be viewed as a step response with the presence of disturbances. The various tools used for analyzing step responses can then give rough metrics for evaluating how quickly the dual Kalman filter is able to make a guess. This is especially useful for analyzing the 10 second chunks for each run as the Kalman filter is reset more often. The final estimate for a chunk is treated as the final value for the calculations. Rise time is calculated as the time it takes for the signal to go from 10% to 90% of the final value. Percent overshoot is calculated from the largest value reached, and the percent larger than the final value. Results split by payload as well as overall results can be found in Table 4.10.

These results suggest that after running for approximately 1 second at 100Hz, a payload estimate from the dual Kalman filter can be considered usable. The RMSE for the fitness function against the test dataset in Table 4.11 is 39.8 grams. If this is compared to the performance of the sliding window least squares estimate in Section 4.3.2 with a window size of 100 it can be seen that the dual Kalman filter approach has greatly increased performance with less than half the RMSE.

Additional tests were made with a dual Kalman filter designed for all 4 strain gages. After following the same procedures as above, the RMSE against the test dataset was found to be 39.2 grams. This is only a 1.51% improvement against the dual Kalman filter using strain gages (one on each link).

Payload [g]	Rise Time [s]	Overshoot %
100	0.65	24.28%
200	1.10	17.10%
300	1.32	19.71%
400	0.88	33.27%
Overall	0.99	23.59%

Table 4.10.: Overshoot and rise time metrics by payload and overall. Calculated from dual Kalman filter mass estimates on the 10 second chunks of the MERIt02 dataset. The final estimate of the chunk is used to calculate step response metrics.

Strain Gages Used	RMSE [g]	NRMSE
Strain 21 and 31	39.8	0.0995
All	39.2	0.0980

Table 4.11.: Payload estimate error metrics against data from MERIt02 with TUDOR in motion. Two combinations of strain gages are shown. NRMSE is normalized by the range of payloads in the dataset (0 to 400 g).

The performance of the dual extended Kalman filter with 2 strain gages is quite satisfactory and does not become unstable for any runs in the MERIt02 dataset. It meets all of the characteristics stated as being desired for dynamic payload estimation in Section 4.3. Kalman filters are iterative methods, making them much more suited for real-time embedded behavior as compared to the least squares and sliding window least squares methods covered in Sections 4.3.1 and 4.3.2.

5

Elastic Link Effector Force Estimation

In contact scenarios with elastic link robot manipulators there are two possible situations which are likely to be encountered. The first is that the forces on the effector are known and the link strains need to be predicted while in contact. The second is the inverse of this: predicting the effector forces from link strain while in contact.

In Chapter 3 it was shown that when in motion, the addition of dynamic components allow more accurate estimates of strain to be made. The dynamic components are not used in the following chapter as the motions of the robot will be relatively slow and the inertial and Coriolis effects will therefore be minimal.

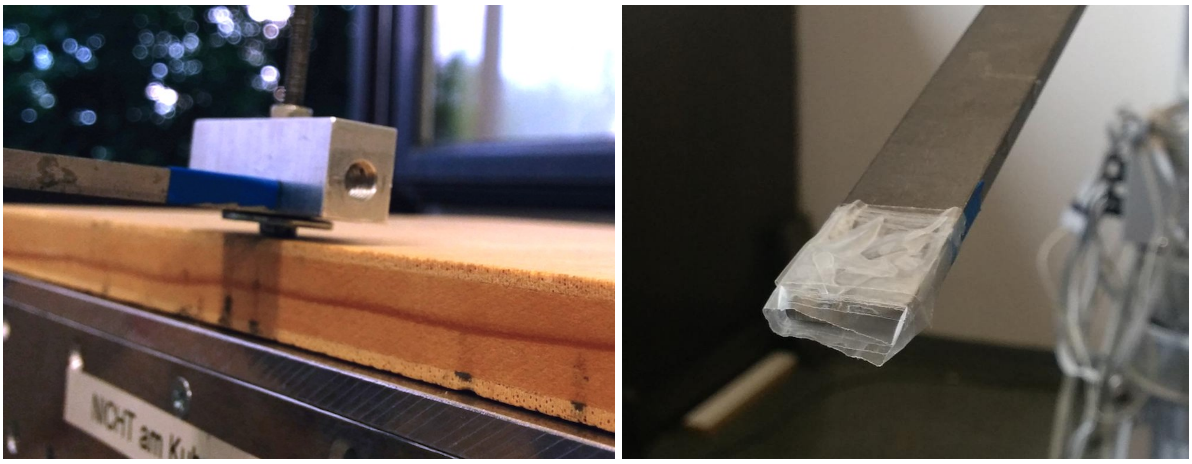
5.1. Building Contact Force Datasets

While collecting strain data with known payloads is relatively straightforward and easily reproducible, collecting strain data for known forces while in contact with the environment poses challenges. Friction effects can vary widely depending on things such as contact surface areas and temperatures. The collection of contact force data in this thesis relies heavily on the use of the force cube described in Section 1.3 to provide 6-dimensional vectors of applied force and moment information. Due to the size and geometry of this sensor system, the joint angles in which contact forces can be recorded are limited. Data recorded from this sensor must also be checked for outliers as infrequent RS-232 transmission errors cause erroneous readings to sometimes appear. This was done by looking for data larger than a certain number of times larger than the standard deviation of the force data. The two datasets recorded for the purpose of this thesis will be discussed.

5.1.1. Contact Dataset 1: Static

Contact Dataset 1 focuses on the TUDOR robot touching and applying a force to a horizontal surface while completely at rest. To focus on the most easy to sense force for the TUDOR robot, efforts were made to maintain force primarily in the effector's Y-axis, or roughly normal to the horizontal contact surface. Later in Section 5.3, it will be shown that it is always possible to estimate Y-axis force F_{ly} without any singularities. This additionally allows force cube measurements to be rotated to the effector frame with only sign changes.

With the force cube in the robot's workspace, the effector is brought close to the contact force cube (joint 2 at 138.5° and joint 3 at 41.5°). A metal bolt is attached to TUDOR's effector and used as the contact point, as shown in Figure 5.1a. To maintain forces primarily in the effector's Y-axis, the joint angles are adjusted so that the link 3 is horizontal. Joint 2 is advanced in 0.1 degree increments while joint 3 is reduced in 0.1 degree increments. After moving to the desired angle, the robot pauses 1.5 seconds to ensure any vibrations have calmed (active damping is enabled) and then 3 seconds of robot and force sensor data are recorded at 100Hz. This data is then averaged to produce a single point for the dataset. This process is continued until a desired maximum magnitude of 5 newtons of force is sensed by the external force cube. This entire process was performed 8 times for a total of 8 runs of collected data.



(a) Contact Dataset 1: Bolt

(b) Contact Dataset 2: Adhesive Tape

Figure 5.1.: Photo of TUDOR's effector during collection of the Contact Datasets. The bolt attached to TUDOR's effector for Contact Dataset 1 can be seen in Figure 5.1a contacting a plywood sheet which is resting on the metal of the contact force cube.

In total there are 722 data points in Contact Dataset 1. The contact surface was chosen as a piece of plywood which was allowed to slide freely on the metal surface of the contact cube. This was to attempt to reduce any effects of friction building up as the effector slides across the top of the force sensor to keep most force in the direction of the effector's Y-axis. The dataset is visualized in Figure 5.2.

Despite the efforts to reduce buildup of static friction in this dataset, it can still clearly be seen in the measured force in Figure 5.2. Spikes are present in both F_{lx} and F_{ly}, and F_{lx} does not increase linearly with respect to the applied joint angles.

5.1.2. Contact Dataset 2: Contact Forces in Motion

While Contact Dataset 1 focused on the robot in contact while at rest, Contact Dataset 2 captures contact force data with the robot arm moving slowly over the top of the force cube. The arm is used to apply a desired force to a surface while moving along a curved path. This is a motion that might be seen in real world applications such as applying a polishing tool to a surface, or drawing an arc with a pen.

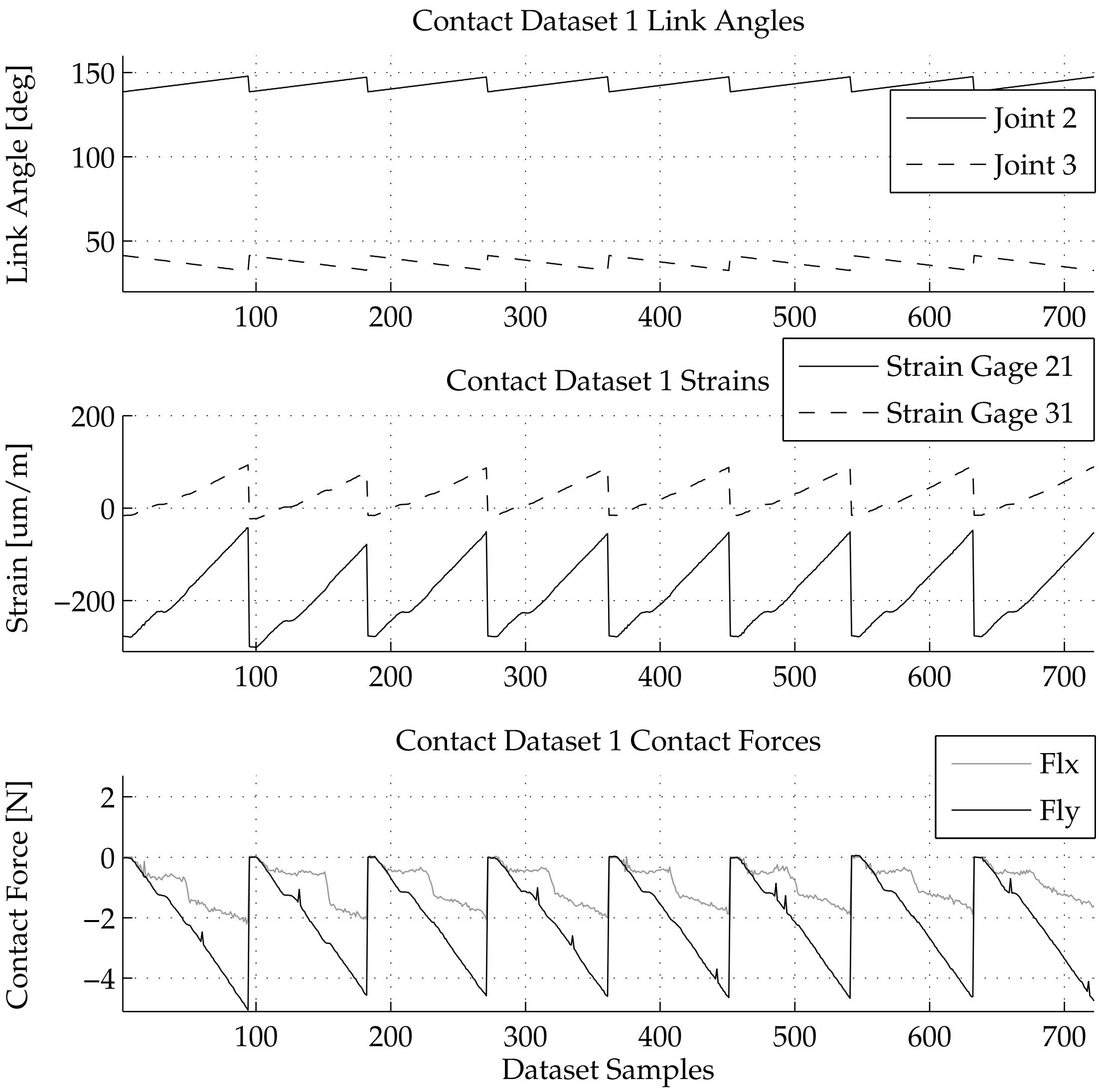


Figure 5.2.: Visualization of Contact Dataset 1. Link angles, strains and measured force from the contact cube are shown.

As this motion is applied over a few seconds, as compared to the minutes of Contact Dataset 1, it is desired to minimize any contact friction effects to improve repeatability of the collected data. The effector of TUDOR was wrapped in adhesive tape with a slippery external surface as seen in Figure 5.1b.

To collect data, joints 2 and 3 are moved so that link 3 is maintained roughly horizontal to the ground so that force cube forces can be rotated to the effector frame easily with only sign changes and to minimize friction. The effector is then started moving to a desired force goal by moving joints 2 and 3. After a short delay, joint 1 is moved to circumscribe a curved path. It should be noted that once the initial force goal is met, no effort is made to regulate it and joints 2 and 3 are merely stopped.

The motion was applied 10 times, with the force goals of 0.2, 0.4, 1, 2, and 3 newtons, for a total of 50 runs. During each of these runs, data is recorded at 100Hz, making for a total dataset length of 36,000 samples. The force measured from the force cube, rotated into the effector frame, for a run with a 3 newton force goal can be seen in Figure 5.3. It should be noted that while this figure contains F_{lz}, because these readings come from the force cube, F_{lz} cannot be determined by the strain gages on TUDOR due to their configuration.

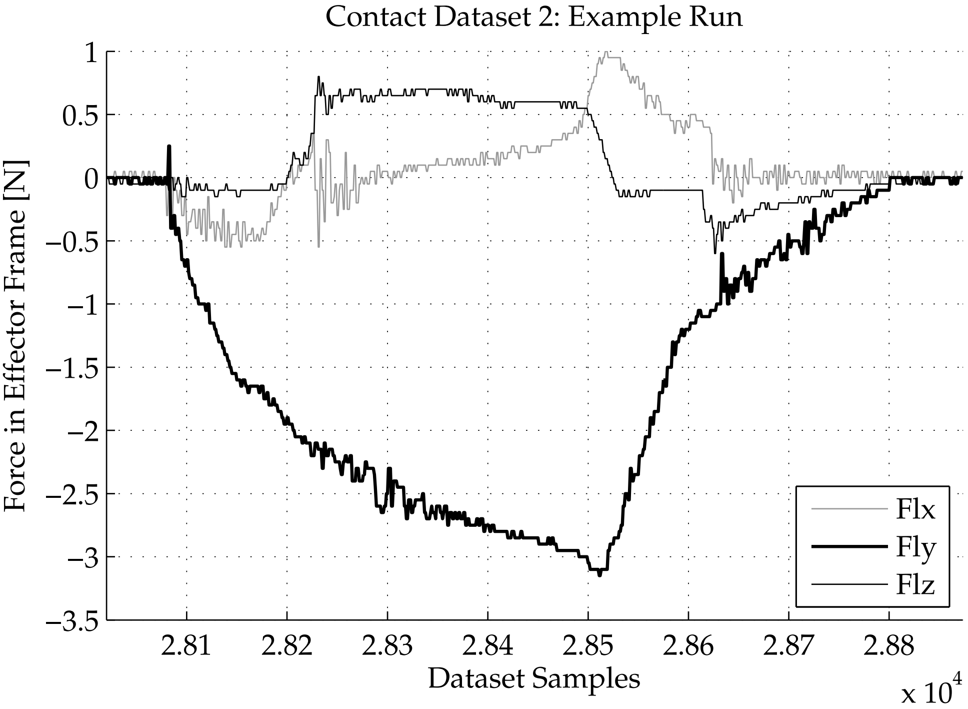


Figure 5.3.: Visualization of force cube readings during a run from Contact Dataset 2. The arm was moved until 3 newtons of force in the effector Y-axis F_{ly} (Z-axis of the force cube) was reached while circumscribing a circular path using joint 1. The arm was then moved away until contact was broken.

This dataset contains a wide variety of phenomena. Things such as static friction being broken and kinetic sliding friction can be seen as the arm rotates and increases force on the top of the force cube. This wider variety of situations provides a more general benchmark (as compared to Contact Dataset 1) by which to measure the performance of force estimates.

5.2. Results of Predicting Strain During Contact

As in Chapter 2, if the joint angles, payload, and contact forces are known, then the strain for each strain gage can be predicted using Equations 2.7.8. As the structure of these equations was modified such that the payload mass and contact forces appear in the same regressor terms, the models that are linear in parameters trained against MERIt03 static payload data (found in Tables 2.3 and 2.5) can be used to estimate strain during effector contact.

This can be shown in the results found in Table 5.1 for Contact Datasets 1 and 2. The results are quite accurate considering this model was never trained on any datasets containing force data, showing this to be a quite effective method. Additionally, from Table 5.1 it can be seen that the performance on Static Datasets 1 and 2 is quite similar. It can be concluded that the quasi-static assumptions used in this section for predicting strain works for the robot both at rest and moving slowly.

	Contact Dataset 1		Contact Dataset 2	
Strain Gage	RMSE [$\mu m/m$]	NRMSE	RMSE [$\mu m/m$]	NRMSE
21	9.85	0.0381	4.82	0.0354
22	14.15	0.0616	5.42	0.0497
31	3.99	0.0343	2.45	0.0300
32	2.94	0.0540	2.39	0.0504

Table 5.1.: Contact strain estimation error metrics against Contact Datasets 1 and 2. NRMSE is normalized by the range of values seen in the dataset for each strain gage.

5.3. Results of Estimating Force During Contact

Equations 2.7.8a and 2.7.8b for link 2 and 3 static strain are put into the form seen in Equation 5.3.1. This system can be solved analytically to produce equations for determining effector forces F_{lx} and F_{ly}. This is shown in Equations 5.3.2 where $R_{i,j}$ refers to the static linear regression parameter j for link i.

$$\begin{bmatrix} \mathcal{E}_2 - E \\ \mathcal{E}_3 - F \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} F_{lx} \\ F_{ly} \end{bmatrix} \tag{5.3.1}$$

$$F_{lx} = \frac{R_{2,1} - \mathcal{E}_2 + (R_{2,5} + m_4 R_{2,4}) g \cos q_2 + (r2_3 + m_4 R_{2,2}) g \cos(q_2 + q_3)}{R_{2,4} \sin q_3} - \frac{(R_{2,2} + R_{2,4} \cos q_3)(R_{3,1} - \mathcal{E}_3 + (R_{3,3} + m_4 R_{3,2}) g \cos(q_2 + q_3))}{R_{2,4} R_{3,2} \sin q_3} \tag{5.3.2a}$$

$$F_{ly} = \frac{R_{3,1} - \mathcal{E}_3 + (R_{3,3} + m_4 R_{3,2}) g \cos(q_2 + q_3)}{R_{3,2}} \tag{5.3.2b}$$

Due to the configuration of TUDOR, it is not possible to estimate force in the Z-axis of TUDOR's effector. This can be seen simply by the fact that if a force is applied in the Z direction, it will not cause a strain in any of the strain gages. As the contact force estimates need only be performed in a restricted workspace where the force cube can be used, less care will be taken to avoid singularities than in Chapter 4 during payload estimation. From Equation 5.3.2b it can be seen that there is no singularity in estimates of F_{ly}. For estimates of F_{lx}, the sine of the angle of joint 3 must not be 0 to avoid singularities. This implies that the robot arm cannot be fully extended to be able to sense forces in the X axis of the effector as the strain gages on link 2 will experience no strain.

Equations 5.3.2 are used on Contact Datasets 1 and 2 with strain gages 21 and 31 to provide estimates of the force experienced on the tip of TUDOR's effector. This is then compared to the actual values measured with the contact force cube. The results are summarized in Table 5.2 and shown in Figures 5.4 and 5.5.

	Contact Dataset 1		Contact Dataset 2	
Effector Force	RMSE [N]	NRMSE	RMSE [N]	NRMSE
F_{lx}	0.169	0.0747	0.219	0.1325
F_{ly}	0.177	0.0346	0.108	0.0271

Table 5.2.: Contact force estimation error metrics against Contact Datasets 1 and 2, using strain gages 21 and 31. NRMSE is normalized by the range of values seen in the dataset for force in each effector axis.

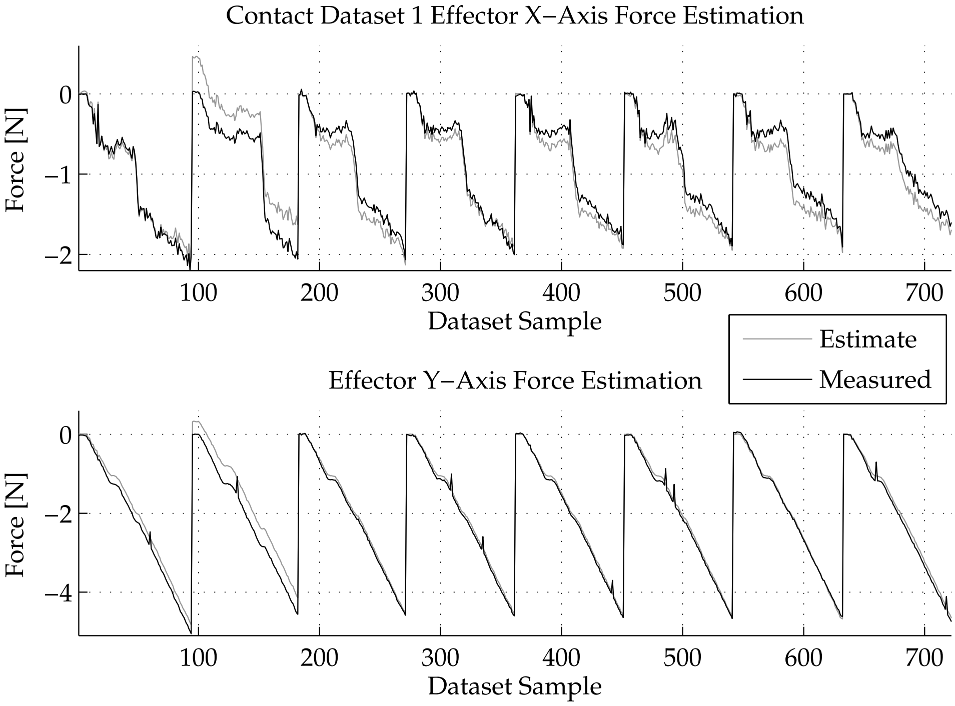


Figure 5.4.: Effector contact force estimates against Contact Dataset 1. Actual measurements and estimates are shown across the entire dataset for forces in both the X and Y axes of the effector frame.

This demonstrates that under controlled circumstances, quite accurate estimates of effector force can be made for a flexible link robot. The estimates for F_{lx} are consistently worse (more than double the NRMSE) than those for F_{ly}, which is likely due to F_{lx} not being able to be sensed by link 3, making reliance upon the noisier strain gages in link 2 necessary. This reliance on link 2 is also limited by the singularity in the estimates of F_{lx} as the sine of the angle of joint 3 approaches zero.

This also demonstrates the ability to make effector force estimations with the robot at rest and while moving slowly. The addition of dynamic terms such as Coriolis and inertia as in Chapter 3 are not necessary as long as motion is sufficiently slow.

These results were deemed sufficiently accurate for the application of various force control algorithms. Techniques such as sensor fusion and dual Kalman filtering could potentially be employed to increase accuracy of these results.

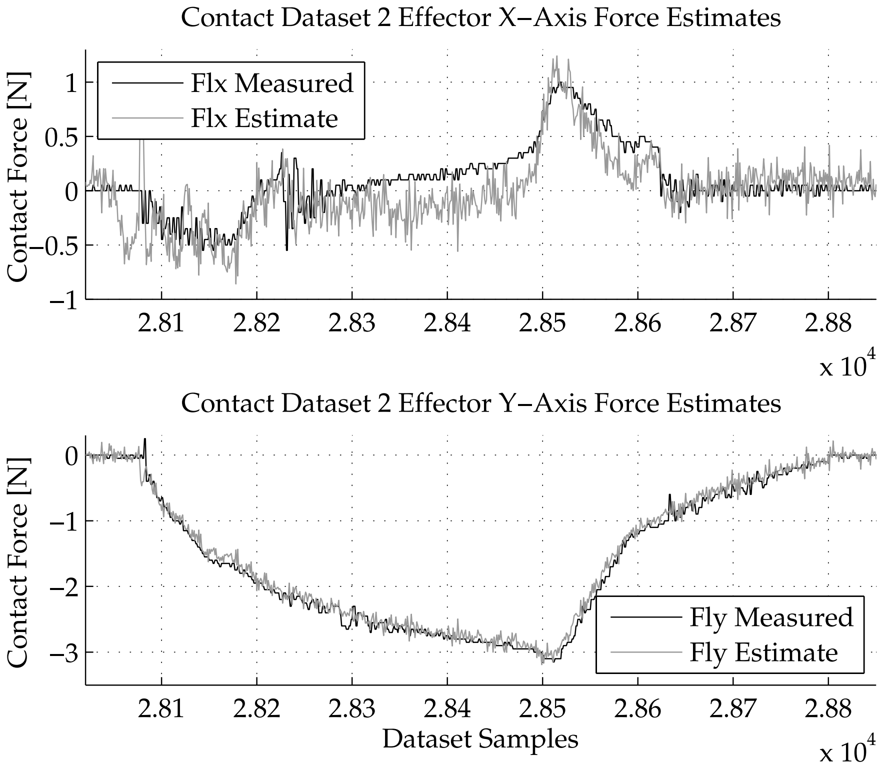


Figure 5.5.: Effector contact force estimates against a run from Contact Dataset 2 with a force target of 3 newtons.

6

Elastic Link Force Control

In this chapter the basic types of force control in robotics will be presented. Two force controllers will then be designed and discussed. These force controllers will then be applied to 4 different force control tasks and experimental results with TUDOR will be reviewed.

6.1. Indirect and Direct Force Control

Force controllers are often categorized as either indirect or direct. Indirect force control contains methods related to impedance or stiffness control. These methods usually employ motion control and do not provide a closed force-feedback loop. They typically are used ensure an upper limit to contact force during movement by making assumptions about the maximum environmental stiffness (Siciliano et al. 2000).

Direct force control methods can be used to apply exact contact forces, not just keep them below certain limits. Direct force control is often implemented using a closed force-feedback loop, operating on the error between measured force and desired force. Typically the force control loop is closed around an internal motion control loop such as a velocity or position controller (Siciliano et al. 1999). While previous work with TUDOR has involved indirect force control (Malzahn et al. 2014c), this thesis focuses primarily on the application of direct force control, closing the force-feedback loop around TUDOR's joint velocity controller.

6.2. Designing Force Controllers

As mentioned in Section 1.3.1, elastic link manipulators can be viewed in terms of slow and fast subsystems. The slow subsystems of many elastic link models are similar in structure to the rigid link robot dynamics, while the fast subsystems capture the behavior of the vibration in the elastic links. Using the damping algorithm described in Section 1.3.1 allows simple robotic controllers for the slow subsystem to be closed around a faster vibration damping control loop (Matsuno et al. 1994; Siciliano et al. 1988). Traditional proportional techniques for force control of rigid link robots from Gorinevsky et al. (1997) will be used with in designing force controllers for TUDOR. Methods developed in Chapter 5 for estimating effector contact force from strain will be used with these controllers.

6.2.1. Three Dimensional PI Manipulator Force Controller

In Gorinevsky et al. (1997), a proportional force controller for one degree of motion manipulators driven by velocity servos is presented. In this section, this controller is extended to work with a 3 degree-of-freedom serial link robot to apply force in 3 dimensions. This controller will also be extended to include integral components. Treatment of the stability of this controller in its original form with contact in various situations can be found in Gorinevsky et al. (1997).

Estimates for tip force in the effector frame are found as $\hat{f}_{lx}$ and $\hat{f}_{ly}$ using Equations 5.3.2, whose derivation is detailed in Section 5.3 ($\hat{f}_{lz}$ cannot be sensed with TUDOR and is therefore set to zero here). While effector frame related forces were useful previously, for performance of actual tasks, a robot must typically operate in a workspace or global coordinate frame. The effector frame forces are rotated to match the world coordinate frame in the form of ${}^{0}\hat{f}_{lx}$, ${}^{0}\hat{f}_{ly}$ and ${}^{0}\hat{f}_{lz}$ as shown in Equation 6.2.1. Where ${}^{0}_{E}\boldsymbol{R}$ is the rotation transform between the effector frame and global coordinates. It should be noted that the force-moment transform used previously in Section 2.6.3 is not needed as only the rotation is changing and not the point of interest.

$$\begin{bmatrix} {}^{0}\hat{f}_{lx} \\ {}^{0}\hat{f}_{ly} \\ {}^{0}\hat{f}_{lz} \end{bmatrix} = {}^{0}_{E}\boldsymbol{R} \begin{bmatrix} \hat{f}_{lx} \\ \hat{f}_{ly} \\ 0 \end{bmatrix} \tag{6.2.1}$$

A PI velocity controller can then be designed for force in each of the 3 Cartesian axes of the world coordinate frame at the point of contact. The desired force in the world frame, referred to as the reference signal vector $\boldsymbol{r}(t_k)$, is defined as in Equation 6.2.2 from ${}^{0}f_{dx}$, ${}^{0}f_{dy}$, and ${}^{0}f_{dz}$. The error vector $\boldsymbol{e}(t_k)$, at discrete time step t_k, is defined as the difference between the reference signal and the estimated forces in global coordinates as in Equation 6.2.2.

$$\boldsymbol{r}(t_k) = \begin{bmatrix} {}^{0}f_{dx} \\ {}^{0}f_{dy} \\ {}^{0}f_{dz} \end{bmatrix} \qquad \boldsymbol{e}(t_k) = \begin{bmatrix} {}^{0}\hat{f}_{lx} \\ {}^{0}\hat{f}_{ly} \\ {}^{0}\hat{f}_{lz} \end{bmatrix} - \begin{bmatrix} {}^{0}f_{dx} \\ {}^{0}f_{dy} \\ {}^{0}f_{dz} \end{bmatrix} \tag{6.2.2}$$

The PI controller with an output vector ${}^{0}\boldsymbol{u}_v(t_k)$ of velocity commands in the global coordinate frame is designed as shown in Equation 6.2.3. The proportional and integral gain vectors for each Cartesian axis of the global frame are represented as ${}^{0}\boldsymbol{k}_p$ and ${}^{0}\boldsymbol{k}_i$ respectively.

$${}^{0}\boldsymbol{u}_v(t_k) = \begin{bmatrix} {}^{0}v_x \\ {}^{0}v_y \\ {}^{0}v_z \end{bmatrix} = {}^{0}\boldsymbol{k}_p \boldsymbol{e}(t_k) + {}^{0}\boldsymbol{k}_i \int_0^{t_k} \boldsymbol{e}(\tau) d\tau \tag{6.2.3}$$

As covered in Gorinevsky et al. (1997), proportional controllers are usually sufficient for many force control tasks. Integration terms were not used frequently in this paper, but are included for completeness. Derivative terms were not included in this controller as the rather noisy estimates of effector forces would likely cause stability issues for the controller.

For use with a robot, the desired velocities ${}^{0}\boldsymbol{u}_v(t_k)$, in global coordinate frame must be made into a 6×1 general velocity vector ${}^{0}\boldsymbol{v}_d(t_k)$ as in Equation 6.2.4. This can then finally be converted to the necessary robot joint velocities in generalized coordinates $\dot{\boldsymbol{q}}_d(t_k)$ using the robot Jacobian, as seen in Equation 6.2.5.

$$ {}^{0}\boldsymbol{v}_d(t_k) = \begin{bmatrix} {}^{0}v_x & {}^{0}v_y & {}^{0}v_z & 0 & 0 & 0 \end{bmatrix}^T \tag{6.2.4} $$

$$ \dot{\boldsymbol{q}}_d(t_k) = {}^{0}\boldsymbol{J}(q)\, {}^{0}\boldsymbol{v}_d(t_k) \tag{6.2.5} $$

The desired joint velocities $\dot{\boldsymbol{q}}_d(t_k)$ are then given to the PI velocity servo controllers of the robot which, for TUDOR, are described in Malzahn et al. (2011).

During the course of this thesis, this controller is used mainly to apply forces normal to the top surface of the force cube, but could be easily used to apply force in any direction with ease by giving an appropriate reference signal $\boldsymbol{r}(t_k)$.

Notes on Safety and Stability

In general, an application of a force in the global z-axis to the force cube will be in a position in which the singularities discussed in Section 5.3 for force estimation will not occur. Special care should be taken to avoid singularities as force will become unobservable close to these singularities. In TUDOR's case, this could occur when trying to apply a force to a vertical surface at nearly the full extension of the arm. Additionally, as TUDOR cannot sense forces in the effector z-axis, this controller will not work for tasks that require regulation of force in this direction. Outside of a controlled experimental setting, further extensive mathematical evaluation and empirical testing should be done to ensure safe operation.

Additionally, during a software crash while using velocity controlled servos, the robot may become stuck in a state where it will continue to move at a fixed velocity, despite force applied to the environment. While the dangers are lessened when using an elastic link robot, they should still be taken very seriously. Implementation of safety systems such as watchdogs, fail-safes, and emergency stops in any environment are absolutely necessary.

Eppinger et al. (1987) and later Gorinevsky et al. (1997) have discussed joint and link compliance, gear backlash, and friction as a examples of possible sources for stability problems in force control. Chiou et al. (1988) demonstrates that link flexibility does contribute to force control instability, something that must often be resolved through less aggressive control strategies to maintain stability. During the experiments to follow, the maximum velocity of the robot joints is severely limited for safety reasons. This limits the actuation bandwidth and will be shown to negatively effect force control performance, something that has been shown to already be very bandwidth restricted (Bazaei et al. 2011; Chiou et al. 1988; Li 1990).

6.2.2. Two Dimensional Contour Force Controller

As TUDOR is only able to sense force in two dimensions of the effector frame, any contour following with force control can only occur in two dimensions. The proportional controller presented in Gorinevsky et al. (1997) for the control of a two degree-of-freedom Cartesian robot is adapted to the serial link robot TUDOR and extended with an integral term. Treatment of the stability of this controller in its original form with contact to various surfaces can be found in Gorinevsky et al. (1997).

For a point on a known 2 dimensional contour, the 2 × 1 vector normal to the surface is referred to as $\boldsymbol{n}$ and the vector tangent to the surface is referred to as $\boldsymbol{\tau}$. The controller for moving along a 2 dimensional contour is given the references F_{dn} and $v_{d\tau}$, a desired force tangent to the contour and a desired velocity normal to the contour respectively. The normal force error e_n is then defined as in Equation 6.2.6 from the difference between the reference normal contour force F_{dn} and the actual force normal to the contour F_n.

$$e_n(t_k) = F_n(t_k) - F_{dn}(t_k) \tag{6.2.6}$$

The controller velocity command vector in the global frame ${}^0\boldsymbol{u_v}$ is composed of two components ${}^0\boldsymbol{u_n}$ and ${}^0\boldsymbol{u_\tau}$, the commanded velocity normal to and tangent to the contour surface respectively. These are calculated as shown in Equation 6.2.7 with proportional and integral gains k_{pn} and k_{in}.

$${}^0\boldsymbol{u_v} = {}^0\boldsymbol{u_n} + {}^0\boldsymbol{u_\tau} = (k_{pn}e_n + k_{in}\int_0^{t_k} e_n(\lambda)d\lambda)\boldsymbol{n} + v_{d\tau}\boldsymbol{\tau} \tag{6.2.7}$$

If the contour tangent and normal vectors $\boldsymbol{\tau}$ and $\boldsymbol{n}$ are known, the velocity commands from Equation 6.2.7 can be treated as in Section 6.2.1 to find joint velocities to produce the desired global frame velocities. In most real-world situations, the contour vectors are not known.

When the effector is in contact with the frictionless surface, the direction of the effector force relative to the surface is the normal vector. Again, in the real world, the presence of friction necessitates that there be some angle β between the frictionless estimate of normal vector $\boldsymbol{n}$ and the actual value. Taking advice from Gorinevsky et al. (1997), a constant guess of γ can be made for the angle β and, as long as the discrepancy between γ and β is not too large, stability can be maintained. Gorinevsky et al. (1997) demonstrated that even for a large discrepancy, this controller will remain stable. The guess with angle γ results in contour normal vector $\boldsymbol{n}^*$ and tangent vector $\boldsymbol{\tau}^*$ which are substituted in Equation 6.2.7 for $\boldsymbol{n}$ and $\boldsymbol{\tau}$. This can be visualized as in Figure 6.1, adapted from Gorinevsky et al. (1997). The angle β is often referred to as the angle of the friction cone.

This controller is designed for use only when in contact with a surface. If no contact is present, then the estimate of the normal and tangent of the surface will be meaningless. This controller can be combined with the three dimensional proportional manipulator force controller from Section 6.2.1 with a switching behavior for actual applications. Noisy guesses for the magnitude and direction of the force shown in Figure 6.1 as $\boldsymbol{F_c}$ will in turn lead to noisy velocity commands.

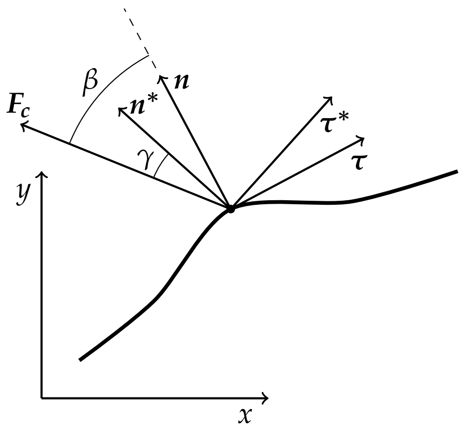


Figure 6.1.: Adaptation of a diagram from Gorinevsky et al. (1997). Real contour normal and tangent vectors ($\boldsymbol{n}$ and $\boldsymbol{\tau}$) are shown in relation to the contact force $\boldsymbol{F_c}$. The angle β between $\boldsymbol{F_c}$ and $\boldsymbol{n}$ is estimated by angle γ to produce estimates of the contour normal and tangent as $\boldsymbol{n}^*$ and $\boldsymbol{\tau}^*$.

6.3. Force Control Task Experiments

6.3.1. Applying a Constant Force

Application of a constant force in one position is one of the most basic force control problems. It includes the tasks of approaching a surface when not in contact and regulating the force applied once contact is made. Real-world applications of this include tasks such as applying a polishing tool to a single point, maintaining pressure on a wound, or feeling for an unknown obstacle. The 3D manipulator force controller described in Section 6.2.1 is applied to the task of applying a desired force in the global z axis to the top of the contact force cube.

The problem of approaching an unknown object in the environment or contact transition is often regarded as important in force control for rigid link robot arms (Gorinevsky et al. 1997). The need for contact transition control with elastic link robots is overall much less as the links are more compliant, giving more time to react before large forces can be generated. In the results shown below, the arm will start not in contact and approach the environment.

Comparison of Various PI Gains with the 3D Manipulator Force Controller

With TUDOR's effector wrapped in foam, a reference step of -0.5 N of force in the global z-axis (vertical) was given with various proportional and integral gains to the manipulator force controller. The results can be seen in Figure 6.2. The leftmost plot shows various values for the proportional gain while the integral gain is zero. The rightmost plot shows various values for the integral gain, while the proportional gain is 0.1.

It can be seen that for a proportional gain K_{pz} of 0.01, and no integral gain, it takes approximately 2250 samples or 22.5 seconds to reach the desired force. With proportional gains of 0.1 and 0.4 the robot initially reaches the desired force at approximately 2 and 1.1 seconds respectively. With a proportional gain of 0.4, a significant overshoot

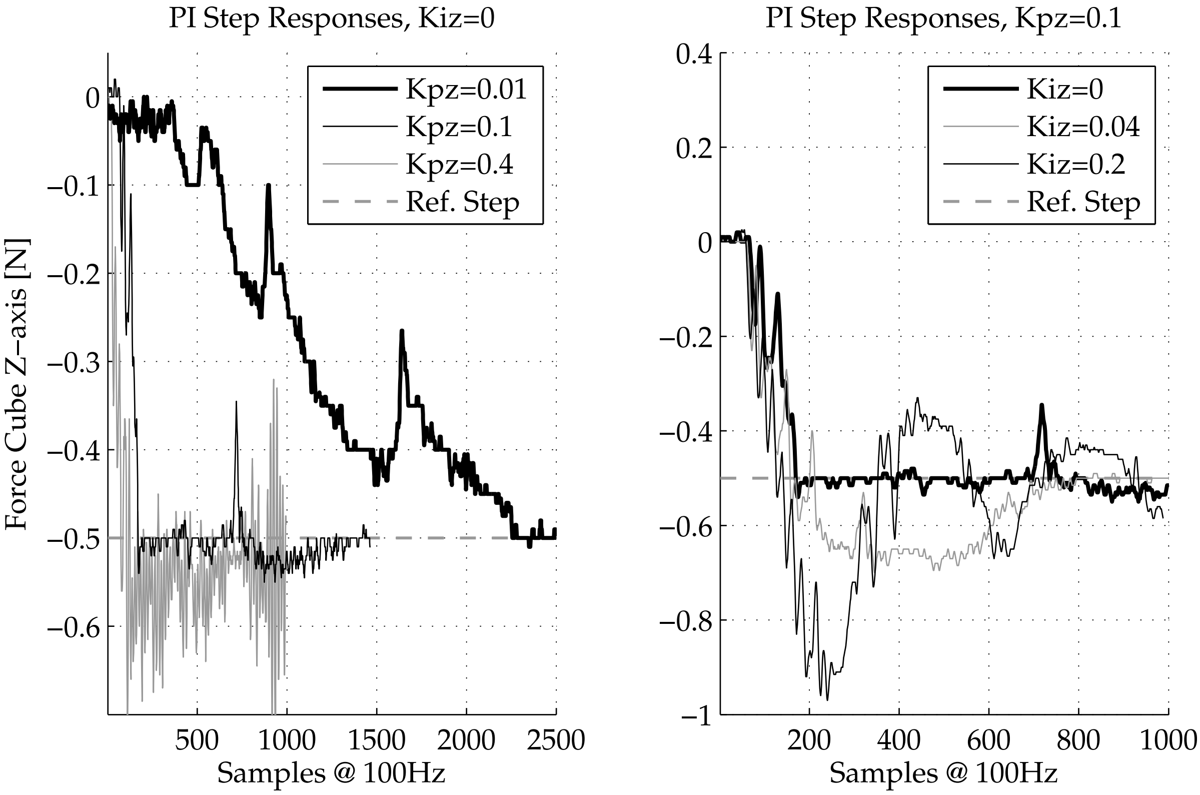


Figure 6.2.: Comparison of manipulator force controller step responses with different integration gains and a proportional gains. An input step reference signal of 0.5 N in the global Z axis is given at sample 1. TUDOR's effector is wrapped in foam and starts freely above the surface of the contact force cube. As this data is quite noisy, it has been filtered with a 5 point moving average to aid plot visibility.

is seen, as well as a amplification and oscillation due to noise in the force estimates.

With a proportional gain K_{pz} of 0.1, the effects of various integral gains can be seen in the rightmost graph of Figure 6.2. They can be seen to not provide any significant advantage, increasing the overshoot and settling time for even small values. In the remaining experiments, an integral gain will not be used as it did not empirically show any advantage.

While the results of these gains are informative, it should be noted that the contact surface plays a very large role in the performance of the force controllers. For different surfaces applied to TUDOR's effector and different workpiece surfaces, different gains may need to be used.

6.3.2. Moving Across a Flat Surface With Force Control

Moving a point of contact across a flat surface while applying a constant force is a simple, but useful type of force control task. This could be used in a variety of applications such as drawing a line with a delicate pen, grinding or polishing an edge, or cutting with a sharp tool. The 2D contour force controller from Section 6.2.2 was tested on this task by controlling the force on the bare top surface of the force cube, moving as illustrated in Figure 6.3.

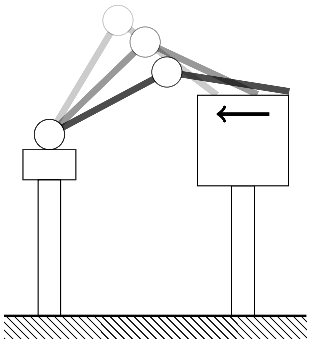


Figure 6.3.: Diagram showing desired arm motion on the contact force cube after contact is made. Direction of motion is indicated.

The controller was configured to apply a desired normal force of 0.7 N while remaining in motion, using a normal force proportional gain of 0.5. In this test, the robot arm does not start in contact to the cube, which presents problems for the contour force controller in estimating a surface normal. To overcome this, a proportional 3D force controller from Section 6.2.1 is configured to apply 0.7 N normal to the cube. After this goal is reached, the 3D force controller is turned off and the contour controller is turned on.

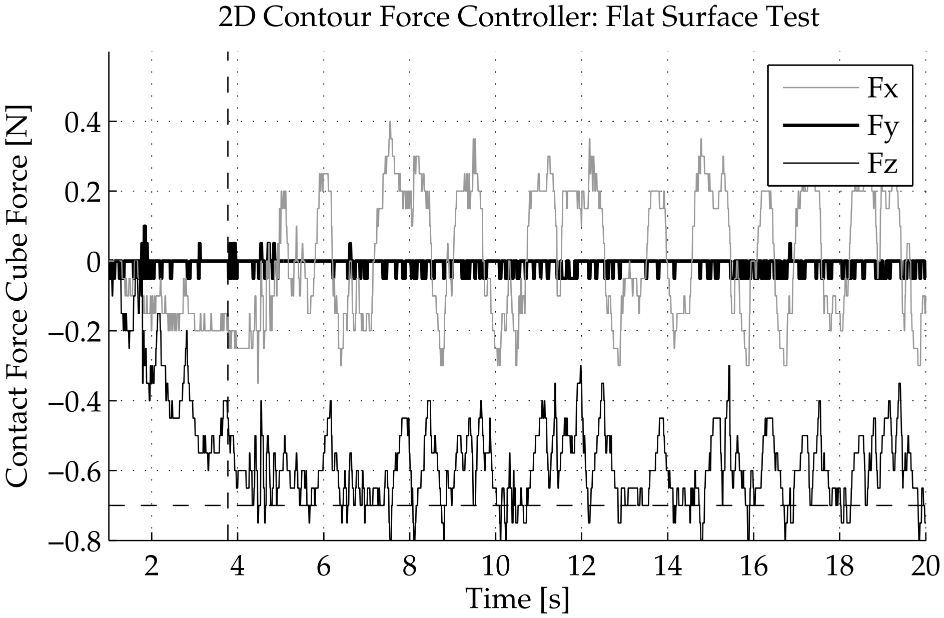


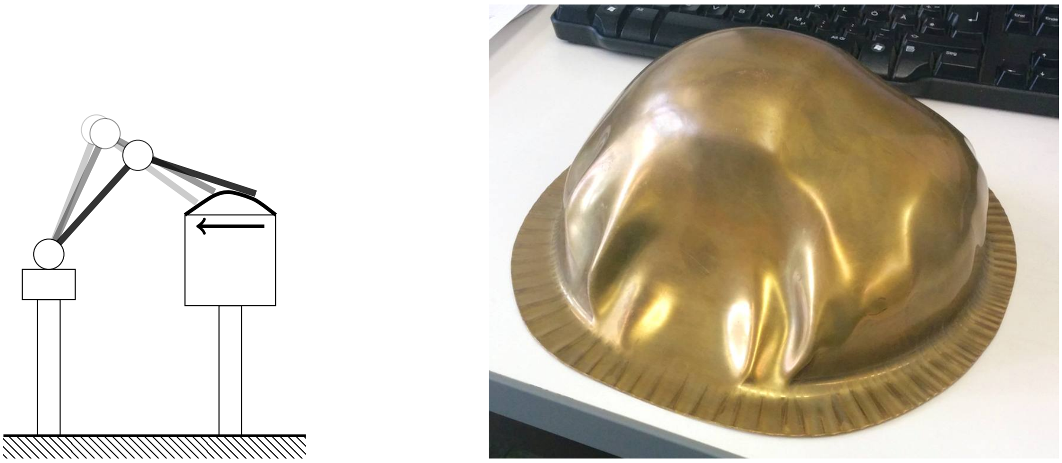
Figure 6.4.: Application of the 2D contour force controller to the top metal surface of the force cube for 20 seconds. Force measurements from the force cube are shown. The vertical dashed line shows the time when the 2D contour controller is enabled. The horizontal dashed line shows the reference signal.

The controller performed satisfactorily and the force readings applied to the top of the force cube can be seen in Figure 6.4. The switching time between the 3D and contour force controllers can be seen as the dashed line at 3.77 seconds. The force in the z-axis, normal to the surface, can be seen to be kept within 0.3 N of the target of 0.7 N.

After the contour force controller is enabled, F_x and F_z can be seen to begin oscillating. This is caused by the effector repeatedly breaking static friction as it moves slowly across the surface. If allowed to move faster, smoother motion would likely be possible. Despite difficult conditions due to friction effects, this controller behaves well at this task.

6.3.3. Moving Across a Curved Surface With Force Control

The task of moving across a curved surface is a more complicated version of the task seen in Section 6.3.2 with a flat surface. The real-world applications are similar, but are no longer limited by the requirement of flat surfaces. This greatly expands the utility of being able to perform such a task. The curved surface used during this test was the vacuum formed brass shape shown in the photograph in Figure 6.5b and the desired motion is illustrated in Figure 6.5a.



(a) Desired arm motion over the contour surface. Direction of motion is indicated.

(b) Vacuum formed brass surface for use with force control tests.

3D Force Controller Application

If the surface is not overly curved, and the requirements for the applied force only specify control of force in a static direction, the 3D force controller from Section 6.2.1 can be applied. Overly curved, in this context, meaning the surface tangent is too far from horizontal. Joints 2 and 3 are controlled by the 3D force controller, with a fixed force target towards the ground. Meanwhile, joint 1 is moved at constant velocity. With the effector in contact with the surface, and joints 2 and 3 regulating the force, the desired behavior can be produced. This assumes joint 1 moves sufficiently slowly for joints 2 and 3 to have time to move with the angle of the surface.

This controller, with a proportional gain of 0.075, was applied with a force goal of -0.3 N in the z-axis of the global frame. The curved metal surface was attached to the top of the force cube and a metal bolt similar the one in Figure 5.1a was attached to the tip of TUDOR's effector. The measured contact cube force during 18 seconds of force control and 5 seconds of joint 1 movement can be seen in Figure 6.6.

The controller can be seen to easily meet the target force goal of 0.3 N of force towards the ground while the effector is at rest. Once the effector begins moving at

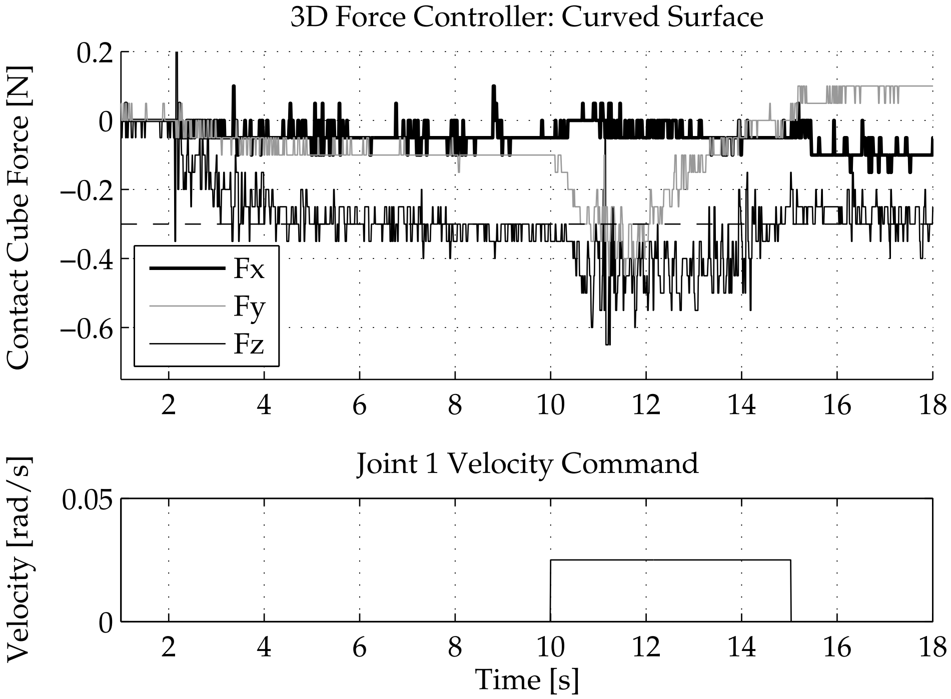


Figure 6.6.: Application of the 3D force controller to a curved metal surface on the force cube for 18 seconds. Force measurements from the contact force cube are shown. Joint 1 velocity is shown to indicate when the effector is being moved across the surface. The controller is configured with a proportional gain of 0.075 and given a reference of -0.3 N in F_z (indicated by the horizontal dashed line).

10 seconds, the force level oscillates some, possibly due to friction, and the controller can be seen to struggle trying to maintain the desired force. Considering the actuator bandwidth limitations put in place for safety during these experiments, the controller has trouble moving fast enough to match the friction forces of the slipping effector. This test was additionally only performed in a 10 cm range of the curved surface. Any attempt to use more sharply curved regions resulted in the effector losing contact. Given the circumstances, this controller performs better than expected, but it will next be shown the contour controller from Section 6.2.2 performs much better.

Contour Force Controller Application

The application of the 2D contour force controller from Section 6.2.2 allows a force normal to a curved surface as well as a speed tangent to that surface to be controlled. Unlike the 3D force controller, there are no requirements for how flat the surface must be. This controller only works while in contact with an object as it needs to estimate the surface normal. For this experiment, a 3D force control is used initially when not in contact. It is configured to apply force towards the ground when no contact is detected. Once contact is detected, it is switched off and the contour controller is switched on.

A target normal force of 0.8 N was configured and the robot arm was positioned in a starting position above the curved metal surface. The arm was then allowed to move for 20 seconds. Measurements from the contact force cube during this duration can be seen in Figure 6.7 as the magnitude of total force on the cube.

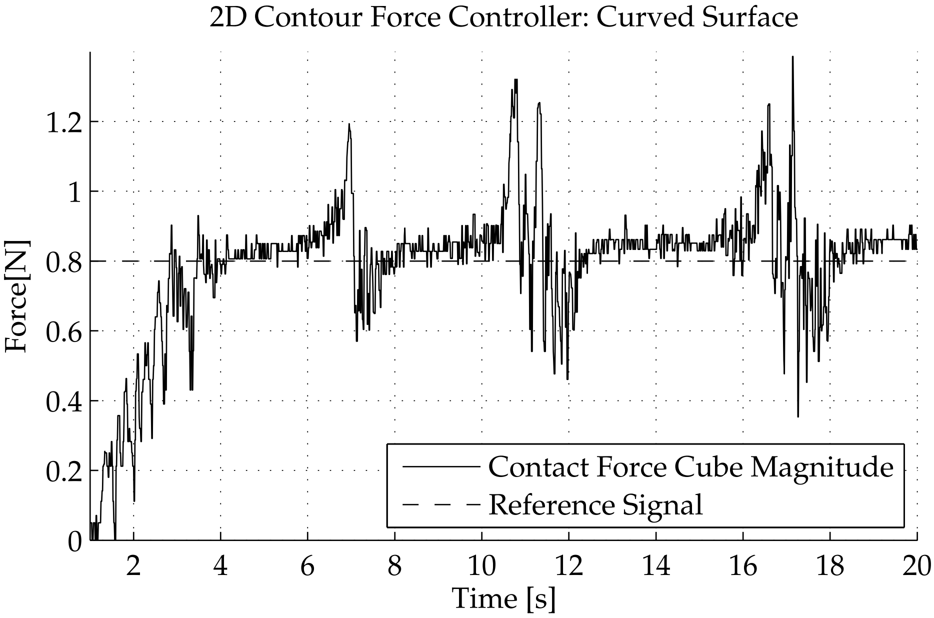


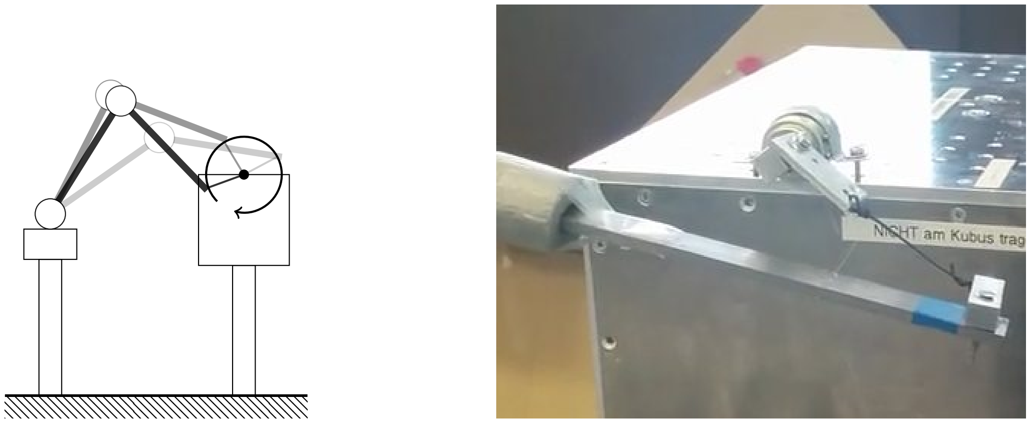
Figure 6.7.: Application of the contour force controller to a curved metal surface on the force cube for 20 seconds. The target normal force is 0.8 N and shown as the reference signal above. The magnitude of all force measurements from the force cube are shown.

The force controller can be seen to perform well in meeting the target force goal. As the surface changes, it misjudges the surface normal and begins to move too strongly into the surface. This can be seen to occur 3 times in the period shown in Figure 6.7. The controller manages to recover and continue moving. With a slower desired velocity, increased actuator bandwidth, or less noisy estimates of the surface normal, this behavior could be eliminated. While the 3D contact controller only managed to cover a roughly 10 cm portion of the flattest part of the surface, this controller can handle the entire surface, making it much more useful.

6.3.4. Circular Motion With Force Control

The task of moving in a circular motion with a given speed and force is a common task for force control. It typically involves moving one object around a revolute joint. This motion could be applied to turning a dial or a steering wheel (Mason 1981). In this thesis, the task of moving the end of a string fixed at a center point while maintaining a desired tension will be considered as shown in Figure 6.8a. Applications of this motion include tasks such as drawing a shape with a stencil using a pen or grinding the inner surface of a concave object.

To solve this task, the 2D contour force controller presented in Section 6.2.2 was applied. Additionally, a switching controller was used so that when not in contact the 3D proportional force controller from Section 6.2.1 would be used instead. The 3D force controller was configured to attempt to apply a force towards the ground. The



(a) Desired Arm Motion

(b) Experimental Setup

Figure 6.8.: Illustration of the desired arm motion of the robot effector around a pinned string can be seen in Figure 6.8a in which direction of motion is indicated. Figure 6.8b shows the experimental setup with TUDOR.

experimental setup with TUDOR can be seen in Figure 6.8b with a string attached to a freely rotating pivot bolted to the top of the force cube.

The controller was allowed to move the effector for two rotations around the pin and results were recorded. The path taken by the effector around the central pin is visualized in Figure 6.9 using the rigid link model to roughly approximate effector position. On the Southwest corner of the effector path, the starting position can be seen where the robot begins with the 3D force controller. As tension is made on the string, the 2D contour controller is enabled and the robot tries to maintain a target tension while staying in motion at 4 cm/s.

During the course of the rotations, the robot begins to lose tension at two different points in the circular motion. It then moves until tension is again reached and continues. This can be seen in the plot of the magnitude of the tension as measured by the contact force cube in Figure 6.10. One full revolution is shown over 80 seconds. Tension is lost once per revolution and additional spikes occur once per revolution, after which a recovery is made.

These losses of tension and spikes in force are caused by the safety controls in place during this experiment. Joint velocities were severely limited to approximately 2.3 degrees per second to reduce chances of damaging the robot or contact force cube in the event of a failure. The transformation of velocities from the global frame to the robot's joint space using the Jacobian, as in Equation 6.2.5, depends heavily on the pose of the robot. In two different poses, a desired global effector velocity can require very different joint velocities. This means the effects of the joint speed limitation will vary for each global effector position.

At certain angles from the string center point, the robot joints need to reverse direction and, to maintain velocity during this, the other joint must move faster to compensate. As can be seen in Figure 6.10, this caused the joint velocity limits to be reached, resulting in either a loss of, or spike in, tension. Allowing a larger range of joint movements should prevent this from occurring and allow this controller to operate without any problems.

The velocity commands shown in Figure 6.10 are quite noisy. This is the result of

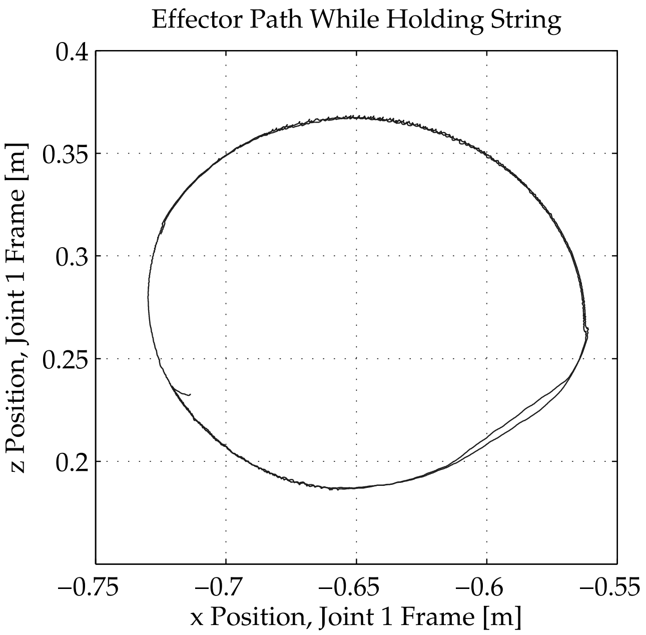


Figure 6.9.: Effector position during a circular motion while holding a string pinned at a central point. The effector is attempting to apply a fixed force while moving at a set velocity. The string is attached to a pivot bolted to the top of the contact cube allowing tension on the string to be measured.

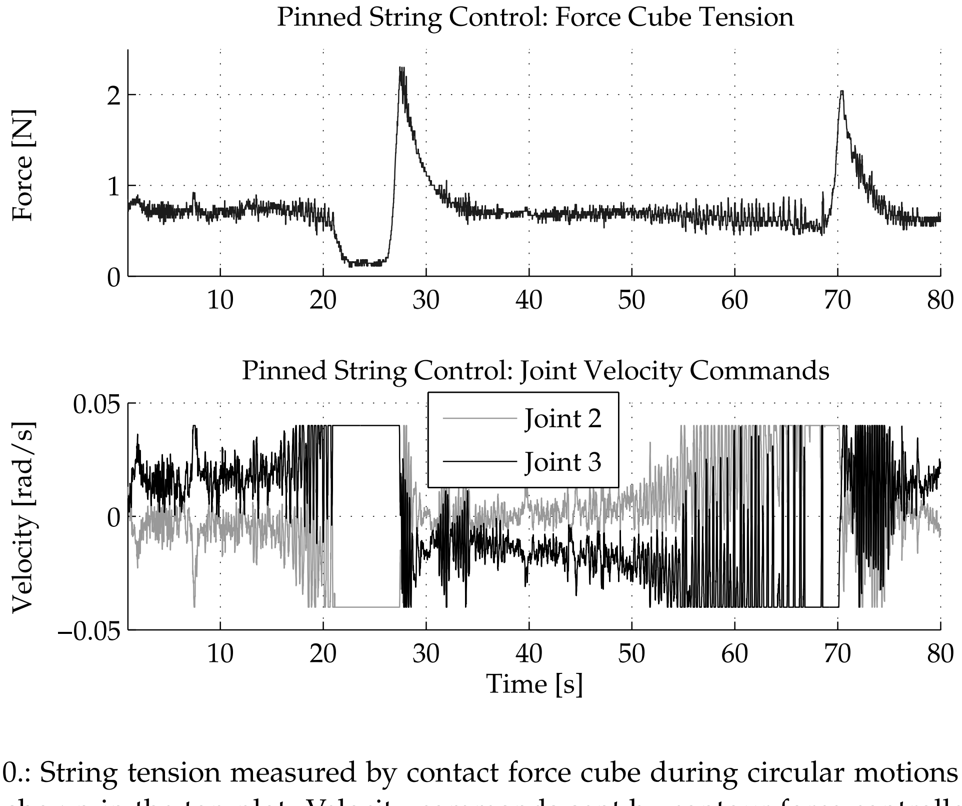


Figure 6.10.: String tension measured by contact force cube during circular motions a pinned string are shown in the top plot. Velocity commands sent by contour force controller, limited in magnitude to 0.04 rad/s, are shown in the bottom plot. Only one complete revolution is shown.

noisy estimates of the current contact force vector and the angle of this vector. Use of filtering similar to those used in Chapter 4 for payload estimate filtering could be used to improve this in the future.

7 Conclusions and Outlook

7.1. Outlook for Elastic Link Robots

With the possibility of commercial space exploration and a growing market for consumer robotics, the number of available applications for elastic link robots is increasing. Techniques developed for elastic link robots could also be used to help improve existing machinery with link elasticity such as cherry pickers and cranes. The techniques developed in this paper also point toward the possibility of retrofitting rigid link robots for force sensing applications given sufficiently sensitive strain sensors.

7.1.1. Design Recommendations

The results in this thesis provide insights that can be used when designing future elastic link manipulators for research or commercial applications.

To apply the results from this thesis, deflections with anticipated payloads should be kept sufficiently small so that the mathematical assumptions made in Chapter 2 continue to hold. The amount of deflection under the range of anticipated payloads should be examined in combination with the sensitivity of the strain measuring devices used. This tradeoff between strain sensitivity and link deflection should be carefully considered.

As in any application involving strain gages, care should be taken to negate temperature effects. As shown in all previous chapters, strain gage noise can be seen to cause most inaccuracies in strain prediction, force and payload estimation. Strain gage noise was also shown to be the cause for difficulties in estimating surface normals accurately in force control situations. Proper shielding of strain gages following a guide such as Hewlett-Packard (1981) can greatly help to reduce this noise.

In Section 4.2.2, it was found that when using multiple strain gages on a single elastic link, measurements can be fused using with Maximum Likelihood Estimation (MLE), reducing the effects of singularities in payload estimation. When designing elastic link robots with applications in weighing, careful placement of multiple strain gages on each link could allow for improved performance and robustness. Use of redundant strain gages and proper noise calibration can provide resiliency to sensor failures.

7.2. Future Work

The work in this thesis could be extended in the future in many ways. The dual extended Kalman filtering techniques applied to payload estimation in Section 4.3.4 could be adapted for effector force estimation. This could provide greatly improved force estimates with a large decrease in the noise originating from the strain gages. In turn, this could likely provide large performance gains in force control tasks.

The force controllers presented in Chapter 6 of this thesis control only force, not tip position. Further work to incorporate hybrid position/force controllers mentioned in Section 1.2 could be done. Additionally, some sort of visual servoing method could be applied in future work to control force and tip position in the presence of unknown environments.

Many of the force control results applied only in simulation or to horizontal flexible link manipulators with one link mentioned in Section 1.2 could be compared on TUDOR to provide useful advice on which control algorithms perform better and in what situations.

In Chapter 6, surface contour following algorithms relied on guesses for friction contact coefficients. Future work could be performed trying to estimate these. In addition, using a haptic finger style sensor as in Tanaka et al. (2003) attached to the effector, characterizations of surface characteristics could be made.

7.2.1. TUDOR 2 Recommendations

While TUDOR is an excellent platform for performing a wide variety of elastic link robotics research, a new version could greatly increase possibilities for future research. As mentioned in Section 1.2, the number of experimental setups with elastic link robots is quite limited. The MERIt dataset described in Section 1.4 makes strong efforts to make elastic link manipulator data available for open research. It is the opinion of the author of this thesis that TUDOR 2 should be made as a freely available open hardware and software platform. Not only would this provide a common base for other researchers to experiment upon, but it would help build a community in which researchers can share solutions to common problems.

TUDOR 2 should incorporate elasticity so that the effector vibrates in 3 dimensions instead of 2. Damping this would be a much more challenging control task and have very useful real-world applications. Depending on the design, this could also provide the ability to sense effector force in more than 2 dimensions of the effector frame. TUDOR 2 should also have more than the current 3 degrees-of-freedom in TUDOR, so as to allow more real-world force control and manipulation tasks to be explored.

With the current version of TUDOR, changing the configuration of links takes many hours. If TUDOR 2 were designed in more modular way it might be possible to reconfigure the link configuration in less than one hour. A design of standardized elastic link modules could be made to offer several motor mounts and several link sizes. Attachment between links could be made using various adapters so as to allow a wider variety of link configurations. In a manipulator whose configuration can be changed, wiring would be a significant concern. A standardized and extensible cabling harness could be used to make wiring simpler and ensure that proper shielding of strain gages is easy to maintain.

The current light gate sensing system used on TUDOR's joints for emergency safety stops is quite flimsy and difficult to accurately calibrate. Future versions of TUDOR could improve this significantly by using a laser cut mask instead of movable pins to trigger light gates.

The current version of TUDOR takes approximately 15 minutes to startup or restart due to requiring a full Simulink recompile to the platform OS. In the event of a crash or emergency stop in the middle of an experiment, this cycle time can be prohibitively long. In any future versions of TUDOR, care should be taken so that a full restart can be performed in seconds, not minutes.

The ability to attach TUDOR 2 to a mobile platform should also be considered. As many of the targeted applications of elastic link manipulators involve mobility, allowing testing in this context would be extremely useful.

7.3. Overview of Results

In Chapter 2, a model was developed for predicting strain in elastic link robot manipulators while at rest. This was extended into a general framework for converting standard torque-referenced robot equations to strain-referenced equations for elastic link robots. This model was applied to TUDOR, made linear in the parameters, and then trained using linear regression against the MERIt03 dataset. Performance was found to be quite good for predicting strain in the TUDOR robot while at rest.

Chapter 3 began by showing that when in motion, the equations developed in Chapter 2 for a static manipulator perform poorly. Rigid link manipulator dynamics components were then added to the strain-referenced elastic robot equations from Chapter 2. A similar process was then followed to make these equations linear in their parameters, and linear regression was used to find parameters from with MERIt02 dataset. Predictions of strain in the TUDOR robot while in motion were good, but contained spikes from numerical approximations of the joint angle derivatives. Performance was further improved through the addition of a CMA-ES optimized Kalman filter to provide better estimates of joint angle derivatives.

The task of static payload estimation was examined in Chapter 3. Initial estimates were made algebraically, and singularities were found to exist in the payload estimation error. A technique for avoiding these singularities by limiting the workspace was presented and shown to improve payload estimates significantly. The variance of payload estimates was examined and combined with Maximum Likelihood Estimation techniques to build a sensor fusion strategy. This was shown to provide significant performance benefits for static payload estimates by limiting the workspace.

Chapter 3 continued on to cover the task of payload estimation while in motion. An ordinary least squares approach was found to be performant but computationally expensive. To provide a tradeoff between estimate error and computational power, a moving window least squares approach was demonstrated with various window sizes. An extended Kalman filter parameter estimation approach was attempted and shown to be unstable through singularities. A dual extended Kalman filter approach was taken and optimized to provide fast and accurate estimates through singularities without requiring as much computing power as other approaches.

The topic of effector force estimation was covered in Chapter 5. Datasets for static contact and moving contact with TUDOR were created and discussed. A technique for estimating effector force was presented and shown to perform well against both datasets. It was determined that for the majority of force control tasks, quasi-static robot equations provide sufficiently accurate results.

Chapter 6 introduced basic types of force control and presented reasoning for how simple force controllers can be applied to elastic robot manipulators. A PI force controller for applying force in a single direction in 3 dimensions was designed. This was followed by a controller for applying force on a contoured surface while moving in a 2-dimensional plane. Step responses of the 3D PI force controller approaching a surface were presented and discussed. The contour controller was shown to be effective at applying a fixed force while moving across a flat surface. Both the contour and 3D force controllers were applied to the task of following a curved contour while applying a force. Finally, the task of moving a pinned string around in a circular motion while applying tension was performed with the contour controller. Deficiencies in these tasks were found to be likely caused by actuator bandwidth limitation.

It can be concluded that with proper damping algorithms, an elastic link manipulator can be used with standard rigid-link force control techniques.

Bibliography

Armstrong, B. (1989). “On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics”. In: *The International journal of robotics research* 8.6, pp. 28–48.

Atkeson, C. G., C. H. An, and J. M. Hollerbach (1986). “Estimation of inertial parameters of manipulator loads and links”. In: *The International Journal of Robotics Research* 5.3, pp. 101–119.

Bar-Shalom, Y., X. R. Li, and T. Kirubarajan (2001). *Estimation with Applications to Tracking and Navigation*. 1st ed. Wiley-Interscience.

Bazaei, A. and M. Moallem (2011). “Improving force control bandwidth of flexible-link arms through output redefinition”. In: *Mechatronics, IEEE/ASME Transactions on* 16.2, pp. 380–386.

Beer, F., E. R. J. Jr., and D. Mazurek (2012). *Vector Mechanics for Engineers: Statics*. McGraw-Hill Science/Engineering/Math.

Benosman, M. and G Le Vey (2004). “Control of flexible manipulators: A survey”. In: *Robotica* 22.05, pp. 533–545.

Carrera, E., G. Giunta, and M. Petrolo (2011). *Beam Structures: Classical and Advanced Theories*. 1st ed. Wiley.

Chapnik, B., G. R. Heppler, and J. D. Aplevich (1990). “Controlling the impact response of a one-link flexible robotic arm”. In: *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*. IEEE, pp. 1444–1449.

Ching, F. M. and D. Wang (2003). “Exact solution and infinite-dimensional stability analysis of a single flexible link in collision”. In: *Robotics and Automation, IEEE Transactions on* 19.6, pp. 1015–1020.

Chiou, B. C. and M Shahinpoor (1988). “Dynamic stability analysis of a one-link force-controlled flexible manipulator”. In: *Journal of robotic systems* 5.5, pp. 443–451. DOI: 10.1002/rob.4620050504.

Corke, P. (Mar. 2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. 1st ed. Springer.

Craig, J. J. (2004). *Introduction to Robotics: Mechanics and Control*. 3rd ed. Prentice Hall.

Einicke, G. A. (2012). *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*. InTech.

Eppinger, S. D. and W. P. Seering (1987). “Understanding bandwidth limitations in robot force control”. In: *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*. Vol. 4. IEEE, pp. 904–909.

Garcia, A and V Feliu (2000). "Force control of a single-link flexible robot based on a collision detection mechanism". In: *Control Theory and Applications, IEE Proceedings-*. Vol. 147. 61. IET, pp. 588–595.

Garcia, A. H., V. B. Feliu, and J. Somolinos (2001). "Gauge based collision detection mechanism for a new three-degree-of-freedom flexible robot". In: *ICRA*, pp. 3853–3858.

Gere, J. M. and B. J. Goodno (2012). *Mechanics of Materials*. 8th ed. Cengage Learning.

Gorinevsky, D., A. Formalsky, and A. Schneider (July 1997). *Force Control of Robotic Systems*. Ed. by 1. CRC Press.

Gourdeau, R. and H. Schwartz (1991). "Adaptive control of robotic manipulators: Experimental results". In: *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, pp. 8–15.

Gujarati, D. and D. Porter (Mar. 2002). *Basic Econometrics*. 4th ed. McGraw-Hill/Irwin.

Hansen, N. (Apr. 2012). "CMA-ES: Evolution Strategy with Covariance Matrix Adaptation for nonlinear function minimization". Matlab Function. URL: `https://www.lri.fr/~hansen/cmaes_inmatlab.html`.

Hansen, N. and A. Ostermeier (2001). "Completely derandomized self-adaptation in evolution strategies". In: *Evolutionary computation* 9.2, pp. 159–195.

Haykin, S. (2001). *Kalman Filtering and Neural Networks*. 1st ed. Wiley-Interscience.

Hewlett-Packard (Sept. 1981). *Practical Strain Gage Measurement*. Application Note 290-1.

Isermann, R. and M. Münchhof (2011). *Identification of Dynamic Systems: An Introduction with Applications*. Advanced Textbooks in Control and Signal Processing. Springer.

Lertpiriyasuwat, V., M. C. Berg, and K. W. Buffinton (2000). "Extended kalman filtering applied to a two-axis robotic arm with flexible links". In: *The International Journal of Robotics Research* 19.3, pp. 254–270.

Lew, J. Y. and W. J. Book (1993). "Hybrid control of flexible manipulators with multiple contact". In: *Robotics and Automation, 1993 IEEE International Conference on*. IEEE, pp. 242–247.

Lewis, F. L., D. M. Dawson, and C. T. Abdallah (2003). *Robot Manipulator Control: Theory and Practice*. 2nd ed. Automation and Control Engineering (Book 15). CRC Press.

Li, D. (1990). "Tip-contact Force control of one-link flexible manipulator: an inherent performance limitation". In: *American Control Conference, 1990*. IEEE, pp. 697–701.

Lightcap, C. A. and S. A. Banks (2010). "An extended Kalman filter for real-time estimation and control of a rigid-link flexible-joint manipulator". In: *Control Systems Technology, IEEE Transactions on* 18.1, pp. 91–103.

Lin, J and T.-S. Chiang (2003). "A new design of hierarchical fuzzy hybrid position/force control for flexible link robot arm". In: *American Control Conference, 2003. Proceedings of the 2003*. 6. IEEE, pp. 5239–5244.

Lohr, S. L. (1999). *Sampling: Design and Analysis*. 1st ed. Cengage Learning.

Maciejewski, A. A. and C. A. Klein (1989). "The singular value decomposition: Computation and applications to robotics". In: *The International Journal of Robotics Research* 8.6, pp. 63–79.

Malzahn, J., R. F. Reinhart, and T. Bertram (2014a). "Dynamics Identification of a Damped Multi Elastic Link Robot Arm under Gravity". In: *IEEE International Conference on Robotics and Automation (ICRA 2014).*

Malzahn, J. and T. Bertram (Sept. 2014b). "MERIt - A Multi-Elastic-Link Robot Identification Dataset". Institute of Control Theory and Systems Engineering (RST), TU Dortmund, Germany. Last accessed September 2014. URL: `http://www.rst.e-technik.tu-dortmund.de/cms/en/research/robotics/TUDOR_engl/index.html`.

Malzahn, J., A. S. Phung, F. Hoffmann, and T. Bertram (2011). "Vibration control of a multi-flexible-link robot arm under gravity". In: *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. IEEE, pp. 1249–1254.

Malzahn, J. and T. Bertram (2014c). "Collision Detection and Reaction for a Multi-Elastic-Link Robot Arm". In: *IFAC World Congress, Cape Town (South Africa).*

Mason, M. T. (1981). "Compliance and force control for computer controlled manipulators". In: *Systems, Man and Cybernetics, IEEE Transactions on* 11.6, pp. 418–432.

Matsuno, F., Y. Sakawa, and O Asano (1991). "Quasi-static hybrid position/force control of a flexible manipulator". In: *Robotics and Automation, 1991 IEEE International Conference on*. IEEE, pp. 2838–2843.

Matsuno, F. and K. Yamamoto (1994). "Dynamic hybrid position/force control of a two degree-of-freedom flexible manipulator". In: *Journal of Robotic Systems* 11.5, pp. 355–366.

Montgomery, D. C. and G. C. Runger (2010). *Applied Statistics and Probability for Engineers*. 5th ed. Wiley.

Murray, R. M., Z. Li, and S. S. Sastry (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

Park, N.-C., H.-W. Park, H. S. Yang, and Y.-P. Park (2002). "Robust vibration/force control of a 2 DOF arm having one flexible link with artificial pneumatic actuators". In: *Journal of Vibration and Control* 8.3, pp. 405–423.

Ruina, A. and R. Pratap (2014). *Introduction to Statics and Dynamics*. Oxford University Press.

Siciliano, B. and W. J. Book (1988). "A singular perturbation approach to control of lightweight flexible manipulators". In: *The International Journal of Robotics Research* 7.4, pp. 79–90.

Siciliano, B. and L. Villani (1999). *Robot force control*. Springer.

Siciliano, B. and L. Villani (2000). "From Indirect to Direct Force Control: A Roadmap for Enhanced Industrial Robots". In: *Invited Paper, Robotica*.

Spong, M. W., S. Hutchinson, and M. Vidyasagar (2005). *Robot Modeling and Control*. 1st ed. Wiley.

Suzuki, J., T. Murakami, and K. Ohnishi (2002). "Position and force control of flexible manipulator with position sensitive device". In: *Advanced Motion Control, 2002. 7th International Workshop on*. IEEE, pp. 414–419.

Swevers, J., W. Verdonck, and J. D. Schutter (Oct. 2007). "Dynamic model identification for industrial robots". In: *Control Systems, IEEE* 27.5, pp. 58–71.

Särkkä, S., J. Hartikainen, and A. Solin (Aug. 2011). "EKF UKF Toolbox for Matlab V1.3". Matlab Library, Last accessed September 2014. URL: `http://becs.aalto.fi/en/research/bayes/ekfukf/`.

Tanaka, M., J. L. Lévêque, H. Tagami, K. Kikuchi, and S. Chonan (2003). “The "Haptic Finger"–a new device for monitoring skin condition”. In: *Skin Research and Technology* 9.2, pp. 131–136.

Ueno, N., M. M. Svinin, and M. Kaneko (1998). “Dynamic contact sensing by flexible beam”. In: *Mechatronics, IEEE/ASME Transactions on* 3.4, pp. 254–264.

Verdonck, W. (2004). “Experimental robot and payload identification with application to dynamic trajectory compensation”. Advisors: Prof. dr. ir. J. Swevers Prof. dr. ir. J. De Schutter. PhD thesis. Katholieke Universiteit Leuven.

Williams, V. V. (2011). *Breaking the Coppersmith-Winograd barrier*.

Yang, J.-H., F.-L. Lian, and L.-C. Fu (1995). “Adaptive hybrid position/force control for robotic manipulators with compliant links”. In: *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*. Vol. 1. IEEE, pp. 603–608.

Yim, W. and S. N. Singh (1993). “Inverse force/end-point control, zero dynamics and stabilization of constrained elastic robots”. In: *American Control Conference, 1993*. IEEE, pp. 2873–2877.

Young, W. and R. Budynas (2002). *Roark’s Formulas for Stress and Strain*. Ed. by 7. McGraw-Hill Professional.

Zisserman, A. (Feb. 2007). “Estimation and Inference Lectures 3 & 4: Estimators”. Last accessed September 2014. URL: `http://www.robots.ox.ac.uk/~az/lectures/est/lect34.pdf`.

A

Appendix: Symbol Definitions

Symbols are grouped by the chapter they are first used in.

Chapter 1

Symbol	Definition
x_{sij}	Strain gage j position from joint i [m]
l_n	Length of link n [m]
$q_1 ... q_n$	Generalized robot joint coordinates for joint n
${}^i x$	x-axis in the ith frame
${}^i y$	y-axis in the ith frame
${}^i z$	z-axis in the ith frame
${}^E x$	x-axis in the effector frame

Chapter 2

Symbol	Definition
x_{sij}	Strain gage j position from joint i [m]
$M(x)$	Internal bending moment of a beam at position x [N·m]
$M_g(x)$	Internal bending moment of a beam's weight due to gravity [N·m]
$M_f(x)$	Internal bending moment of a beam's perpendicular tip force [N·m]
$M_M(x)$	Internal bending moment of a beam's applied moment [N·m]
$V(x)$	Internal shear force of a beam at position x [N]
$w(x)$	Beam perpendicular force, applied at position x [N]
$\boldsymbol{\mathcal{F}}$	Generalized 6-dimensional force vector [N]
f_i	Force in the ith axis [N]
n_i	Moment about the ith axis [N·m]
V_f	Internal shear force caused by a perpendicular tip force [N]
ρ_b	Force per unit length of distrusted beam mass [N/m]

Symbol	Definition
m	Beam mass [kg]
g	Gravitational constant [m/s^2]
θ	Beam angle with respect to horizontal
x	Position in a beam
V_g	Internal shear force caused by a beam's weight due to gravity [N·m]
M_{appl}	Tip-applied bending moment [N·m]
F_{xc}	Critical buckling load [N]
$F_{xc,i}$	Critical buckling load of link i [N]
E	Young's Modulus [Pa]
I	Area moment of inertia [m^4]
K	Column effective length factor
λ	Beam slenderness ratio
λ_n	Beam slenderness ratio of link n
A	Beam cross-sectional area [m^2]
y_b	Distance of a beam's surface from the beam's neutral fiber [m]
$M(x)$	Internal bending moment of a beam at position x [N·m]
M_n	Internal bending moment of link n [N·m]
x_s	Strain gage position
x_{sn}	Strain gage position in link n
$\mathcal{E}$	Strain [m/m]
$\mathcal{E}_n$	Strain for link n [m/m]
a_n	Length of link n n [m]
m_n	Mass of link n n [kg]
m_4	Mass of payload [kg]
F_{lj}	Effector tip force in the j-axis of the effector frame [N]
$\boldsymbol{\mathcal{T}}$	Vector of joint torques [N·m]
$\boldsymbol{q}$	Vector of generalized robot coordinates
$\boldsymbol{g}(\boldsymbol{q})$	Gravity torque model with respect to generalized robot coordinates [N·m]
$\boldsymbol{J}(\boldsymbol{q})$	Robot Jacobian
$\boldsymbol{\Delta\tau}$	Difference term vectors for converting torque to strain
$\boldsymbol{\Delta\tau}_g$	Difference term vectors for gravitational effects due to the beam's mass
$\boldsymbol{\Delta\tau}_f$	Difference term vectors due to beam tip force
$\boldsymbol{\Delta\tau}_M$	Difference term vectors due to applied moment
$\boldsymbol{\rho_b}$	Vector of gravitational force per unit length [N/m]
$\boldsymbol{x_s}$	Vector of strain positions [m]
$\boldsymbol{\theta}$	Vector of beam angles [rad]
$\boldsymbol{l}$	Vector of beam lengths [m]
$\boldsymbol{f_y}$	Vector of perpendicular tip forces [N]
$\Delta\tau_{g,n}$	Difference term for gravitational effects of link n
$\Delta\tau_{f,n}$	Difference term vectors for beam tip force of link n
${}^{B}\boldsymbol{P}_{AORG}$	Vector offset between frames A and B

Symbol	Definition
${}^{k}\boldsymbol{f}$	3 component force vector in frame k
${}^{k}\boldsymbol{n}$	3 component force vector in frame k
${}^{B}_{A}\boldsymbol{R}$	Rotation transform between frames A and B
$m_{s,n}$	Mass of links after n in the effector chain
${}^{n}\boldsymbol{J}(\boldsymbol{q})$	Robot Jacobian matrix with respect to link n
${}^{n}\boldsymbol{j}_{k}(\boldsymbol{q})$	kth row of the Jacobian with respect to link n
$\hat{\mathbf{j}}_{\mathbf{n}}^{*}$	Unit vector of the y-axis of the link n augmented with 3 additional zeros
${}^{n}_{E}\boldsymbol{T}$	Force-moment transform from Section 2.6.3 from effector frame E to the frame of link n
$\mathcal{T}_{n}$	Torque of joint n [N·m]
$\mathcal{C}_{23}$	$\cos{(q_2 + q_3)}$
$\mathcal{S}_{23}$	$\sin{(q_2 + q_3)}$
$\mathcal{C}_{2}$	$\cos q_2$
$\mathcal{C}_{3}$	$\cos q_3$
R_i	ith linear regression parameter

Chapter 3

Symbol	Definition
$\boldsymbol{I}(\boldsymbol{q})$	Robot inertia matrix with respect to generalized coordinates $\boldsymbol{q}$
$\boldsymbol{C}(\boldsymbol{q})$	Robot Coriolis matrix with respect to generalized coordinates $\boldsymbol{q}$
$\dot{\boldsymbol{q}}$	First derivative of generalized robot coordinates vector $\boldsymbol{q}$
$\ddot{\boldsymbol{q}}$	Second derivative of generalized robot coordinates vector $\boldsymbol{q}$
g_{Ri}	Gear ratio for joint i
r_{ij}	Center of mass offset in the i-axis for link j
I_{ijk}	Component of the moment of inertia tensor matrix for axis i and axis j for link k
$\boldsymbol{\Phi}_{\tau}$	Regressor matrix
$\boldsymbol{\theta_{linear}}$	Parameter matrix for which the regressor matrix $\boldsymbol{\Phi}_{\tau}$ is linear
$\boldsymbol{x}$	State space state vector
$\boldsymbol{y}$	State space output vector
$\boldsymbol{x}$	State space output vector
$\boldsymbol{\mathcal{O}}$	State space observability matrix
$\boldsymbol{R}$	Kalman measurement noise matrix
$\boldsymbol{R}$	Kalman process noise matrix

Chapter 4

Symbol	Definition
N_n	Numerator function of the link n payload estimate equation
D_n	Denominator function of the link n payload estimate equation
X	Random variable
Y	Random variable
f	Linear function
$\hat{m}_4$	Estimate of payload mass [kg]
$\mathcal{L}(x)$	Likelihood function of x
$P(z\|x)$	Probability of measurement z given x
$\hat{x}$	Estimate of the true value x
$\propto$	Symbol meaning proportional to
z_i	Sensor i measurement
$\hat{x}_{MLE}$	MLE estimate of the true value of x
w_i	MLE weight of sensor i

Chapter 6

Symbol	Definition
$\hat{f}_{li}$	Force estimate in the $i$the axis
${}^n\hat{f}_{li}$	Force estimate in the $i$the axis rotated to frame n
${}^0_E\boldsymbol{R}$	Rotation transform between the effector frame and the global frame
t_k	Time at discrete step k
$\boldsymbol{r}(t_k)$	Reference signal vector at time t_k
$\boldsymbol{e}(t_k)$	Error vector at time t_k
${}^i\boldsymbol{e}(t_k)$	Controller velocity command output vector at time t_k in the frame i
${}^ik_p$	Proportional gain vectors for each Cartesian axis in the ith frame
${}^ik_p$	Integration gain vectors for each Cartesian axis in the ith frame
${}^jv_i$	Velocity command in the ith axis relative to the jth frame
$\dot{\boldsymbol{q}}_d$	Desired joint velocities
$\boldsymbol{\tau}$	Contour tangent vector
$\boldsymbol{n}$	Contour normal vector
e_n	Normal force error
F_n	Normal force
F_{dn}	Desired normal force
${}^jv_n$	Normal component velocity command relative to the jth frame
${}^j\tau_n$	Tangent component velocity command relative to the jth frame
β	Angle between frictionless surface tangent estimate and the actual value
γ	Guess for the value of β
$\boldsymbol{\tau}^*$	Estimate of contour tangent vector
$\boldsymbol{n}^*$	Estimate of contour normal vector