Modeling of Geometrically Nonlinear Flexible Structures for Control

by

Carl Blaurock

S.B. Massachusetts Institute of Technology (1991)
S.M. Massachusetts Institute of Technology (1994)

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the

Massachusetts Institute of Technology

September 25, 1997

Signature of Author Carl Blaurock
Department of Aeronautics and Astronautics
September 25, 1997

Certified by
Professor David W. Miller
Professor, Department of Aeronautics and Astronautics
Thesis Committee Chair

Certified by
Professor Eric Feron
Professor, Department of Aeronautics and Astronautics

Certified by
Professor Warren Seering
Professor, Department of Mechanical Engineering

Certified by
Mr. Eric Schmitz
Research Scientist, Lockheed-Martin Astronautics, Denver CO

Accepted by
Professor Jaime Peraire
Chairman, Department Graduate Committee

Abstract

Space-based robotic arms are a versatile element of a manned space program. They act as "force multipliers" to allow completion of many tasks more rapidly, cheaply, and safely, by reducing or eliminating time spent by astronauts in Extra-Vehicular Activities (EVA). One class of manipulators are the macro arms such as the Shuttle Remote Manipulator System (SRMS). These large (50 foot) anthropomorphic arms offer the ability to move capture, deploy, and maneuver massive payloads. However, mass budgets dictate that the manipulators be built lightly and as a result are structurally flexible. Fundamental vibration modes occur well within the bandwidth of the human operator, forcing a setting period at the end of each move.

High-performance global control of flexible manipulators has been an active topic of research for two decades. Experimental results on ground-based flexible arms have shown that such control algorithms can reduce residual vibration of the manipulator end-point. However, on-orbit implementation has been complicated by the *ad hoc* gain-scheduling approaches generally used to account for configuration changes. Such approaches offer minimal guarantees of stability and require a separate controller for each operating point and payload. Recently, however, a number of time-varying robust control algorithms, based on linear robust control methodologies, have been investigated in the controls literature. These algorithms parameterize the plant as a Linear Time-Invariant (LTI) system and a structured time varying operator. Linear algebra tools can then be brought to bear on the LTI plant model. Among the advantages over competing nonlinear control design approaches are the ability to formulate performance objectives in the frequency domain, and the experimentally demonstrated ability of linear design tools to account for parametric uncertainty in the plant.

Time-varying robust control requires a suitable parameterization of the nonlinear dynamics of the manipulator. The current thesis presents an approach for deriving the dynamics in a Linear Fractional form directly suitable for small-gain control design. The method is also analytically advantageous, being applicable to arbitrary

configurations and numbers of flexible links. Additionally, the resulting model offers run-time computational advantages over other formulations of the equations of motion, and is naturally parallelizable. The method is demonstrated by modeling the SRMS. Modeling accuracy is shown to be comparable to a standard modeling approach, and the parameterized model enables the direct application of time-varying robust control tools.

Acknowledgments

This thesis was sponsored by the M.I.T. Space Engineering Research Center under NASA training grant NAGW-1335 with SharonLeah Brown as contract monitor.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Space-based robotic arms are a versatile element of a manned space program. They act as "force multipliers" to allow completion of many tasks more rapidly, cheaply, and safely, by reducing or eliminating time spent by astronauts in Extra-Vehicular Activities (EVA). They also enable many tasks which are beyond the capabilities of an unassisted human, such as manipulation of massive payloads. Among the tasks which benefit are space construction, inspection and repair, experiment manipulation, satellite capture, station-keeping, and release, and EVA during which the astronaut is supported by the SRMS.

Many types of arms have been studied: long-reach "macro" arms, such as the Shuttle Remote Manipulator System (SRMS), "dextrous" arms such as the Japanese Small Fine Arm, and free flying arms. The macro arms form an interesting class, both due to their utility, and due to the challenge presented by achieving high-precision positioning in the face of strict structural constraints. Their utility lies in the ability to maneuver payloads into, out of, and around the Shuttle payload bay, and to anchor the Shuttle to other orbital structures. Structural constraints arise from operational requirements such as long reach and high payload mass, as well as mass budgets dictated in part by launch costs. Such design constraints necessarily result in a structurally flexible manipulator, generally with frequencies within the bandwidth of the human operator. These dynamics lengthen the time required for any given task, and thus reduce astronaut effectiveness and adversely affect safety. A study by Newsom *et al.* estimates that 30% of the operational time of the arm is spent waiting for the arm vibrations to settle out[1].

Recent studies of high-precision controlled structures have demonstrated that modern control algorithms offer a realizable means to achieve performance objectives in the presence of tight structural constraints. However, the design of high-performance flexible manipulator controllers is complicated by geometric nonlinearity and large changes in payload mass (up to two orders of magnitude). The controlled arm is also subject to strict requirements on safety (and thus closed loop stability). The solution has been to implement low performance, high stability margin controllers. The first, and currently the only, operational space-based manipulator, the Shuttle Remote Manipulator System (SRMS), uses independent joint-rate feedback. The resulting closed loop system is stable for a wide range of payload masses and dynamics. However, the control system cannot control flexible link deformations that lead to end-effector oscillation. The operational solution is to wait for the system damping to reduce residual oscillation to an acceptable level before completing a fine positioning maneuver. This waiting period will become significant for large payloads; the SRMS holding a $60,000$ pound payload will have a period of oscillation of about 100 seconds, and the residual damping is low enough that many cycles will be required for any oscillation to damp out.

Much work has been done in the years since the SRMS came online. Theoretical research into nonlinear and flexible structure control has pointed the way to implementable high-performance control of flexible arms. Numerous experimental demonstrations have shown that these techniques can dramatically improve flexible arm performance[2, 3, 4, 5, 6, 7, 8]. For example, reductions of $70 - 85\%$ in settling time, compared to joint proportional-derivative (PD) control, are typical.

Table 1.1: Flexible Structure Design Approach

	Structural Modeling	Uncertainty Modeling	Control Design
Linear Structures	FEM[9, 10] Suspension and Gravity[9, 10] Measurement Models[11, 12]	Measured bounds[13] Uncertainty projection[14, 15, 16]	Small Gain[17] Polytopic[18] Robustified[19]
Geometrically Nonlinear Structures	Kinematics[20, 2] Symbolic Modeling[21] Nonlinear FE[22, 23] Measurement Modeling[24]	Uncertainty Realization? Order Reduction[25, 26]	Polytopic[18] Observer[27] Small Gain[28, 18]

1.2 Objectives

The primary objective of the current work is to create a modeling framework which enables the application of modern robust and time-varying control design algorithms to the control of flexible spaced-based manipulators. Robust control approaches have been experimentally demonstrated on high order ground and space based flexible testbeds. As shown in Table 1.1, the application of such algorithms requires three key steps: creation of a state space model of the structure, identification and modeling of the uncertainties, and application of the control synthesis algorithm.

The process begins with a linear structural model. The process used to create the model, represented in the upper left block of Table 1.1, may incorporate a physical model with model updating[9, 10]. Alternatively a measurement model can be created, if the physical system exists and can be measured [11, 12]. The latter restriction exists because such measurements must be done *in situ*, for example in space.

The modeling process returns a linear dynamic model of the physical system. The model will contain deviations from the true system. Deviations can arise, for example, from parameter errors, modeling simplifications, or time-variations and nonlinearities. The next step in creating the control is to quantify the "location" and magnitude of the errors. This will be called the uncertainty model. The "location" of the uncertainty indicates which model parameters are considered to be uncertain. The uncertainty modeling further breaks down into two steps. First, the uncertainties must be measured[13]. This requires identification of measurable parameters which uniquely determine the uncertain model parameters. Examples are modal frequencies and modal amplitudes. Next, the measured variations must be realized as a variation in the structural model parameters. Different approaches will be used to capture errors in a physical model and in a measurement model. For a physical model, the measured uncertainties must be projected onto the physical model parameters[14, 15]. Measurement models may simply require uncertainties on the system modes[16].

The final step in the control design process is the application of the design algorithm. The design algorithm will typically be chosen based on the statement of performance objectives and on the types of uncertainty expected in the system. The design algorithm will generally be distinguished by the form of the uncertainty model.

A natural parallel exists between the control of linear flexible structures and geometrically nonlinear structures. Specifically, the static uncertainties which lie within some bounds can be replaced by bounded time-varying quantities. Once in the robust control design framework (upper right block), extensions allow the application of the control design algorithms to time-varying plants (lower right block). In recent years many time-varying design algorithms, differing mainly in the parameterization of the time-varying model, have been proposed[29, 27, 28, 30]. The common characteristic is that the design algorithms use linear design tools. This places a constraint on the dynamic model and the uncertainty model. Specifically the dynamic model must be linear, with a specific parameterization for the uncertainty.

The proposed geometrically nonlinear control design process will follow the same steps as the linear design process. First the system to be controlled is modeled.

This generally takes the form of a physical model, due to the necessity of capturing the time-varying parameter dependence of the model. However, measurement based models have been studied. Demeo *et al.*[24] create an Observer Kalman Filter Identification (OKID) model of the Manipulator Development Facility, a full size mockup of the SRMS.

Next the parameter dependence must be expressed as a time-varying "uncertainty" on a linear system. In the controls literature this is typically done with a polynomial fit to the model parameter trajectories, as a function of the time-varying uncertainties. Note that an important difference from the uncertainty model for the linear system is that the time-varying parameters are known. That is, not only their absolute bounds but their current value are known. The trajectories of the system parameters (the state matrices) are correlated via the uncertainties, and an accurate model must capture the correlation. The uncertainty model may therefore be generated at the same time as the structural model.

Finally, the robust/time-varying control synthesis algorithm is applied. It is worth noting that the typical sample problem in the controls literature is a flight control problem. The characteristics of the sample plant often differ significantly from the characteristics of a flexible manipulator. Specifically the flight dynamics are normally low order, with wide variations in modal damping as well as frequency (unstable dynamics at some flight regimes are common). In contrast, manipulators are most often conservative (the eigenvalues lie on the imaginary axis of the s-plane) but of high order. This has implications for the numerical tractability of the synthesis algorithm, as well as the run-time complexity of the resulting controller.

A side note must be made regarding the uncertainty model. As noted for the linear design process, the uncertainty model is driven by the control design algorithm. Hence the algorithm must be known in order to perform the modeling. The current work will therefore specialize at the outset, in Chapter 3, to a particular design algorithm. However, connections exist between the different uncertainty parameterizations. With simple manipulations it will be possible to move between the uncertainty model appropriate for the current control algorithm, and that appropriate for many other time-varying robust control design algorithms.

1.3 Survey of Previous Work

1.3.1 Flexible Multibody Dynamics

The field of modeling for flexible multibody dynamics has received a great deal of attention since the advent of the Shuttle Remote Manipulator System (SRMS) in the late 1970's. The need to determine loadings, simulate specific operations, and design specialized control algorithms for macro arms, has driven a great deal of interest in the generation of accurate, numerically efficient dynamic models.

Accuracy has been addressed in part by investigating which terms in the equations of motion are dynamically significant. Book[21] has examined various assumptions used in the generation of flexible multibody models, to determine their suitability and the limitations they impose. The same work discusses the utility of symbolic mathematics programs for the creation of nonlinear models. Similarly, Padilla[20] and Oakley[2] have examined methods to account for large flexible deformations in the kinematics and dynamics. The latter citation also discusses the appropriate form for spatial shape functions used in assumed modes modeling. Sincarsin and Hughes[23] have generated perhaps the seminal form for the flexible dynamics, in terms of time varying mass and Coriolis matrices.

Numerical integration approaches for accurate time simulations of structural systems have also received much attention. Standard techniques such as Runge-Kutta schemes have been applied. Stiff equation solvers for systems with wide time scales, or for contact dynamics, have been studied. Similarly, perturbation methods have been used to sequentially solve for fast dynamics such as stiction effects, followed by slower structural flexural motions. Park notes that stability of the integrated equations of motion is a function of the integration method and of the structural equations[22].

Numerical efficiency has also received attention. Book[31] describes a recursive formulation for manipulator dynamics. Liu[32] derives a recursive Lagrangian formulation which simplifies the derivation of a 19 DOF model of the SRMS. Parallel computational architectures[33, 22] have been examined to speed computation. Component Mode Synthesis has been studied as an approach to create low order models, driving the understanding of the linkage of multiple component models[34].

Of particular interest are the models which are generated for the purposes of control design. Accuracy is important in achieving stability of the model-based controller. Computational efficiency is necessary if the model forms part of a nonlinear observer[2, 24]. Many researchers have used Lagrangian methods as a flexible manipulator framework[35, 36, 37, 38]. Schmitz[35] has derived a model of a two link planar manipulator testbed, at Lockheed-Martin's Research and Technology Department. Gebler[36] has determined the dynamics of a two-link, three dimensional manipulator. In both cases the elements of the system matrices are evaluated numerically. Khorrami[37] has determined the dynamics of a planar link symbolically, and analyzed them via asymptotic expansions to determine a feedback-linearizing controller.

Kane's equations are also widely used[2, 39]. Oakley[2, 40] used an assumed modes model to create a series of endpoint controllers for a planar arm. Both controllers were experimentally validated. Ramey[39] uses an assumed modes model for the Lockheed-Martin planar arm, to implement an LQR controller.

Newtonian methods are generally restricted to Finite Element Model (FEM) formulations. Carusone[41] uses a FEM model of a two link, planar manipulator. Menq and Chen[42] develop a beam element incorporating rigid body rotations to model a single flexible link. Other modeling packages have been used. Schmitz and Ramey[4] use TREETOPS to model the Lockheed-Martin Arm.

Another large class of modeling methods are based on system identification concepts. Rovner[43] used a linear fractional parameterization to identify an uncertain payload on a single flexible link. Similarly, Alder and Rock[44] used a subspace-fitting technique to determine lightly damped poles in a closed region of the S-plane which were associated with a lightly damped payload mode. Yurkovich *et al.* at Ohio State have done extensive experimental work in system identification of flexible manipulators, including time-domain Auto-Regressive Moving Average (ARMA) models[3], and frequency domain transfer function estimation[45]. Demeo *et al.*[24] used an Observer/Kalman Filter Identification (OKID) procedure to implement robust LQG control on the Systems Engineering Simulator mockup of the SRMS at Johnson Space Center. A current area of research is in the area of multidimensional system realization[46].

1.3.2 Uncertainty Modeling

A substantial body of literature exists concerning uncertainty modeling for linear systems. Classes of uncertainties are defined, for example, by Doyle and Stein in 1981[17].

Parametric uncertainties on linear systems have been estimated from measurements[15, 14, 13]. Crawley *et al.*[13] measure modal parameter variations due to laboratory suspension effects, disassembly/re-assembly, 0-g/1-g effects, etc. to determine mean variations. Campbell[14] uses the Discrete Extended Kalman Filter to estimate modal parameters and their covariance, then uses a projection method to map the modal uncertainties to FEM mass and stiffness matrices. Douglas[15] uses a small gain uncertainty description in a coupled system/uncertainty description. It is worth noting that the latter work is motivated by the desire to apply small gain control methods.

Linear parameterizations of time-varying systems have been used in the flight controls literature[47, 18]. A polynomial representation of the parameter variations in a physical model is often used[30, 47, 18]. Bennani *et al.*use a polynomial description of the aircraft parameter variation to control the short period mode of an aircraft[30]. Bodenheimer *et al.*[47] create a polynomial fit to the parameter-varying system matrices of a ducted fan testbed. Beck[25, 46] presents an LMI approach to the order reduction of such polynomial fits.

1.3.3 Flexible Manipulator Control Design

The costs associated with manipulator flexibility have motivated investigations into a wide range of control algorithms. Cetinkunt and Book[48] investigate the performance limitations imposed by joint feedback, motivating the use of non-colocated sensors. Similarly, Schmitz[4, 35], Oakley[2], and Scott*et al.*[49] experimentally verify the performance advantages of non-colocated feedback. This motivates a search for control design approaches which can accommodate the phase losses inherent in non-colocated feedback.

Typically, endpoint position, or its temporal derivatives, are included in the sensor suite to enhance performance. The simplest class of endpoint controllers use constant gain or proportional-derivative (PD) loops. Schmitz[4] uses a lead-lag controller to stabilize the Lockheed-Martin arm around a nominal position. Kotnik[50] uses constant gain acceleration feedback to damp oscillations in a single link arm.

The largest class of control methods to be validated experimentally are the Linear Quadratic Regulator (LQR) and Linear Quadratic Gaussian (LQG). Scott*et al.*[49] use a robust Multiple Model LQG controller, using tip accelerometers, to actively damp the Systems Engineering Simulator. Ramey[39] uses an LQR design to control the endpoint of the Lockheed-Martin planar testbed. The flexible state measurements are provided by a series of strain gauges located along the link.

Nonlinear and time-varying extensicns to LQR and LQG have also been demonstrated. Carusone[41] implements a gain-scheduled LQR design on a two link planar robot, in which a series of fixed gain controllers are designed for many set points in the workspace. The gains are interpolated at run-time based on the joint positions. Again, strain gauges are used to measure the flexible states. Oakley[2] uses a robust, constant gain LQR design, coupled to a Constant Gain Extended Kalman Filter state estimator, to control a two link, planar arm.

Direct applications of nonlinear control theory are not prevalent in the literature. However, many instances can be found of nonlinear control for rigid body motion, with Linear Quadratic control for the flexible states. Aoustin and Chevallereau[5] and Siciliano and Book[51] use singular perturbation theory to separate the (closed loop) rigid body dynamics from the flexible dynamics, then apply LQR control to the fast flexible dynamics. Madhavan and Singh[52] use a variable structure (switching) controller to guide a single link manipulator to the vicinity of the desired rest point, then use an LQR controller to achieve final positioning. Khorrami[53] applies the feedback-linearizing controller for a rigid planar manipulator with equivalent mass parameters to a planar testbed with a rigid inner link and flexible outer link. Then an LQR controller is used to stabilize the system.

Adaptive control is the one of the most demonstrated direct nonlinear control design algorithms. Yurkovich *et al.*[3] demonstrate a one-step ahead self-tuning controller on a two flexible link, planar arm. The same group uses a Time-varying Transfer Function Estimate (TTFE) approach to identify pole locations in order to tune controller parameters[54]. Askew and Sundareshan[55] use a neural network to identify a payload on a single link beam, and schedule a variable structure controller based on the parameter. Clarke *et al.*[56] proposes a time-domain adaptive control algorithm called Generalized Predictive Control which is robust to nonminimum phase behavior. Zaki[38] uses an implementation of Model Reference Adaptive Control (MRAC) to stabilize a two-link, three degree of freedom simulation.

Feedforward techniques also merit some attention, based on the number of approaches which have been used. Singer and Seering[57] use impulse prefiltering to reduce residual motion at the end of a commanded move of the Draper RMS Simulator (DRS), a high fidelity simulation of the SRMS. Khorrami[53] uses a feedback linearizing inner loop to minimize natural frequency changes with configuration, then applies impulse prefiltering to the closed loop system. Gebler[36] feeds forward an estimate of the tip deflection to the joint position servo, improving trajectory tracking performance. Bayo[58] and Pfieffer[59] use a computed torque approach to improve trajectory tracking. These methods offer improved command-following performance, generally at low computational cost, without compromising system stability.

1.4 Summary of State of the Art

1.4.1 Modeling State of the Art

Flexible multibody dynamics have been actively investigated for many years. Numerous modeling approaches have been investigated in the literature. Four groupings emerge:

(i) Lagrange methods, which derive the EOM from a system energy function

(ii) Kane's Equations, a highly formalized way to solve for inertial forces from expressions for the inertial velocities

(iii) Finite Element Modeling (FEM), a subset of Lagrange methods which spatially discretizes the system energy function using a simplified set of functions

(iv) Simplified modeling packages which use high-level parameters (lumped masses, etc.) to arrive at approximations to the system dynamics

These approaches must be examined in more detail.

Lagrangian methods employ the system energy function, or Lagrangian[60]. The energy function is written in terms of a set of assumed degrees of freedom (DOF). Then Lagrange's equation is applied to the energy function to determine the EOM. The primary advantage is accuracy. The Lagrangian model will be as accurate as the energy function allows it to be. Generally the system energy function is straightforward to formulate. That is, all of the "important" forces for a particular system can be included in the energy function. The source of the flexibility, however, is also the cause of the greatest drawback. The greatest disadvantage is the difficulty of evaluating Lagrange's equations. In most cases there are many possible choices for assumed DOF, and few clues as to which will lead to the most compact model. The consequences of choosing the wrong DOF can range from an inaccurate model (by not accurately capturing system motions), to unnecessarily complex system dynamics, up to being unable to muster the analytical power to perform the evaluation. In many cases the energy function is complex enough that symbolic modeling packages (such as Mathematica) must be used. For realistic systems, such as three-dimensional manipulators, the evaluation may be too intensive and the package may fail to provide an answer. Even if this is not the case, such packages cannot adequately manipulate the resulting EOM to reflect any structure which might be present. For all practical purposes this structure is lost to the controls designer who attempts to use the model.

Kane's equations are essentially a highly formalized version of $F = ma$[61]. A force balance is written in terms of generalized coordinates, between inertial forces produced by generalized accelerations and externally imposed forces. The velocities of a point on the flexible structure are defined in terms of motions in various reference frames, and differentiated to determine accelerations. The accelerations are then integrated over the mass to determine inertial forces. Inertial forces are then equated to external forces to determine the system EOM. The advantage is that the process is highly formalized, so it can be generalized to highly complex systems. The primary disadvantage is that the computations necessary to determine the accelerations are demanding, similar to the Lagrangian form.

It is worth noting that Kane's method has been advanced as a means to produce a more compact set of EOM. However, flexible manipulator models derived using Kane's method have been of the same order complexity as Lagrangian models. See, for example, the 2 link planar arm model of Reference [2], derived using Kane's equations, compared to the dynamics for the same type of system derived using Lagrange's equations[37]. Both approaches capture the nonlinear kinematics of a structure undergoing flexible deformation and large angle, large rate rigid body motion, thus are of comparable accuracy (given appropriate model parameters). This indicates that the complexity of the dynamics of such structures is a result of the nonlinear kinematics.

FEM is a subclass of Lagrangian methods, in that the EOM are derived from a system energy function. However, the energy function is spatially discretized, using a set of simplified DOF. As a result there is a systematic method for determining the DOF and resulting EOM. Further, for a linear structure, the system dynamics reduce to a set of constant matrix equations. For geometrically nonlinear structures, there are extensions which can account for joint rotations (although they are not widely used).

FEM has the advantage of being widely available, in a variety of packages. There is a large body of understanding of how to represent many engineering structures. The main disadvantage is that the simple shape functions used in the spatial discretization cannot accurately capture the strain distribution in a complicated structure unless many are used. The resulting model is much larger than necessary and must be truncated to a lower order in order to allow the control design algorithms to succeed. Further, the matrix equation form of the solutions can obscure the physical nature of the forces being modeled. Again any structure may be lost to the user of the model.

Simplified modeling packages such as Simulink are engineering modeling packages in which rigid body modes, flexible states, and nonlinear blocks such as friction, are combined to produce the correct input/output behavior. The advantage is that the package is a high-level system, allowing rapid generation of models. Many nonlinear elements are simple to incorporate. The major disadvantage is that the states of the model are not related to physical quantities. As a result, the simplified model is only as good as the engineering insight used to create it. This makes it problematic to generate a model without physical measurements to judge it against. If such measurements are available, it may be difficult to use them to tune the model parameters.

The clear implication is that modeling of geometrically nonlinear structures is still a fertile area to research. Two primary requirements emerge from the study of common methods in use at the present time. First, the structure of the manipulator dynamics is in general not preserved. The mass matrix for a geometrically nonlinear structure is a function of a set of rotation matrices. These rotation matrices are, for example, the rotations of the joints for a revolute joint robot, and may also be revolutions due to flexible deformations. These rotation matrices multiply into the linear mass matrices for each link of the robot. As a result the mass matrix is a complex function of the rotation variables (thus sines and cosines of the joint angles). A useful extension to any of the above techniques would be to separate out the rotation matrices which are the largest nonlinear terms in the EOM. In this form, the model would be directly amenable to control using modern state space methods. This will be discussed further.

The next point is that modeling effort is an issue; it is conceivable that a particular modeling method may not return an answer for a particular problem (time constraints may be a factor). There is a continuum between a simple formulation of the problem, with a computationally demanding evaluation of the dynamics (Lagrange) and a relatively complex formulation stage, but with a straightforward evaluation of the dynamics (FEM). In other words, the modeling effort may be front-end loaded, or may appear in the later stages of the modeling process. Model accuracy adds a third axis, so that modeling effort increases with accuracy. While there is no reason to believe that one can get off that continuum, there is reason to believe that for the manipulator modeling problem, there is a way to trade a moderate amount of additional work later (back-loading the effort) for a slightly more complex but still straightforward modeling formulation.

It must be stated that each of the above modeling methods has their place. It is not the intent of this work to advocate a specific choice for use in all cases. For this work, the Lagrangian framework will be used. The primary reason is that the derivation of the EOM remains in terms of physical quantities, which makes the terms in the EOM somewhat more intuitive. In addition, an assumed modes model can be fairly accurate with a fairly low system order. This will be important when the model is used for control design. The choice of the energy method over Kane's method is primarily one of computational simplicity - as will be shown, evaluation of Lagrange's equations can be highly automated. It is noted, however, that to be of general use, a finite element framework would be strongly of interest. This would enable maximum utility to the largest number of potential users. All of the concepts used in the modeling framework could be applied to a suitable finite element model.

1.4.2 Robust Control State of the Art

Robust control, and time-varying control strategies derived from it, have been well-studied in the literature. Of primary interest for the control of variable-configuration flexible structures are the time-varying control strategies in which the variations are measurable. This thesis will be specifically concerned with time-varying robust control; that is, the time variations of the plant are known to some degree, and used to modify the controller gains in real time. The alternative, using a single fixed controller for the entire range of configurations, has the disadvantage of achieving lower performance at a given configuration, due to the necessity of being stable over the entire workspace.

Several general categories of time-varying control have been studied in the literature. These include:

(i) *ad hoc* time-varying implementations of fixed gain controllers

(ii) adaptive control

(iii) feedback inversion

(iv) variable structure/sliding mode control

(v) linear robust control for time-varying parametrically uncertain systems

Each of these broad categories encompasses a range of ideas.

Ad hoc time-varying control is generally built around a linear controller, such as Linear Quadratic Regulator (LQR) or Linear Quadratic Gaussian (LQG). A set of controllers is designed using standard techniques, for a set of operating points, which is deemed to give suitable stability margins and performance. Time variation of the controller is achieved by combining the controller gains[41] or the controller outputs[62], or by using a nonlinear state observer with constant gain regulator and observation matrices[2].

The principle advantages are the availability of LTI design tools, and the need for only a few, LTI design models. Difficulties include the lack of techniques for showing stability for the combined system, necessitating exhaustive simulation to determine stability; lack of techniques for determining how many design models are needed, resulting in a highly iterative and labor-intensive design process; and difficulties with determining how to combine controller parameters stably.

Adaptive control uses real-time error estimates to tune controller parameters. Classes of adaptive control include direct, and indirect. Direct adaptive control tunes the controller parameters to decrease a measured error. Indirect control tunes an adaptive plant model, which is then used to derive a controller.

The primary advantage to adaptive control is that an exhaustive modeling and model updating procedure is not needed. In effect, it is automated. Secondarily, the self-tuning nature of the control means that unexpected changes (such as a new payload being added) can be accounted for on-line (an off-line controller redesign is not needed). The primary difficulty is that of showing stability of the closed loop. The two general means are to show the existence of a Lyapunov function, which is non-trivial and often depends on restrictive assumptions such as collocation of the sensor and actuator (rarely the case in robotics since the objective is to control one end of a link with a torque at the other end); and via exhaustive simulation. Since the entire controller can vary, exhaustive simulation may not be feasible.

Feedback inversion techniques use time-varying state transformations to form an LTI input/output representation for the nonlinear system, by canceling nonlinear dynamics. The transformed system can then be controlled with a simple proportional-derivative (PD) controller.

Far and away the biggest restriction is that there is no systematic approach to determining the proper state transformation. In addition, as in all inversion techniques, exact knowledge of the system parameters is crucial. Stability robustness to parameter uncertainty has only begun to be addressed. Finally, the robotics problem is typically non-minimum phase, leading to an unstable inverse for which work-arounds must be sought.

Variable structure/sliding mode techniques attempt to force the system states onto a sliding manifold; that is, a hyperplane in the state space which represents some desired trajectory. Once the state is on the manifold, it is in *sliding mode*, and will generally be forced to move toward the origin of the state space.

Variable structure controls offers the advantages of robustness to certain kinds of modeling errors (of interest to the robotics engineer is friction). Again, stability robustness remains to be addressed. In addition, the ability to control flexible modes is a topic which requires more study.

Linear robust control is perhaps the most highly studied category, since it allows for many interesting structures to be represented faithfully. The system is represented by a linear state space system matrix, and an unknown, uncertain, or time varying element. "Unknown" elements refer to unstructured uncertainties, where nothing is known except the frequency-domain characteristics. These are of little use to the configuration dependent system. Structured uncertainties refer to parametric variations in the system model; that is, a nominal value for a parameter such as a mass is known, but the actual parameter may have a fixed error (an "uncertain" parameter). With proper choice of representation, the uncertain element may in fact be allowed to vary with time; this is of particular interest to the robotics engineer. Within the parametric uncertainty field, many analysis frameworks exist; small gain, polytopic, Lyapunov, and so on. In many cases connections exist between the frameworks. This is interesting because it also creates the hope that, given a model parameterization which is suitable for a given control technique, simple manipulations can re-parameterize the model into a form suitable for many other control design techniques.

The advantage to linear robust control is the proven ability to control high-order flexible systems. Also, since the system model is linear, numerical matrix algebra tools can be used; the computational burden can be shifted to a computer, leaving the controls designer to contend with higher-level criteria, and avoiding complex variable math, Lie derivatives, and the other tools of the nonlinear control world. Finally, linear design tools, such as frequency domain specifications, are second nature to most controls engineers.

The primary disadvantage to linear robust control is conservatism, in which a controller may be returned which sacrifices too much performance in return for a guarantee of stability (or no controller may be found at all). Specifically, conservatism results from the inability to create a linear system model which accurately captures the relationships in the nonlinear dynamic body. For example, the time-varying parameters may be system states. This connection is lost in most design algorithms.

Included in this assessment is the fact that a suitable parametric representation of the dynamic system may not be simple to derive.

1.4.3 Flexible Multibody Dynamics for Control

The flexible multibody dynamics modeling problem is a fairly mature field. Applications exist in many areas, thus much insight into the nature of the field is available in the literature. Control of such systems is a relatively new field. However, in the 20 years since the Space Shuttle Remote Manipulator System (SSRMS) came on-line, bringing the issue of control of flexible manipulators into the forefront, a reasonable database of what works in a laboratory setting has been built up.

A useful cut into the problem of control of flexible multibody dynamics may be taken by listing the instances of experimental work in the literature. These include

(i) LQG[35, 44, 63, 2, 40, 43]

(ii) adaptive variable structure control[55]

(iii) gain scheduled LQR[41]

(iv) rigid body feedback linearization with input preshaping[53]

(v) acceleration feedback[50]

(vi) Multiple Model LQG[49]

(vii) indirect adaptive control[64]

The preponderance of experimental results take the form of ad hoc implementations of LTI control. In addition, with the noted exceptions, all experimental work was performed in a 2D, planar configuration. The noted advantages and disadvantages of each control category notwithstanding, the primary difficulty in control for configuration-dependent flexible systems is the sheer complexity of generating controls solutions for realistically sized control problems. As of this writing, ad hoc solutions offer the best way to meet performance and stability criteria. However, ad hoc solutions fail for more complex systems (such as three-dimensional arms) because the modeling and control design effort increases dramatically.

On the other hand, the most notable absence from the above list are time-varying robust controllers. This is especially true since the problems being addressed by such control design algorithms often explicitly include flexible dynamics. In many cases these take the form of extensions to the LTI controllers which have proven to work quite well. Thus there is the implication that such methods will offer good performance at a reasonable design cost. The primary barrier which has so far halted the implementation of time-varying robust control is the lack of deriving a properly parameterized model for the design of said controller. Typically such a form is assumed in the controls literature[29]; however, few papers in the modeling literature address the necessary parameterizations. In the few cases where the issue is addressed, ad hoc modeling approaches such as curve fits to Lagrange model system matrices are used to derive the parameterized model[47].

Given such a model, robust control offers the hope of designing a single controller which offers performance and stability guarantees with a finite but useful degree of conservatism. Further, with the amount of research in this area, driven by such fields as robotics and aerodynamics, many of the current disadvantages of the robust control framework for time-varying control may be ameliorated in a reasonable time frame.

1.5 Thesis Roadmap

The thesis is organized as follows: the potential for space-based manipulator performance improvement is investigated in Chapter 2. In Chapter 3, a particular Time-Varying Robust Control approach is identified from the literature and developed. Next, motivated by the need to generate a suitable representation for application of the control algorithm, a nonlinear flexible multibody dynamics modeling algorithm is presented in Chapter 4. In Chapter 5, a model for a flexible manipulator link, suitable for the aforementioned dynamics modeling approach, is created. In Chapter 6, the modeling and control algorithms are verified on a high fidelity simulation of the

SRMS. Finally, in Chapter 7 the conclusions of the thesis, and recommendations for future work, are presented.

Chapter 2

Benefits of Control

The field of flexible robot control has received increasing attention in recent years. Space based robotic systems, such as the Shuttle Remote Manipulator System (SRMS), are prominent drivers of this research; up to 30% of SRMS mission time is estimated to be spent waiting for the arm to settle out. Numerous linear and nonlinear control algorithms have been investigated to reduce the impact of arm flexibility. In many cases, experimental results have demonstrated the possible performance improvements. The next step is to determine:

(i) Can these algorithms be applied to the SRMS?

(ii) What are the benefits?

(iii) What are the associated costs?

An approach to answering these questions is in the form of a comparison between the various approaches. The present chapter will examine the results of such a comparison. It will be shown that gain-scheduling achieves the best overall performance among the controllers tested. However, certain deficiencies in the gain-scheduled controller will be noted, in the areas of stability, real time computation requirements, and design effort. The results motivate the search for a controller which retains the performance advantage of gain-scheduling, while overcoming the noted shortcomings. The remainder of the thesis will detail such a controller.

This chapter will detail such a comparison on a simulation of the SRMS. To begin, the basis for comparison is defined. The candidates are selected. The SRMS simulation is described. Then the results are presented and analyzed. Finally the possibility for extension of the results to the SRMS itself is examined.

2.1 Cost versus Performance Comparison

It then becomes natural to ask, "How much will these controllers cost?". That is, what is the cost incurred by each controller for additional sensors, computing power, et cetera, and what is the relative performance. The resulting cost versus performance curve would be useful in selecting, from among the candidates, which best meets the needs and financial constraints of the end users.

The present chapter discusses a preliminary comparison of advanced controllers on a simulation of the SRMS. The objective of the research was to demonstrate the performance improvement possible by using advanced controllers on the SRMS. Additionally, some insights on the relevant metrics of cost for an SRMS-like system were desired. The approach to reach the objectives began with creating a general 3D flexible manipulator modeling capability. The formulation was used to create a linear model of the SRMS suitable for control design, and a nonlinear model suitable for performance analysis. Then, a test matrix was defined to capture the performance requirements of the SRMS in a form which could be tested in simulation. Cost metrics were also defined. Controllers were designed on the linear model, then tested on the nonlinear simulation. The chapter concludes with some observations about advanced control for flexible manipulators, and with a discussion of future work.

2.2 Flexible Arm Modeling

The project required an accurate 3D flexible arm modeling capability. There are pre-existing models such as the Draper RMS Simulator (DRS), which is a validated simulation of the SRMS. There are also computer packages which allow general mod-

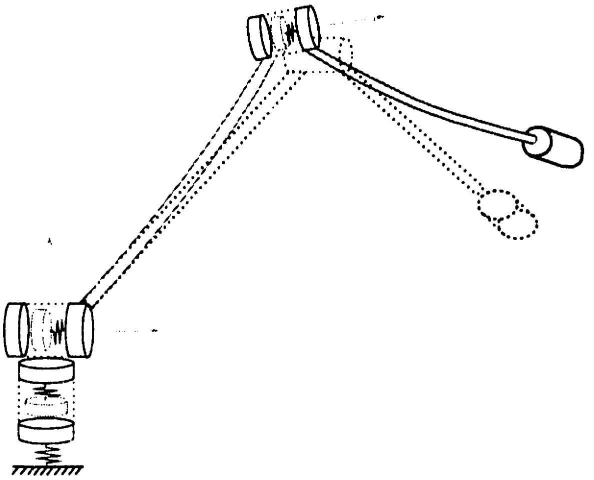

Figure 2-1: 3D manipulator model showing lumped stiffness at base and gears, lumped mass at armature, gearbox, and payload. Links are modeled with assumed modes. Arrows mark rotation axes of the three joints.

eling of 3D flexible-chain bodies, such as Simulink and TREETOPS. Such models can run time simulations of nonlinear plants, as well as provide linearized models suitable for control design.

It was decided to develop an in-house modeling capability. This approach allowed flexibility in tailoring arm parameters. The modeling formulation which was used allows the derivation of the sensitivities of the equations of motion, which will contribute to model updating. Model updating will be crucial to the design of stable high performance controllers. Ease of parameter tailoring will also facilitate the design of the hybrid arm. The method also contributes to control design, by producing a parameterized linearized model. The parameterized model allows controllers to be designed and tested at arbitrary arm configurations and payloads, without requiring re-linearization of the equations of motion.

The modeling procedure uses a recursive Lagrangian formulation of the fully 3-dimensional flexible manipulator equations of motion (EOM). The EOM are derived from the Lagrange equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \sum_{i=1}^{N} \vec{F}_i \frac{\delta \vec{r}_i}{\delta q} = F \tag{2.1}$$

where $L = T - V$, T is kinetic energy, V is potential energy, F_i are nonconservative

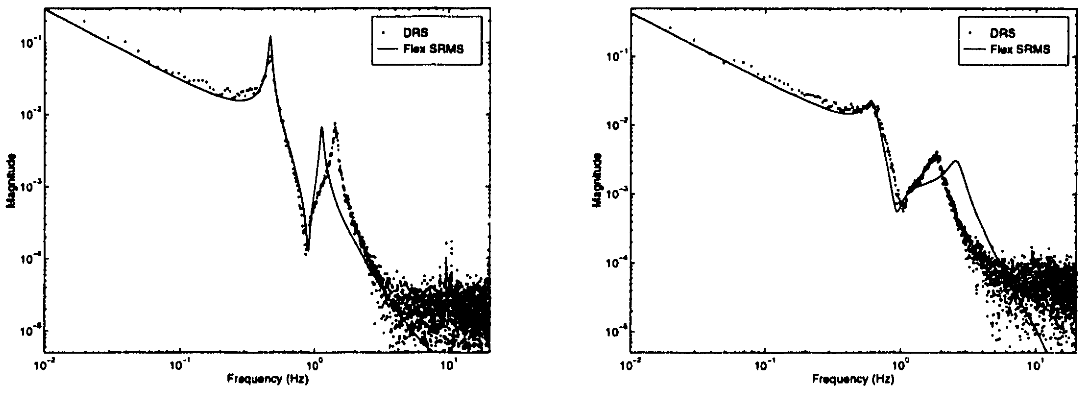

Figure 2-2: SRMS model (solid) vs DRS (dotted) at (0°, 90°, −90°): shoulder yaw to tip Y (left), elbow pitch to tip Z (right).

forces, and q are the modal degrees of freedom. As the model order increases, the complexity of Equation 2.1 increases drastically. The recursive formulation splits Equation 2.1 into a sum over two flexible bodies, plus a constraint equation that describes how the motion in body 2 couples into the motion in body 1:

$$
\begin{aligned}
L(q_1, \dot{q}_1, q_2, \dot{q}_2) &= L_1(q_1, \dot{q}_1) + L_2(\dot{q}_c, q_2, \dot{q}_2) && (2.2) \\
q_c &= G(q_1) && (2.3)
\end{aligned}
$$

where q_1 and q_2 are generalized coordinates for body 1 and body 2, respectively, and q_c are the constraint degrees of freedom (for example the displacement and rotation at the end of the first link). Each body can be analyzed separately to find the components of the EOM:

$$
\begin{aligned}
\frac{d}{\mathrm{d}t}\frac{\partial L_1}{\partial \dot{q}_1} - \frac{\partial L_1}{\partial q_1} + \nabla_{q_1} G^T \frac{d}{\mathrm{d}t}\frac{\partial L_2}{\partial \dot{q}_c} + & \\
\left[\nabla_{q_1} \dot{G} - \frac{\partial \nabla_{q_1} G \dot{q}_1}{\partial q_1}\right]^T \frac{\partial L_2}{\partial \dot{q}_c} &= F_1 && (2.4) \\
\frac{d}{\mathrm{d}t}\frac{\partial L_2}{\partial \dot{q}_2} - \frac{\partial L_2}{\partial q_2} &= F_2 && (2.5)
\end{aligned}
$$

where $\nabla_{q_1} G$ is the gradient of the constraint with respect to q_1. The first two terms in Equation 2.4 are the free-body EOM of body 1, and the second two are coupling terms. Equation 2.5 is the free-body EOM of body 2. To analyze multiple bodies, the first two are analyzed and assembled, then treated as the new body one. The next body is then analyzed as body two (with its constraint equations dependent on both the previous links).

The nonlinear EOM are then converted to Fortran format and incorporated into a Fortran routine for numerical integration. The payload mass and inertia are left in symbolic form, to represent any rigid payload which the SRMS can carry. The symbolic EOM are also linearized in Mathematica, creating a linear model parameterized by joint configuration and payload. The linearized model is ported to a Matlab M-file for control design.

For the work presented here, the recursive formulation was used to create a full three-dimensional model of the SRMS. The model included link bending and torsion degrees of freedom, joint and base lumped stiffnesses, and payload mass and inertia (Figure 2-1). Two bending modes per direction and one torsion mode per link were included, to capture the SRMS dynamics to about 20 Hz. The Fortran code calculated joint forces, consisting of Coulomb friction, motor back-EMF, and a simplified model of the SRMS rate controller which included torque limiting (saturation). By changing physical parameters, the same model can represent the Space Station Remote Manipulator System (SSRMS), the proposed mid-deck manipulator, and ground-based test hardware.

The SRMS model was verified against the DRS. Figure 2-2 compares the SRMS model to the DRS. The left axis plots the transfer function from shoulder yaw torque to tip lateral displacement for the SRMS model (solid) and the DRS (dotted). The arm configuration is $(0^\circ, 90^\circ, -90^\circ)$, where the first angle is shoulder yaw, second is shoulder pitch, and third is elbow pitch. First mode is captured within a percent, and there is a 20% mismatch in second mode. The overall gain and features of the transfer functions are very close. The right axis plots the transfer functions from elbow torque to tip vertical displacement. As before, the overall features are close.

Given the modeling simplifications used in the SRMS model, the degree of match was fairly good. Further, the first mode, which the was captured very well, has the greatest impact on arm performance. Since the DRS has been shown by extensive flight testing to be an accurate model, the SRMS model was considered acceptable for the preliminary testing.

2.3 Controller Design

In the fifteen years since the original SRMS control design, many approaches have been proposed. Both feedforward and feedback schemes have been investigated. The primary advantage of feedforward control is that stability is not an issue. However, there is no possibility of rejecting unknown disturbances. On the other hand, feedback control can improve performance in the presence of unknown disturbances, but stability is an issue. It has been suggested that it may be best to use these complementary techniques in conjunction. However, research into how to combine the techniques is not well-developed. The results presented here will consider only feedback control.

Proposed flexible manipulator feedback control techniques include most of the currently available linear and nonlinear design tools. Broadly categorized, they fall into four groups, based on the degree of nonlinear information used in real time:

1). linear robust control, in which geometric nonlinearities are framed as dynamic uncertainties[4, 65].

2). adaptive control, typically by estimating plant and/or controller parameters[2, 3, 44, 38].

3). output linearization/plant inversion, sometimes applied to the "rigid body" degrees of freedom, with linear controllers to account for flexibility[8, 52]. Singular perturbation is a formal approach for coupling the linear and nonlinear controllers[5]. Tracking has also been achieved by inverting outputs with stable zero dynamics[66].

4). gain-scheduling, often using linear controllers designed at a finite number of set points which are interpolated across the workspace[6, 18]. More recent formulations return a single, parameterized controller[28].

These categories are arranged in order of increasing real-time usage of knowledge of the nonlinearities. Linear control uses knowledge of parameter changes in the design process, but the controller dynamics are fixed. Adaptive control assumes a structure for the model, but does not use known information. Many of the nonlinear techniques use nonlinear control for rigid body states which are measurable, but use linear controllers for the flexible states. Gain scheduling allows controllers to be tailored for both flexible mode and rigid body (inertia) changes. These approaches were down selected to arrive at a reasonable number for comparison.

The preliminary down select for feedback controllers included ratings in four main categories. First and foremost, maturity, with experimental results showing application to two-link manipulators. Second, formulation complexity, to ensure that the derivation of controllers for the three-dimensional arm would be tractable (analytically and numerically). This was a concern since most studies have been done in only two dimensions. Third, implementation complexity, which included real-time computational requirements, special actuation hardware (e.g. link piezoelectric actuators), and number and type of sensors (where simple sensors such as link strain are rated higher than sensors such as 3D tip position). Finally, the ability to assess closed loop stability was considered.

The results of the down select were a linear robust control technique, a variation of rigid arm feedback linearization which used a robust linear outer loop controller, and a gain-scheduled controller. These selections captured three of the four categories shown above. Adaptive control was not selected, primarily due to the problem of showing controller stability. The SRMS rate controller was used as a baseline for comparison. It was not designed to add damping to the system, but to reject flexible mode forces at the joints[67]. However, it can be evaluated as a separate controller to represent the performance of the current system.

Note that the objective of the research is *not* to show that each controller is the "best" within its category. For example, linear robust control design tools include robust pole placement, robust $\mathcal{H}_2$, $\mathcal{H}_\infty$, μ-synthesis, and so on. The performance of the linear controller need only be representative of that group. Rather, the axis this work explored was down the groupings, in the direction of increasing real-time knowledge of the nonlinearities.

The advanced controllers were all designed as outer loops to the SRMS rate controller. This would be done on the Shuttle to ensure a safe mode in the event of control system component failure, such as sensor failure. In addition, the rate controller linearizes out joint friction, which can limit advanced controller performance. They were designed to use the same sensor suite, consisting of angle encoders for each joint and one strain gage per link to sense flexible mode response. The strain gage is necessitated by the high gain on the SRMS rate controller, which makes link modes unobservable at the joints. The metric used in the design process was average settling time across the workspace, as evaluated on the linear model. The closed loop eigenvalues of the linear model were used to assess stability of the closed loop system across the workspace.

2.3.1 Linear Controller

The linear robust controller was designed using a combination of Sensitivity-Weighting and Multiple Model (MM) robustification techniques[16]. The design procedure minimizes the $\mathcal{H}_2$ position error of the end-effector. Sensitivity Weighting augments the LQG weighting matrices to improve robustness. Multiple Model design applies the same compensator to the arm at several angles:

$$J = \sum_{i=1}^{N} \alpha_i \int_0^\infty \left[\begin{array}{c} x_i^T u^T \end{array} \right] \left[\begin{array}{cc} R_{xx} & R_{xu_i} \\ R_{xu_i}^T & R_{uu_i} \end{array} \right] \left[\begin{array}{c} x_i \\ u \end{array} \right] \mathrm{d}t \tag{2.6}$$

where x_i are the plant states, u are the commands to the rate controller, R_{xx_i} and R_{uu_i} represent the desired performance level and control effort, respectively, at the design point θ_i. R_{xu_i} is a cross-weighting. There is a coupled estimator problem which is not reproduced for conciseness. Each plant model is created by evaluating the linearized model at a different configuration:

$$\dot{x}_i = A(\theta_i)x_i + B(\theta_i)u \tag{2.7}$$

$$y_i = C(\theta_i)x_i + D(\theta_i)u \tag{2.8}$$

The design models consist of the eighteen states in the pitch axes (nine generalized coordinates).

The controller is designed using a continuation procedure in which a Sensitivity Weighted controller is designed for an initial angle. It is fed to the MM numerical design procedure, with the nominal model and models on each side which are near instability. As the MM algorithm stabilizes the outer models, they are moved to greater angles. Additional models are placed at intermediate angles where modes begin to go unstable. Once the controller is stabilized for the entire workspace, the state weights (the R_{xx_i}'s) and α_is are iteratively tuned to give the best time domain performance.

A drawback of this design procedure is that stability is only guaranteed for a linear time invariant (LTI) plant, and is only guaranteed at the design points. In order to stabilize the nonlinear robot model for all intermediate angles, many design points may be needed. This will increase the time needed for the controller calculation. The controller designed for the SRMS simulation using this technique will subsequently be referred to as MMLQG or the MMLQG controller. It had 12 states, and was designed on five models.

2.3.2 Approximate Feedback Linearization

The feedback linearization approach follows a procedure suggested by Khorrami[53]. The exact feedback linearizing controller for the flexible arm is not realizable without bending and torsion actuation in the links. The Approximate Feedback Linearization (AFBL) approach involves zeroing out the flexible terms in the equations of motion to get an "equivalent" rigid manipulator under independent joint rate control:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) = K_d\left(u - \dot{\theta}\right) \tag{2.9}$$

where the rigid states θ are joint angles, and K_d is the diagonal matrix of rate gains. The exact feedback linearizing controller for the rate controlled rigid system is realizable as[68]:

$$u = \dot{\theta} + K_d^{-1}\left(M(\theta)v + C(\theta,\dot{\theta})\right) \tag{2.10}$$

Note the positive feedback of $\dot{\theta}$ and the K_d^{-1} factor in Equation 2.10. The effect of this term is to "undo" the rate control so that torques instead of rates are commanded by the feedback linearizing controller. Applied to the rigid arm, Equation 2.10 creates a system whose poles are integrators driven linearly by the new input v. The AFBL method applies this controller to the flexible arm. This may reduce the dependence of dynamic frequencies on configuration[53]. Note also the term $C(\theta,\dot{\theta})$ which for the SRMS is primarily the Coulomb joint friction model. Joint friction proved to be a strong contributor to flexible mode vibration, since it prevented disturbances from getting to the joints to be damped by the servos. Under rate control, the high gain linearizes joint response in the presence of Coulomb friction. Under AFBL control, the joint friction term had to be identical to the SRMS model's joint friction to achieve good performance. An accurate friction model will be difficult to achieve in practice.

A linear robust controller (solved using Equation 2.8), was designed around the arm and feedback linearization controller to produce the input v. This controller will be referred to as AFBL. It was designed using the continuation procedure, using four plant models, and also had twelve states.

2.3.3 Gain-Scheduled Controller

The gain-scheduled controller consists of several individual MM/SW LQG controllers, again solved using Equation 2.8, designed for the linearized arm at several design points. The controllers are run in parallel. Between design points, the command to the arm is generated from a linear interpolation of the commands generated by the nearest controllers:

$$u = \frac{\theta - \theta_i}{\theta_{i+1} - \theta_i}u_i + \frac{\theta_{i+1} - \theta}{\theta_{i+1} - \theta_i}u_{i+1} \tag{2.11}$$

where the design points are at θ_i and θ_{i+1}, and u_i and u_{i+1} are the commands generated by the controllers at those design points (the structure was suggested by an adaptive aircraft flight controller described by Athans *et al.*[62]). Because each linear controller need be stable for a smaller range of angles (θ_{i-1} to θ_{i+1} rather than +2° to −161°), each controller can achieve higher performance. The gain-scheduled controller will be denoted as GSLQG. It consisted of six controllers designed at +2.4°, −40°, −80°, −90°, −120°, and −140°. Each is twelve states.

The controllers were discretized and written to Fortran code, which was interfaced with the nonlinear model.

2.4 Test Matrix

Once the controllers were designed and validated, the test matrix for comparison was developed. Three considerations drove the selection of test moves:

1). minimize the number of free variables.

2). results should be as general as possible (maximum traceability to the SRMS).

3). test moves should capture the essential features of operational moves.

These factors drove the preliminary testing to consider moves only in the pitch axes, with a single payload. This configuration captured the fundamental geometric nonlinearity of flexible manipulators, while resulting in dynamics which vary with a single parameter, elbow angle. The specific moves during test were selected by considering operational tasks of the SRMS.

The operational tasks of the SRMS break out into three main groupings. Tasks like inspection, payload deployment, or payload retrieval and berthing in the bay, require an accurate path to be followed. Tasks such as payload capture or parts mating during construction require accurate placement of the tip. There is another group of tasks which require rejecting tip disturbances, such as support an astronaut performing manual assembly. This latter group was not considered in the tests being reported.

The path following performance of the closed loop system was evaluated by performing a 25 foot vertical move out of the payload bay. The root mean square tip position error was used as a performance metric. The mating performance was assessed by measuring tip rise and settling times in response to a step command, at many points through the workspace. Rise and settling times were calculated as the period from the initial command to, respectively, the the end-effector entering, and remaining within, a ball centered around the desired position. The ball had a radius of 2% of the total move distance. A .05° step command was used, to avoid saturating the motors. All controllers were stable for larger steps, but the torque limiting became the dominant effect in determining rise and settling times.

It was desirable to choose appropriate metrics for hardware, software, and human factors costs. Hardware costs would normally include additional sensors. Since all controllers in the current test use the same sensors, the figure is not relevant here. Computational requirements also add hardware costs, representing GPC usage or additional computer hardware needed by each controller. Computer requirements were quantified using the number of operations per second executed by the controller code. Since the absolute numbers are only meaningful for a given computer platform, they are normalized by the MMLQG requirements to indicate usage relative to one another. Software costs are expected to be primarily for debugging and validation, and should be roughly proportional to the number of lines of code. Human factors costs are nearly impossible to assess in a simulation study. The only cost quantified was time delay. This plays a role in perceived arm responsiveness. The effect is quantified in the rise time metric above.

These cost metrics were less than perfect, in that they did not allow the absolute cost of each controller to be assessed. In the ideal case, a scale factor for each metric would be found, to allow the total cost to be considered. For example, scale factors of $\$X$ per line of code and $\$Y$ per instruction per second would allow the costs to be combined and the total cost of each controller to be predicted. Nonetheless the numbers do give an indication of relative costs.

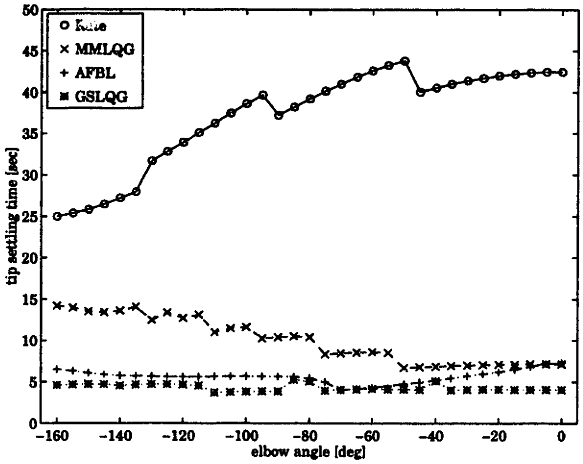

Figure 2-3: Settling time versus elbow angle for the tested controllers.

2.5 Simulation Results and Discussion

This section presents the results of the test matrix performed on the SRMS simulation. For conciseness, the terms rate, MMLQG, AFBL, and GSLQG will refer both to the controller and to the closed loop system with that controller applied. The meaning will be evident from context. Figure 2-3 shows the settling times for the feedback controllers as a function of elbow angle. Note that the SRMS joint coordinates are defined such that 0° is straight, and a bent arm corresponds to a negative angle. The rate controller settling times are shown with circles. The MMLQG times are plotted as "x"'s. The pluses show the AFBL controller performance, and the asterisks show the GSLQG controller. Discontinuities occur where slight changes in damping ratio cause a cycle which is just grazing the target ball to move across the boundary.

The rate control settling times decreased substantially as elbow angle increased. The MMLQG controller performed best near full extension, where it matched the AFBL controller, with settling times increasing with elbow angle. AFBL had fairly consistent performance across the workspace, but is only as fast as GSLQG near −70°. GSLQG had the best and most consistent performance.

These results can be explained by looking at the tip position error at representative angles. Rate, MMLQG, and AFBL control showed significant changes in performance between −20° and −160°. Figure 2-4 is a plot of tip position errors in end-effector X

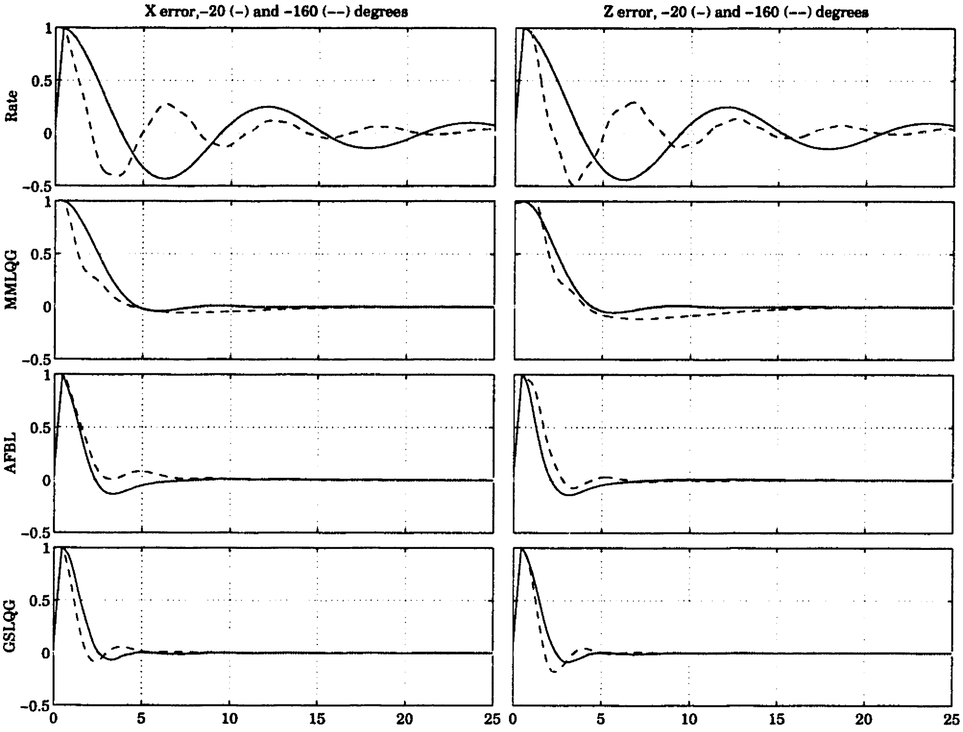

Figure 2-4: X and Z tip position errors, for rate, MMLQG, AFBL, and GSLQG, at $-20°$ (solid) and $-160°$ (dashed).

and Z (on the left and right respectively), under (from top to bottom) rate control, MMLQG, AFBL, and GSLQG. The solid line shows the error at $-20°$, and the dashed line the error at $-160°$. The 2% settling time criterion is shown by the shaded area.

The rate control error shows that settling time decreased with elbow angle because the flexible mode frequency increased. This in turn occurred because the effective inertia decreased. The damping ratio decreased at the same time, slightly negating the gains. The LQG controller at $-20°$ had a frequency of about 0.11 Hz and nearly optimal overshoot. At the larger angle, the overshoot and slow settling indicate a much lower frequency dominant mode. The AFBL controller at $-20°$ did not have a single dominant mode; there was a 0.22 Hz resonant mode and a real pole. At $-160°$, there is larger overshoot but the frequency has not significantly increased. The GSLQG plots show damped resonances from 0.19 Hz (at $-20°$) to 0.29 Hz (at $-160°$), however the envelope of the responses remains similar, resulting in similar settling times.

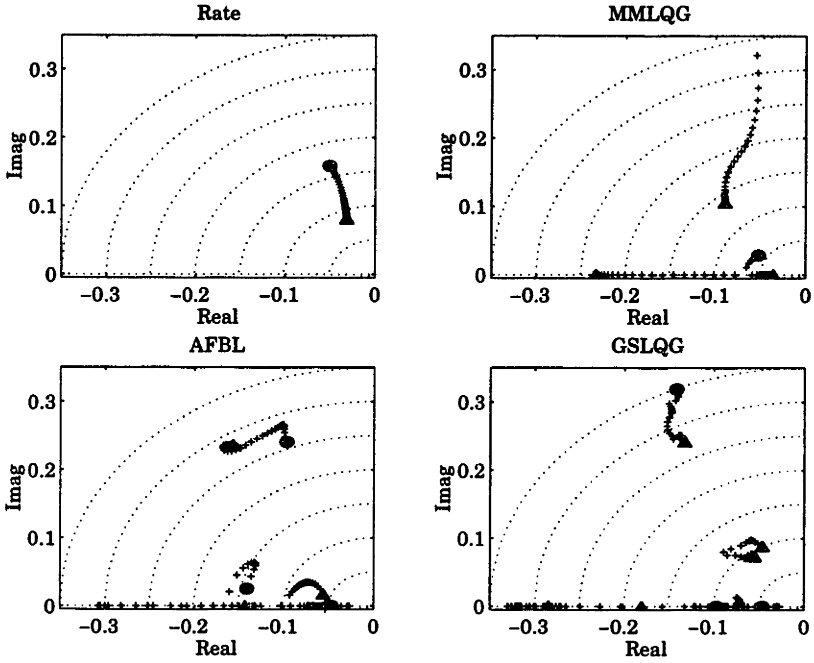

Figure 2-5: Root loci, as elbow angle varies, of the linearized closed loop system under rate control (top left), MMLQG (top right), AFBL (lower left), and GSLQG (lower right). Roots at −20° are triangles, −160° are circles.

An analysis of the closed loop poles of the linearized model can also help explain Figure 2-3. In Figure 2-5, the loci of the dominant poles of the controlled system are plotted for 5° increments in elbow angle, from 0° to −160°. The pole frequencies are in Hz to enable comparison with Figure 2-4. Dotted lines show constant natural frequency. Clockwise from the top left are the root loci of the system under rate control, MMLQG, AFBL, and GSLQG. On each curve, poles at −20° are shown with triangles, and at −160° with circles.

The rate control poles clearly show the change in frequency and damping ratio with elbow angle. The time constant increases slightly as elbow angle increases, bending the locus to the left. This corresponds to the decrease in settling time evident in the time response. The MMLQG root loci are more intricate, showing two branches corresponding to two pole pairs. The first branch goes vertically up as angle increases. The second remains on the real axis except near maximum elbow deflection where the poles coalesce into a complex pair. The first branch contributes the resonant response and fast rise time seen in the time response. The second branch contributes a very slow component to the response, whose magnitude varies with elbow angle. At small angles, the slowest real pole lies near a real transmission zero in the command to tip position transfer function and is therefore almost unobservable. As angle increases, the real pole moves away from the zero (and eventually into the complex plane), which increases its observability and thus increases the tip position settling time.

The locus for the AFBL controller is yet more intricate. There are now three branches. The first branch is isolated in the complex plane, while two pairs of modes split and coalesce along the real axis. The additional branch is due to the positive rate feedback in Equation 2.10. The rigid body poles of the plant which the rate controller had pushed out to higher frequency, are now within the bandwidth of the AFBL controller. The first branch, which contributes the fast resonance to the time response, is leftwards and upwards compared to the corresponding MMLQG branch. This gives AFBL higher damping and a faster settling time. At the same time, poles on the two branches near the real axis remain "near" transmission zeros (close enough to make them unobservable in the output).

The GSLQG locus shows even more complexity. There is a branch far into the left half plane, corresponding to the complex branches in MMLQG and AFBL. A second complex branch clusters near 0.1 Hz, and two branches lie near the real axis. The first branch is the dominant component of the time response. It moves vertically with elbow angle, showing how the time constant, and thus the settling time, remain fairly constant, while the damped frequency changes. The 0.1 Hz branch moves with the locus of a complex zero pair, making it unobservable. Again, the real axis zeros are close enough to real zeros to make them unobservable also.

Taken in aggregate, the four root loci in Figure 2-5 show how increased controller complexity allows finer ability to locate the closed loop poles. The rate controller poles do not vary drastically, but the controller is not pushing for the best performance. The MMLQG poles vary considerably with elbow angle. The AFBL poles cluster more tightly, showing that AFBL has better ability to place the closed loop poles.

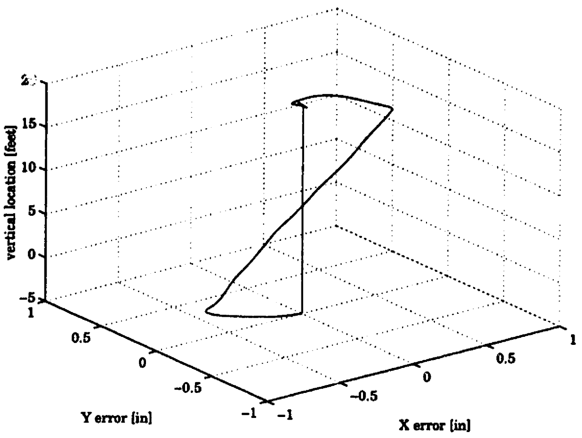

Figure 2-6: Tip trajectory of the 3D model, under GSLQG control.

The GSLQG poles show the tightest clustering. The ability to place the closed loop poles in the face of elbow angle changes is crucial to achieving good tip positioning performance. In order to achieve good average performance, the MMLQG controller necessarily has too small a gain at small angles and too large a gain at larger angles. Additionally the compensator dynamics must be robust to large plant pole shifts. The AFBL controller compensates for inertia changes, but since the dynamics are fixed, it cannot move closed loop modes on top of zeros - it must add damping to all modes. The GSLQG controller can vary its dynamics, so it can place closed loop modes over zeros to make them unobservable. This allows higher damping and increased frequency in the observable modes.

For the sake of completeness, the controllers were applied to the 3D nonlinear model for comparison with the 2D model. Only the pitch degrees of freedom were fed back. Due to the complexity of the model, the step response test would take significant computation time and was not repeated. The slew test was repeated, and the performance agreed with the 2D nonlinear simulation. The performance during the slew test is plotted for the best controller, GSLQG, in Figure 2-6. The vertical axis is distance in feet over the payload bay. The horizontal axes are tip position errors in inches. Note the gain-scheduled controller keeps the tip within one-half inch of the desired path.

Table 2.1: Feedback Controller Performance and Cost on the SRMS Simulation

Controller	t_s [sec]	t_r [sec]	e_{rms} [in]	# lines	R_{comp} [sec/sec]
Rate	37.2	5.8	16.10	0	0.0
MMLQG	10.0	6.5	1.25	181	1.0
AFBL	5.6	3.9	0.65	212	1.3
GSLQG	4.3	2.4	0.58	512	4.0

2.6 Summary and Discussion of Results

The test results are summarized in Table 2.1. The performance results are reported as average settling and rise times in seconds (t_s and t_r respectively) for the step inputs, and RMS error in inches (e_{rms}) for the path-following maneuver. Costs are reported as number of lines of code, and normalized real-time computational requirements (R_{comp}).

The settling times for the mating maneuvers show that performance improved as more information about the nonlinearity was used. The baseline rate controller settled in an average time of 37 seconds. The average settling time for the MMLQG controller was about 26% of the baseline time. AFBL performed about twice as well as MMLQG, averaging 15% of baseline. GSLQG averaged 12% of baseline.

Differences in rise time are not as dramatic. The MMLQG controller actually showed 12% slower response than the baseline. The AFBL controller was 33% faster than baseline, and GSLQG was 58% faster. These improvements were not inconsequential, but the baseline controller already had reasonably fast response. The rate controller is high gain, so the arm joints move rapidly to the correct angles.

The positioning error in the path following test also attests to the value of additional nonlinearity information. The baseline error was 16 inches (this was mostly lag error along the move direction). The MMLQG controller error was 8% of the baseline error. AFBL error was 4% of baseline. GSLQG was slightly better, at 3.6% of baseline.

These results show that the greatest performance is achieved for the greatest amount of real-time knowledge of the nonlinear terms. GSLQG had the highest performance, followed closely by AFBL. Both were roughly twice as good as the MMLQG controller. However, performance improvements are only part of the story. The cost metrics show a dramatically different aspect. The cost increase from MMLQG to AFBL was minimal, consisting of 17% more code and 1.33% more calculation. The cost of GSLQG was much greater than either MMLQG or AFBL, requiring 2 to 2.8 times as much code and 1.3-4 times as much real-time computation.

Based on these results, it is clear that given unlimited resources, a gain-scheduled controller offers the best absolute performance among the controllers tested. Unfortunately, there is a stiff penalty to be paid in terms of computer resources. If the goal is to maximize the performance achieved per unit cost, the AFBL controller appears to offer the best return.

2.7 Conclusion

Advanced control can improve the performance of structurally flexible manipulators. However, sensor and computer integration and software validation necessary to implement such controllers on space based manipulators such as the SRMS will be expensive. The present research has shown that there is an element of decreasing returns, and that the costs for a more advanced controller can increase drastically. The present investigation showed that a 30% increase in cost (from MMLQG to AFBL) gave a factor of 2 improvement, but implementing GSLQG at triple the cost of MMLQG gave only a factor of 2.2 improvement. This improvement may not warrant the extra expense. In general, it would be greatly beneficial to compare controllers, and quantify the cost as well as performance, so that the best controller per unit cost can be determined. One can picture a graph with performance on one axis, and cost on the other, and the various algorithms as data points.

In addition, only one testing axis was explored (degree of elbow angle information used in real time). In actuality, there are many other axes to be tested, such as different sensors and interfaces. For example, tip position feedback has long been recognized to be the best sensor, however, one can measure tip position several ways:

using vision sensors, integrated tip accelerometers, or distributed sensors. Each has different advantages and penalties. If the performance benefit of each sensor could be determined without actually installing it on the SRMS, a better use of resources could be made.

In summary, the gain-scheduled LQG controller is found to have the best performance of the controllers evaluated. However, the design approach which was used involved a number of setpoint controllers. This increased design effort, as many controllers were needed. In addition, a stable scheduling scheme must be created in an *ad hoc* manner. Stability for the closed loop must be tested in an exhaustive fashion using time simulations of the controller applied to the nonlinear plant. Finally, the controller real time computation requirements were increased by the multiple setpoint controls, which needed to be run concurrently.

The next chapter will examine an approach to gain-scheduled controller design which addresses these issues. First, a single controller is designed using linear algebra tools. Stability for the joint workspace is assured using multivariable stability criteria (specifically, small gain theory). Finally, real-time computation is reduced due to the availability of a parameterized, time-varying controller.

Chapter 3

Control Design

The previous chapter has demonstrated that there are performance advantages to be gained by implementing a gain-scheduled controller for geometrically nonlinear flexible structures. It then becomes necessary to examine how best to design such a gain-scheduled control. The present chapter addresses this topic. First, a suitable framework for gain-scheduled design is identified from the robotics control and robust control literature. Next, the appropriate description of the manipulator performance objective is studied. This motivates the development of a specific control design algorithm, which is the subject of the bulk of the chapter. Finally, the proposed method is evaluated on a simple example to identify its characteristics and determine its suitability for control of flexible manipulators.

3.1 State of the Art

Numerous approaches to the control of manipulators have been examined in the literature. Most of the experimental results for two link manipulators which are reported, involve the use of some type of modern linear state space design technique. Either a single robust controller is designed for the entire workspace[49, 2, 40, 4, 39], or a set of linear controllers are designed for various design points, and an ad hoc scheduling scheme is used to gain-schedule between the controllers[41].

Many factors contribute to the attractiveness of linear design tools for configuration-dependent systems. First, linear state space controllers have been shown experimentally to work well with high-order flexible systems. State space design tools offer a consistent and well-understood means of stating performance objectives; in other words, modern control design algorithms offer useful "handles" to the controls engineer. Also, linear techniques offer many useful analysis tools, including frequency domain tools, to determine stability as well as indications on how to modify the controller parameters to achieve stability.

Finally, control of parametrically uncertain systems has been widely studied in the linear controls literature. Configuration-dependent nonlinearities can be described as parametric uncertainties, opening up the entire field of linear robust control as potential candidates for design. The previous chapter has described one method for writing the EOM of the geometrically nonlinear structure in a linear form with a recognizable parametric "uncertainty" block. The present chapter will discuss approaches to determining the control design algorithm for such a system, with emphasis on a particular algorithm.

The primary disadvantage of such linear methods is the lack of a simple, systematic linear design tool to compute the single controller, or the set of controllers. Such questions as "how many controllers are needed" remain open. As a result, the design process is a highly iterative process. Typically an inner automatic iteration loop is used to solve for the robust controller. Then an interactive outer loop must be performed by the designer to determine if the number of set points and linear controllers is sufficient. There is no simple check to determine beforehand whether the resulting controller will be computationally feasible in real time. Finally, there is no systematic way to determine stability of the resulting closed loop system (short of exhaustive simulation).

Recent extensions to robust control offer the opportunity to use linear design tools in achieving specified levels of performance with time-varying systems. The remainder of the chapter will discuss one such approach. First, the manipulator *performance* objective will be examined using a simplified model, and found to be amenable to expression in a small-gain framework. This suggests the use of a small-gain robust

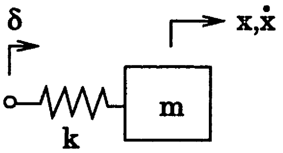

Figure 3-1: Flexible manipulator typical section.

control method to account for the *stability* objective. One such algorithm from the current literature is developed[28, 29]. The algorithm consists of a system of Linear Matrix Inequality (LMI) conditions, whose feasibility can be checked using standard Matlab©tools. If the conditions are feasible, a numerically reliable algorithm for determining a controller which produces that closed loop system exists. Some extensions to the basic algorithm are discussed. Finally, the algorithm will be verified on the sample problem. Characteristics of the solution method and of the resulting gain-scheduled controller for the manipulator will be discussed.

3.2 Performance Metric

The control problem is to accurately position the endpoint of the manipulator. Generally, the desire is to move the endpoint to the desired position as quickly as possible, with as little residual motion as possible. This is a transient time domain metric. An appropriate metric is the difference between the desired and the actual tip positions. A simple single degree of freedom model of a flexible manipulator can be used to illustrate the problem. Such simple models, sometimes referred to as typical section models[69] after those used in aerodynamics, are commonly used to give insight into the behavior of complex systems[70, 71, 72, 18].

The robot typical section consists of a spring-mass system (Figure 3-1). The spring represents the flexibility of the link, and the mass the combined mass of the link and the payload. The left end of the spring moves a commanded distance δ, representing the angle command fed to the manipulator joint servo. The EOM is

$$\ddot{x} = -\frac{k}{m}x + \frac{k}{m}\delta \tag{3.1}$$

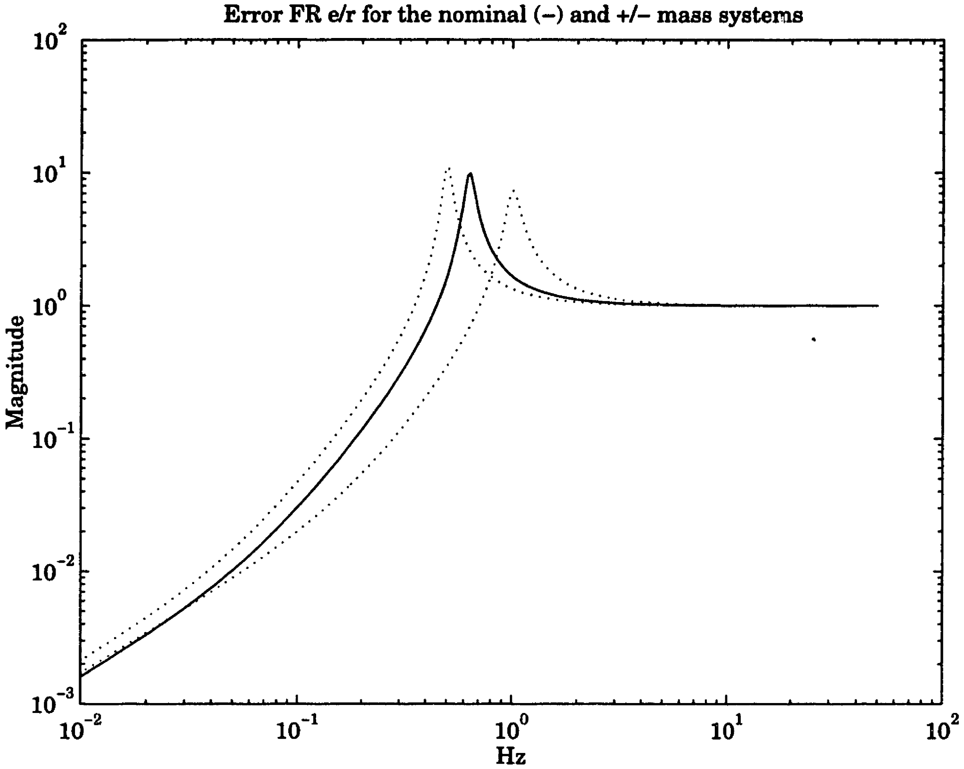

Figure 3-2: Tip position error vs frequency for the typical section: nominal (solid line), low mass (dotted line, higher frequency pole), high mass (dotted line, lower frequency pole).

The performance metric which is being proposed consists of the error e between the commanded and the actual position:

$$e = x - \delta \tag{3.2}$$

The frequency response function from the command to the error is shown in Figure 3-2. It is small at low frequency, because the mass follows the command. At the natural frequency the error is large since the mass is oscillating about the desired position. At high frequency, the inertia of the mass holds it in place, so the error transfer function goes to unity. The mass is essentially isolated from the command by the spring.

The transfer function from command to error is in fact the sensitivity[73]. Note that Bode's integral must hold:

$$\int_0^\infty \log(S(\jmath\omega))d\omega = const \tag{3.3}$$

That is, decreasing the error over some frequency range will increase it over other ranges ("push-pop"). In other words, performance cannot be arbitrarily improved. A frequency domain performance criteria allows the error to be reduced over some reasonable working bandwidth, perhaps set by the bandwidth of the human operator. At the same time the controller can be forced to roll off at high frequency, to avoid unstable interaction with unmodeled structural modes ("spillover"[69]). Note also the well-understood role of non-minimum phase zeros in limiting the achievable performance[74].

The robot control problem therefore resolves into minimization of the sensitivity. Performance can be improved by increasing the damping in the system, which decreases the large error at the natural frequency. The settling time for the system will likewise decrease. Once the mode reaches critical damping, time domain performance can only be improved by shifting the mode higher in frequency. This will cause rise (and settling) times to decrease.

The robot manipulator performance metric therefore is reasonably expressed as a frequency domain metric. Caution is required about which inputs and outputs are used to capture the motion of the tip. Often in the literature[74], the performance outputs are the positions of the end of each link, relative to the base of that link. This is convenient since the performance inputs and outputs are independent of configuration. However, a 3D manipulator can rotate in torsion at the elbow. Such a motion is not observable to an elbow position sensor, and is not controllable by the elbow. Hence the link end-point positions are not appropriate metrics for a 3D manipulator. The set of performance outputs must include the torsion at the end of each link.

The choice of performance metrics for the 3D manipulator will be explored further in Chapter 6. The remainder of this chapter will be devoted to a time-varying robust control approach. The algorithm will be applied to the typical section problem of this section, with the proposed sensitivity metric.

3.3 Mathematical Preliminaries

This section defines some terms and mathematical operations which will be needed to develop small-gain time-varying robust control design method. The terminology will be consistent with Apkarian and Gahinet[29].

The general form for a Linear Time-Varying (LTV) plant is

$$\dot{x} = A(t)x + B(t)u \tag{3.4}$$

$$y = C(t)x + D(t)u \tag{3.5}$$

In certain cases the dynamics depend linearly on a set of time-varying parameters. Such plants are termed Linear Parameter-Varying (LPV), and have the form

$$\dot{x} = A(\theta(t))x + B(\theta(t))u \tag{3.6}$$

$$y = C(\theta(t))x + D(\theta(t))u \tag{3.7}$$

where A, B, C, and D are linear fractional functions of θ. Such systems allow the use of small gain LTI techniques. Nothing of θ is known except its range of variations.

The Linear Fractional Transformation (LFT) is defined for appropriately dimensioned matrices M ana K. The lower LFT is

$$F_l(M, K) = M_{11} + M_{12}K(I - M_{22}K)^{-1}M_{21} \tag{3.8}$$

and the upper LFT is

$$F_u(M, K) = M_{22} + M_{21}K(I - M_{11}K)^{-1}M_{12} \tag{3.9}$$

Note that the LFT essentially denotes a feedback to M through K.

The $\mathcal{H}_\infty$ norm for a stable, real rational transfer function is defined as

$$||G(s)||_\infty = sup_w \bar{\sigma}(G(j\omega)) \tag{3.10}$$

where $\bar{\sigma}(M)$ denotes the largest singular value of the matrix M. The $\mathcal{H}_2$induced norm of a general operator T will also be denoted

$$||T||_\infty = sup_{u \in L_2} \frac{||Tu||}{||u||} \tag{3.11}$$

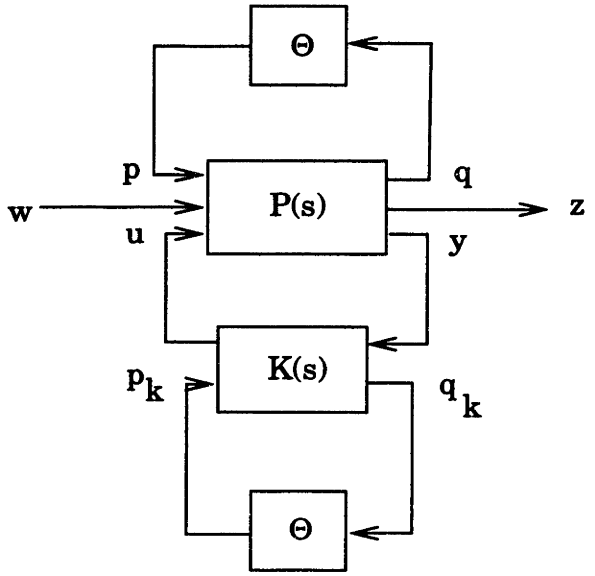

Figure 3-3: Gain scheduled controller setup, with disturbances w, performance outputs z, measurements y, and control inputs u. Parameter variations Θ are also measured by controller K.

For real symmetric matrices M, $M > 0$ stands for "positive definite" and denotes that all eigenvalues are positive. $M < 0$ is "negative definite" and indicates all eigenvalues are negative. For $M = M^T > 0$, $M^{\frac{1}{2}} = M^{\frac{1}{2}^T} > 0$ is the unique positive definite square root. The *Schur complement* of the matrix inequality

$$\begin{bmatrix} P & R \\ R^T & Q \end{bmatrix} > 0 \tag{3.12}$$

is

$$P > 0 \tag{3.13}$$

$$P - RQ^{-1}R^T > 0 \tag{3.14}$$

Finally $ker(P)$ denotes the null space of the linear operator associated with P.

With these concepts and definitions, the small-gain design framework can be developed.

3.4 LPV Gain Scheduled Control Setup

The gain-scheduled closed loop system which is desired is shown schematically in Figure 3-3. The linear plant model, $P(s)$, has disturbance inputs w, performance outputs z, control inputs u, and sensors y. The time variation of the plant arises from the feedback of the auxiliary variables q through a time-varying matrix operator Θ. The operator impinges on P through the inputs p.

The time-varying operator Θ is a function of a set of parameters $(\theta_1(t), \ldots, \theta_r(t))$. Θ has a *structure* associated with it, which describes how the time-varying parameters appear. Typically the structure will be block diagonal, with the parameters appearing on or near the diagonal of Θ. In the present work, Θ will be restricted to be diagonal:

$$\Theta = \begin{bmatrix} \theta_1 I_{r_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \theta_r I_{r_r} \end{bmatrix} \tag{3.15}$$

where I_{r_i} denotes an identity matrix of size r_i. In other words, each parameter θ_i is repeated r_i times. This restriction will simplify the representation of some of the matrix variables in the control design algorithm. It does not reduce the generality of the result as the plant model P can always be rearranged to produce a diagonal Θ block. The set of operators with structure 3.15 is denoted by

$$\Delta := \left\{ \begin{bmatrix} \theta_1 I_{r_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \theta_r I_{r_r} \end{bmatrix}, \theta_i \in R \right\} \tag{3.16}$$

Adopting the terminology of linear time invariant (LTI) robust control, Δ is referred to as the uncertainty structure, and Θ is the uncertainty block.

The time-varying plant model consists of the linear system coupled to the time-varying parameters, and is written

$$\begin{bmatrix} q \\ z \\ y \end{bmatrix} = \begin{bmatrix} P_{\theta\theta} & P_{\theta 1} & P_{\theta 2} \\ P_{1\theta} & P_{11} & P_{12} \\ P_{2\theta} & P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} p \\ w \\ u \end{bmatrix} \tag{3.17}$$

$$p = \Theta q \tag{3.18}$$

Note that this is an upper LFT, $F_u(P, \Theta)$. Equivalently it is a feedback of the outputs q to the inputs p, through a time varying gain.

The gain-scheduled controller which is sought, K, is also operated on by the uncertainty block Θ. The parameter variations operate on the controller signal q_k, and impinge on the controller through the input p_k. The time varying controller has the form

$$\begin{bmatrix} u \\ q_k \end{bmatrix} = \begin{bmatrix} K_{11} & K_{1\theta} \\ K_{\theta 1} & K_{\theta\theta} \end{bmatrix} \begin{bmatrix} y \\ p_k \end{bmatrix} \tag{3.19}$$

$$p_k = \Theta q_k \tag{3.20}$$

The signals to the gain-scheduled controller, p_k and q_k, will be referred to as the *parameter measurements.*

In the absence of the controller parameter measurements p_k and q_k, the setup is the same as for conventional LTI small gain control design (except for the time-varying nature of the uncertainty block). However, the presence of what is essentially a measurement of the disturbance, changes the problem. The gain-scheduled controller consists of an interconnection of K with Θ. Due to this interconnection the controller is time-varying (due to the time variation of Θ) and thus is not amenable to LTI analysis. It is necessary to restate the problem in a form which allows the use of LTI tools.

The modified design plant is generated based on an augmented uncertainty block. The system block diagram is re-drawn as in Figure 3-4. The original delta block is augmented with a copy which interconnects with K. Note that the same disturbance, control, and uncertainty signals (w, u, and p respectively) act on the plant P, and the same performance outputs, measurements, and uncertainty signals (z, y, q) are created. However, the parameter measurements p_k and q_k are now represented in the plant model, with a direct feedthrough path from the measurements to the controller inputs and outputs (the dashed lines leading to and from the lower block of the uncertainty block). The dotted line around P and the feedthrough path represents the LPV design model P_a. It has been augmented with inputs and outputs representing the uncertainty measurements. The input/output relationship for the augmented

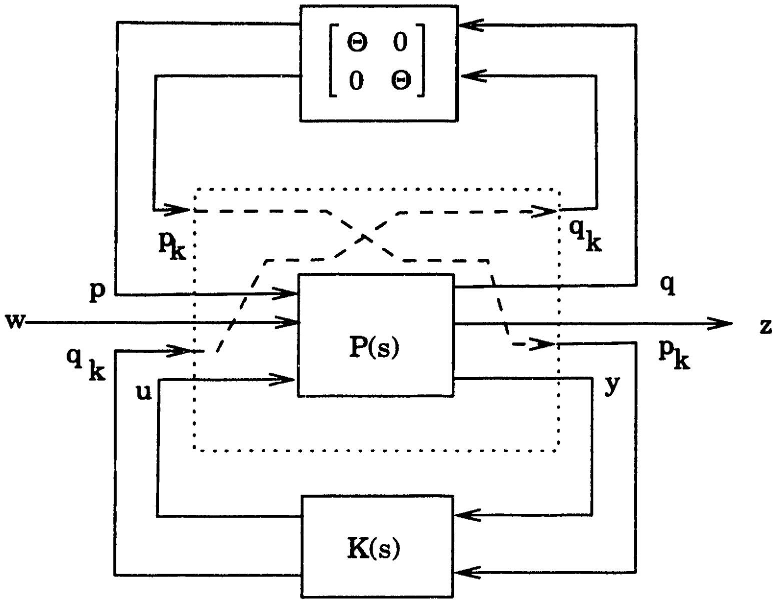

Figure 3-4: The re-drawn plant for the gain-scheduled controller design problem, showing the design plant (dotted box) augmented with the parameter measurements (dashed lines).

plant is

$$\begin{bmatrix} q \\ q_k \\ z \\ y \\ p_k \end{bmatrix} = \begin{bmatrix} 0 & 0 & I \\ 0 & P & 0 \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ p_k \\ w \\ u \\ q_k \end{bmatrix} \tag{3.21}$$

From Figure 3-4, the controller for the augmented plant is LTI, and can be treated as a standard small-gain design problem. Further, as will be shown, the block-repeated structure of the augmented uncertainty block alleviates certain convexity problems associated with the solution of the scaled $\mathcal{H}_\infty$ design problem. Specifically, for the gain-scheduled controller the design equations can be shown to be equivalent to a set of Linear Matrix Inequalities (LMIs) in a set of design variables. The implication is that the equations are convex, and a globally optimal solution can be found in a straightforward manner.

3.5 State Space Realization

The nominal design plant can be realized as the state space system

$$\begin{bmatrix} \dot{x} \\ q \\ z \\ y \end{bmatrix} = \begin{bmatrix} A & B_\theta & B_1 & B_2 \\ C_\theta & D_{\theta\theta} & D_{\theta 1} & D_{\theta 2} \\ C_1 & D_{1\theta} & D_{11} & D_{12} \\ C_2 & D_{2\theta} & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} x \\ p \\ w \\ u \end{bmatrix} \tag{3.22}$$

with state variables $x \in R^n$, $q \in R^r$, , $u \in R^{m_2}$, $y \in R^{p_2}$. Following Apkarian and Gahinet[29], the performance inputs w and outputs z are the same dimension ($z, w \in R^{p_1}$), p_1 being equal to the larger dimension of the two. This is not generally the case for the original plant model, but may be achieved by zero-padding the plant appropriately. As with the restriction to the diagonal Δ block, this step is taken to reduce the notational burden in carrying through the scaling matrices. The augmented plant realization is

$$\begin{bmatrix} \dot{x} \\ q_k \\ q \\ z \\ y \\ p_k \end{bmatrix} = \begin{bmatrix} A & 0 & B_\theta & B_1 & B_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & I \\ C_\theta & 0 & D_{\theta\theta} & D_{\theta 1} & D_{\theta 2} & 0 \\ C_1 & 0 & D_{1\theta} & D_{11} & D_{12} & 0 \\ C_2 & 0 & D_{2\theta} & D_{21} & D_{22} & 0 \\ 0 & I & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ p_k \\ p \\ w \\ u \\ q_k \end{bmatrix} \tag{3.23}$$

with the parameter measurements $p_k, q_k \in R^{r_c}$. The parameter measurement dimension r_c is unknown, and will be determined during the problem solution. The controller is realized as

$$\begin{bmatrix} \dot{x}_K \\ u \\ q_K \end{bmatrix} = \begin{bmatrix} A_K & B_{K1} & B_{K\theta} \\ C_{K1} & D_{K11} & D_{K1\theta} \\ C_{K\theta} & D_{K\theta 1} & D_{K\theta\theta} \end{bmatrix} \begin{bmatrix} x_K \\ y \\ p_K \end{bmatrix} \tag{3.24}$$

with $x \in R^k$, where again k is unknown and will be determined in the course of the problem solution.

3.6 Scaled Bounded Real Lemma

The Scaled Bounded Real Lemma is used in a mathematical framework termed *small gain theory*[75]. The framework hinges on the ability to bound the input to output gain of a closed loop system. The bound is expressed as an $\mathcal{H}_\infty$ norm. Given

$$V(x) > 0 \tag{3.25}$$

and given a scalar γ for which

$$\dot{V} + z^T z - \gamma^2 w^T w < 0 \tag{3.26}$$

then

$$\left\| \frac{z}{w} \right\|_\infty \leq \gamma \tag{3.27}$$

Proof: Integrate Equation 3.26 from 0 to ∞, then

$$\|z\|_2^2 - \gamma^2 \|w\|_2^2 \leq 0 \tag{3.28}$$

That is, γ is an upper bound on the worst-case norm of the system.

Small gain theory allows an $\mathcal{H}_\infty$ bound to be used to analyze the stability as well as the performance of a system[76]. Small gain theory states that if the $\mathcal{H}_\infty$ gain through the uncertainty block is bounded, the closed loop system will always be stable in the presence of appropriately bounded uncertainties[75]. Through appropriate manipulations, the same tool can be used to synthesize a controller which meets the performance criteria while stabilizing the plant under all allowable uncertainties. Since the bound on the size of the uncertainties is set by the smallest value of the $\mathcal{H}_\infty$ norm γ, matrix-valued scalings L on z and w can be used to reduce the size of the minimum γ which can be found[77]. The criteria for these scalings is that they lie in the set L_Δ, defined as

$$L_\Delta = \{L > 0 : L\Theta = \Theta L, \vee\Theta \in \Delta\} \tag{3.29}$$

In other words, the quantities $z^T Lz$ and $w^T Lw$ are always positive, and the scalings commute with the Δ block. These concepts are captured in the *Scaled Bounded Real* Lemma.

Lemma 1 *Bounded Real Lemma:*

Given a transfer function with state space realization as in Equation 3.22, a parameter structure Δ, and an associated scaling set L_Δ, the following statements are equivalent:

1. A is stable, and there exists $L \in L_\Delta$ such that

$$\left\| L^{1/2} \left(C(sI - A)^{-1}B + D \right) L^{-1/2} \right\|_\infty < \gamma \tag{3.30}$$

2. there exist positive definite solutions P and L to the matrix inequality

$$\begin{bmatrix} A^TP + PA & PB & C^T \\ B^TP & -\gamma L & D^T \\ C & D & -\gamma L^{-1} \end{bmatrix} < 0 \tag{3.31}$$

Proof: Assume A is stable. Given the Lyapunov function

$$V(x) = x^TPx > 0 \tag{3.32}$$

By Equation 3.26, γ represents a bound on the norm if

$$\dot{V} + z^Tz - \gamma^2 w^Tw < 0 \tag{3.33}$$

Since

$$\dot{x} = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \tag{3.34}$$

$$z = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \tag{3.35}$$

the inequality can be written

$$\begin{bmatrix} x \\ w \end{bmatrix}^T \left(\begin{bmatrix} A^TP + PA & PBL^{-\frac{1}{2}} \\ L^{-\frac{1}{2}}B^TP & 0 \end{bmatrix} + \begin{bmatrix} C^TLC & C^TLDL^{-\frac{1}{2}} \\ L^{-\frac{1}{2}}D^TLC & L^{-\frac{1}{2}}D^TLDL^{-\frac{1}{2}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \gamma^2 I \end{bmatrix} \right) \begin{bmatrix} x \\ w \end{bmatrix} < 0 \tag{3.36}$$

Pre-and post-multiplying by a positive definite matrix does not change the validity of a matrix inequality. Pre- and post-multiplying Equation 3.36 by

$$\begin{bmatrix} I & 0 \\ 0 & L^{\frac{1}{2}} \end{bmatrix} \tag{3.37}$$

produces the inequality

$$\begin{bmatrix} A^T P + PA + C^T LC & PB + C^T LD \\ (PB + C^T LD)^T & D^T LD - \gamma^2 I \end{bmatrix} < 0 \tag{3.38}$$

Using Schur complements this can be rewritten

$$\begin{bmatrix} A^T P & PB & C^T \\ (PB)^T & -\gamma^2 L & D^T \\ C & D & -L^{-1} \end{bmatrix} < 0 \tag{3.39}$$

Redefining the scaling matrix L to be $\frac{1}{\gamma}L$, the result holds:

$$\begin{bmatrix} A^T P & PB & C^T \\ (PB)^T & -\gamma L & D^T \\ C & D & -\gamma L^{-1} \end{bmatrix} < 0 \tag{3.40}$$

The Scaled Bounded Real Lemma can be used to analyze the block-repeated uncertainty structure in Figure 3-4, and equivalently Equation 3.23. The result will hinge on identifying the uncertainty scalings which commute with the block-repeated uncertainty structure. This set can be defined as

$$L_{\Delta\oplus\Delta} := \left\{ \begin{bmatrix} L_1 & L_2 \\ L_2^T & L_3 \end{bmatrix} > 0 : L_1, L_3 \in L_\Delta \text{ and } L_2\Theta = \Theta L_2, \forall\Theta \in \Delta \right\} \tag{3.41}$$

where the aggregate scaling matrix $L_{\Delta\oplus\Delta}$ must be positive definite. The off-diagonal block L_2 need not itself be positive definite, thus may not be in the set L_Δ.

3.7 Scaled $\mathcal{H}_\infty$ Synthesis

The treatment by Apkarian and Gahinet[29] is straightforward. The following derivation will be presented consistent with the notation in that work, with an error in the sufficiency conditions rectified. Given a plant with disturbances w, performance outputs z, controller inputs u and sensor outputs y with state space realization:

$$\dot{x} = Ax + B_1 w + B_2 u \tag{3.42}$$

$$z = C_1 x + D_{11} w + D_{12} u \tag{3.43}$$

$$y = C_2 x + D_{21} w + D_{22} u \tag{3.44}$$

The Scaled Bounded Real Lemma can be used to formulate a controller which achieves a given level of performance (if one exists). Start with a realization for the (currently unknown) closed loop system

$$\begin{bmatrix} A_{cl} & B_{cl} \\ C_{cl} & D_{cl} \end{bmatrix} = \begin{bmatrix} A_0 + \mathcal{B}\Omega\mathcal{C} & B_0 + \mathcal{B}\Omega\mathcal{D}_{21} \\ C_0 + \mathcal{D}_{12}\Omega\mathcal{C} & D_{11} + \mathcal{D}_{12}\Omega\mathcal{D}_{21} \end{bmatrix} \tag{3.45}$$

where the plant is composed of the open loop plant, augmented by appropriately dimensioned zero and identity matrices:

$$A_0 = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} \tag{3.46}$$

$$B_0 = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tag{3.47}$$

$$C_0 = \begin{bmatrix} C_1 & 0 \end{bmatrix} \tag{3.48}$$

$$\mathcal{B} = \begin{bmatrix} 0 & B_2 \\ I_k & 0 \end{bmatrix} \tag{3.49}$$

$$\mathcal{C} = \begin{bmatrix} 0 & I_k \\ C_2 & 0 \end{bmatrix} \tag{3.50}$$

$$\mathcal{D}_{12} = \begin{bmatrix} 0 & D_{12} \end{bmatrix} \tag{3.51}$$

$$\mathcal{D}_{21} = \begin{bmatrix} 0 \\ D_{21} \end{bmatrix} \tag{3.52}$$

The subscript k denotes the compensator order, which is unknown, as is the compensator realization

$$\Omega = \begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix} \tag{3.53}$$

These quantities must be determined by the synthesis algorithm.

The problem is to find a compensator for which

$$\left\| L^{-1/2} \left(C(sI - A)^{-1} B + D \right) L^{-1/2} \right\|_{\infty} < \gamma \tag{3.54}$$

in other words, for which the norm from disturbance to performance is below some specified value. The Scaled Bounded Real Lemma can be applied to the plant data, such that if there is a symmetric positive definite X_{cl} for which

$$\begin{bmatrix} A_{cl}^T X_{cl} + X_{cl} A_{cl} & X_{cl} B_{cl} & C_{cl}^T \\ B_{cl}^T X_{cl} & -\gamma L & D_{cl}^T \\ C_{cl} & D_{cl} & -\gamma L^{-1} \end{bmatrix} < 0 \tag{3.55}$$

there is a closed loop system for which Equation 3.54 is true. This can be re-written

$$\begin{bmatrix} A_0^T X_{cl} + X_{cl} A_0 & X_{cl} B_0 & C_0^T \\ B_0^T X_{cl} & -\gamma L & D_{11}^T \\ C_0 & D_{11} & -\gamma L^{-1} \end{bmatrix} + \begin{bmatrix} X_{cl} \mathcal{B} \\ 0 \\ \mathcal{D}_{12} \end{bmatrix} \Omega \begin{bmatrix} \mathcal{C} & \mathcal{D}_{21} & 0 \end{bmatrix} + \begin{bmatrix} \mathcal{C}^T \\ \mathcal{D}_{21}^T \\ 0 \end{bmatrix} \Omega^T \begin{bmatrix} \mathcal{B}^T X_{cl} & 0 & \mathcal{D}_{12}^T \end{bmatrix} < 0 \tag{3.56}$$

Defining the matrices Ψ, P, and Q, Equation 3.56 can be written

$$\Psi + P_x^T \Omega Q + Q^T \Omega^T P_x < 0 \tag{3.57}$$

This is a matrix inequality in the unknowns X_{cl}, L, and Ω. It is nonlinear in L, X_{cl}, and Ω. The dependence on Ω can be removed using the *Elimination Lemma*, which states that Equation 3.57 has a solution if and only if

$$W_{P_x}^T \Psi W_{P_x} < 0 \tag{3.58}$$

$$W_Q^T \Psi W_Q < 0 \tag{3.59}$$

with W_{P_x} and W_Q the null spaces of P_x and Q, respectively:

$$P_x W_{P_x} = 0 \tag{3.60}$$

$$Q W_Q = 0 \tag{3.61}$$

Note that the matrix P_x is

$$P_x = \begin{bmatrix} \mathcal{B}^T & 0 & \mathcal{D}_{12}^T \end{bmatrix} \begin{bmatrix} X_{cl} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \tag{3.62}$$

so that Equation 3.60 can be written

$$P_x W_{P_x} = P \begin{bmatrix} X_{cl} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} W_{P_x} = P W_P = 0 \tag{3.63}$$

Equation 3.58 can be written

$$W_P^T \begin{bmatrix} X_{cl}^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \Psi \begin{bmatrix} X_{cl}^{-1} & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} < 0 \tag{3.64}$$

and with the definition

$$\Phi = \begin{bmatrix} A_0 X_{cl}^{-1} + X_{cl}^{-1} A_0^T & X_{cl} & X_{cl}^{-1} C_0^T \\ B_0^T & -\gamma L & D_{11}^T \\ C_0 X_{cl}^{-1} & D_{11} & -\gamma L^{-1} \end{bmatrix} \tag{3.65}$$

the conditions Equations 3.58-3.59 can be rewritten

$$W_P^T \Phi W_P < 0 \tag{3.66}$$

$$W_Q^T \Psi W_Q < 0 \tag{3.67}$$

where the outer factors are now functions of known plant data.

The closed loop Lyapunov matrix (and its inverse) can be partitioned conformably with the closed loop plant

$$X_{cl} = \begin{bmatrix} S & N \\ N^T & E \end{bmatrix}, \; X_{cl}^{-1} + \begin{bmatrix} R & M \\ M^T & F \end{bmatrix} \tag{3.68}$$

Equations 3.66 and 3.59 become

$$W_P^T \begin{bmatrix} AR+RA^T & AM & B_1 & RC_1^T \\ M^TA^T & 0 & 0 & M^TC_1^T \\ B_1^T & 0 & -\gamma L & D_{11}^T \\ C_1R & C_1M & D_{11} & -\gamma L^{-1} \end{bmatrix} W_P < 0 \tag{3.69}$$

$$W_Q^T \begin{bmatrix} A^TS+SA & A^TN & SB_1 & C_1^T \\ N^TA & 0 & N^TB_1 & 0 \\ B_1^TS & B_1^TN & -\gamma L & D_{11}^T \\ C_1 & 0 & D_{11} & -\gamma L^{-1} \end{bmatrix} W_Q < 0 \tag{3.70}$$

Turning to the null spaces W_P and W_Q, the plant data imbues them with a particular structure:

$$PW_p = \begin{bmatrix} 0 & I & 0 & 0 \\ B_2^T & 0 & 0 & D_{12}^T \end{bmatrix} \begin{bmatrix} 0 & P_1 \\ 0 & 0 \\ I & 0 \\ 0 & P_2 \end{bmatrix} = 0 \tag{3.71}$$

$$QW_Q = \begin{bmatrix} 0 & I & 0 & 0 \\ C_2 & 0 & D_{21} & 0 \end{bmatrix} \begin{bmatrix} 0 & Q_1 \\ 0 & 0 \\ 0 & I \\ Q_3 & 0 \end{bmatrix} = 0 \tag{3.72}$$

The zero rows in the null spaces appear conformably with the compensator states. Essentially the zeros arise because the compensator states are known. Applying Equations 3.71 and 3.72 to Equations 3.69 and 3.70 amounts to cancelling the second row and column. Re-ordering rows and columns, and defining

$$J = L^{-1} \tag{3.73}$$

these become

$$N_R^T \begin{bmatrix} AR+RA^T & RC_1^T & B_1 \\ C_1R & -\gamma J & D_{11} \\ B_1^T & D_{11}^T & -\gamma L \end{bmatrix} N_R < 0 \tag{3.74}$$

$$N_S^T \begin{bmatrix} A^TS+SA & SB_1 & C_1^T \\ B_1^TS & -\gamma L & D_{11}^T \\ C_1 & D_{11} & -\gamma J \end{bmatrix} N_S < 0 \tag{3.75}$$

$$\begin{bmatrix} B_2^T & D_{21}^T & 0 \end{bmatrix} N_R = 0 \tag{3.76}$$

$$\begin{bmatrix} C_2 & D_{12} & 0 \end{bmatrix} N_S = 0 \tag{3.77}$$

The matrices R and S must be related such that there is a matrix X_{cl} for which Equation 3.68 holds. This can be insured with the matrix constraint

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} > 0 \tag{3.78}$$

These conditions are identical to Equation 3.57. However, while the latter is not linear in the matrix unknowns X_{cl} and L, the above are linear in R and S. However, the inequality condition in Equation 3.73 is non-convex since it is nonlinear in the matrix variables. However, as the next section will show, if the parameter variations are measurable, the problem reduces to a convex one.

3.8 Gain Scheduled $\mathcal{H}_\infty$ Synthesis

The augmented plant defined in Section 3.5 is considered. Perform the following substitutions, from the augmented plant data,

$$C_1 \rightarrow \begin{bmatrix} 0 \\ C_\theta \\ C_1 \end{bmatrix}, \quad B_1 \rightarrow \begin{bmatrix} 0 & B_\theta & B_1 \end{bmatrix}$$

$$C_2 \rightarrow \begin{bmatrix} C_2 \\ 0 \end{bmatrix}, \quad B_2 \rightarrow \begin{bmatrix} B_2 & 0 \end{bmatrix}$$

$$D_{11} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & D_{\theta\theta} & D_{\theta 1} \\ 0 & D_{1\theta} & D_{11} \end{bmatrix}, \quad D_{12} \rightarrow \begin{bmatrix} 0 & I \\ D_{\theta 2} & 0 \\ D_{12} & 0 \end{bmatrix}$$

$$D_{21} \rightarrow \begin{bmatrix} 0 & D_{2\theta} & D_{21} \\ I & 0 & 0 \end{bmatrix}, \quad D_{22} \rightarrow \begin{bmatrix} D_{22} & 0 \\ 0 & 0 \end{bmatrix}$$

$$L \rightarrow \begin{bmatrix} L_1 & L_2 & 0 \\ L_2^T & L_3 & 0 \\ 0 & 0 & I \end{bmatrix}, \quad J \rightarrow \begin{bmatrix} J_1 & J_2 & 0 \\ J_2^T & J_3 & 0 \\ 0 & 0 & I \end{bmatrix}$$

Again partition the null space

$$\begin{bmatrix} B_2^T & 0 & D_{\theta 2}^T & D_{12}^T \\ 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ 0 \\ P_{21} \\ P_{22} \end{bmatrix} = 0 \tag{3.79}$$

$$\begin{bmatrix} C_2 & 0 & D_{2\theta} & D_{21} \\ 0 & I & 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ 0 \\ Q_{21} \\ Q_{22} \end{bmatrix} = 0 \tag{3.80}$$

where

$$\begin{bmatrix} B_2^T & D_{\theta 2}^T & D_{12}^T \end{bmatrix} \begin{bmatrix} P_1 \\ P_{21} \\ P_{22} \end{bmatrix} = 0 \tag{3.81}$$

$$\begin{bmatrix} C_2 & D_{2\theta} & D_{21} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_{21} \\ Q_{22} \end{bmatrix} = 0 \tag{3.82}$$

$$\tag{3.83}$$

Turning to Equations 3.74-3.75, Schur Complements are used to reduce them to

$$N_R^T \begin{bmatrix} AR + RA^T + \frac{1}{\gamma}B_1 J B_1 & RC_1^T + \frac{1}{\gamma}B_1 J D_{11}^T \\ (RC_1^T + \frac{1}{\gamma}B_1 J D_{11}^T)^T & -\gamma J + \frac{1}{\gamma}D_{11} J D_{11}^T \end{bmatrix} N_R < 0 \tag{3.84}$$

$$N_S^T \begin{bmatrix} A^T S + SA + \frac{1}{\gamma}C_1^T L C_1 & SB_1 + \frac{1}{\gamma}C_1^T L D_{11} \\ (SB_1 + \frac{1}{\gamma}C_1^T L D_{11})^T & -\gamma L + \frac{1}{\gamma}D_{11}^T L D_{11} \end{bmatrix} N_S < 0 \tag{3.85}$$

Performing the substitutions 3.79 and deleting the row and column which are multiplied by zero produces the following LMIs

$$N_R^T \begin{bmatrix} \begin{array}{c} AR + RA^T + \\ \frac{1}{\gamma}(B_\theta J_3 B_\theta + B_1 B_1^T) \end{array} & & sym. \\ \begin{array}{c} C_\theta R + \\ \frac{1}{\gamma}(D_{\theta\theta} J_3 B_\theta^T + D_{\theta 1} B_1^T) \end{array} & \begin{array}{c} -\gamma J_3 + \\ \frac{1}{\gamma}(D_{\theta\theta} J_3 D_{\theta\theta}^T + D_{\theta 1} D_{\theta 1}^T) \end{array} & \\ \begin{array}{c} C_1 R + \\ \frac{1}{\gamma}(D_{1\theta} J_3 B_\theta^T + D_{11} B_1^T) \end{array} & \frac{1}{\gamma}(D_{1\theta} J_3 D_{\theta\theta}^T + D_{11} D_{\theta 1}^T) & \begin{array}{c} -\gamma I + \\ \frac{1}{\gamma}(D_{1\theta} J_3 D_{1\theta}^T + D_{11} D_{11}^T) \end{array} \end{bmatrix} N_R < 0 \tag{3.86}$$

$$N_S^T \begin{bmatrix} \begin{array}{c} A^T S + SA + \\ \frac{1}{\gamma}(C_\theta^T L_3 C_\theta + C_1^T C_1) \end{array} & & sym. \\ \begin{array}{c} B_\theta^T S + \\ \frac{1}{\gamma}(D_{\theta\theta}^T L_3 C_\theta + D_{1\theta}^T C_1) \end{array} & \begin{array}{c} -\gamma L_3 + \\ \frac{1}{\gamma}(D_{\theta\theta}^T L_3 D_{\theta\theta} + D_{1\theta}^T D_{1\theta}) \end{array} & \\ \begin{array}{c} C_1^T S + \\ \frac{1}{\gamma}(D_{\theta 1}^T L_3 C_\theta + D_{11}^T C_1) \end{array} & \frac{1}{\gamma}(D_{\theta 1}^T L_3 D_{\theta\theta} + D_{11}^T D_{1\theta}) & \begin{array}{c} -\gamma I + \\ \frac{1}{\gamma}(D_{\theta 1}^T L_3 D_{\theta 1} + D_{11}^T D_{11}) \end{array} \end{bmatrix} N_S < 0 \tag{3.87}$$

$$\begin{bmatrix} R & I \\ I & S \end{bmatrix} \geq 0 \tag{3.88}$$

where the matrices are symmetric. The condition

$$\begin{bmatrix} L_3 & I \\ I & J_3 \end{bmatrix} \geq 0 \tag{3.89}$$

ensures that a matrix L can be constructed such that Equation 3.73 holds[29].

The inequalities in Equations 3.86-3.89 are sufficient conditions for the closed loop inequality, Equation 3.57, to be satisfied. The matrix variables are R, S, L_3 and J_3, each of which appears linearly in the inequalities. The feasibility of these conditions can be readily determined using available Matlab© solvers.

Note that the $\mathcal{H}_\infty$ bound γ no longer appears linearly in the inequalities, in contrast to the conditions given in Reference [29]. One consequence is that the $\mathcal{H}_\infty$ performance cannot be explicitly minimized.

3.9 Computation of the Controller

The sufficiency conditions return a quadruple (R, S, L_3, J_3) for which Equation 3.57 holds. The final problem is to determine a controller realization. The algorithm given in Reference [29] is straightforward and will be reproduced.

First compute the Lyapunov matrix X_{cl}. Begin by computing via Singular Value Decomposition two matrices M, N for which

$$MN^T = I - RS \tag{3.90}$$

then compute X_{cl} as the unique solution of

$$X_{cl}\begin{bmatrix} I & R \\ 0 & M^T \end{bmatrix} = \begin{bmatrix} S & I \\ N^T & 0 \end{bmatrix} \tag{3.91}$$

Next compute L and J,using the same SVD approach, such that

$$L = \begin{bmatrix} L_1 & L_2 \\ L_2^T & L_3 \end{bmatrix} > 0, \; L^{-1} = \begin{bmatrix} * & * \\ & J_3 \end{bmatrix} \tag{3.92}$$

where the starred elements are matrix-valued variables such that the inverse relationship holds. Note, however, that the SVD must be performed on each block of L_3 and J_3, partitioned conformably with the Δ block.

Next form the matrices

$$\mathcal{L} = \begin{bmatrix} L & 0 \\ 0 & I \end{bmatrix}, \; \mathcal{J} = \mathcal{L}^{-1} \tag{3.93}$$

and solve the LMI Equation 3.57 in the matrix variable Ω, with X_{cl} and $\mathcal{L}$ known. The inequality condition is reproduced below.

$$\Psi + \begin{bmatrix} X_{cl} & 0 \\ 0 & I \end{bmatrix} P^T\Omega Q + Q^T\Omega^T P \begin{bmatrix} X_{cl} & 0 \\ 0 & I \end{bmatrix} \tag{3.94}$$

with

$$\Psi = \begin{bmatrix} A_0^T X_{cl} + X_{cl}A_0 & X_{cl}B_0 & C_0^T \\ B_0^T X_{cl} & -\gamma\mathcal{L} & D_{11}^T \\ C_0 & D_{11} & -\gamma\mathcal{J} \end{bmatrix} \tag{3.95}$$

$$P = \begin{bmatrix} \mathcal{B}^T & 0 & \mathcal{D}_{21}^T \end{bmatrix} \tag{3.96}$$

$$Q = \begin{bmatrix} 0 & \mathcal{C} & \mathcal{D}_{12} \end{bmatrix} \tag{3.97}$$

$$\Omega = \begin{bmatrix} A_k & B_{k_u} & B_{k_\theta} \\ C_{k_y} & D_{k_{yu}} & D_{k_{y\theta}} \\ C_{k_\theta} & D_{k_{\theta u}} & D_{k_{\theta\theta}} \end{bmatrix} \tag{3.98}$$

Note that, as the condition is an inequality, solutions can range within an entire half-space. If a particular solution is desired, the least-norm solution can be found[78].

3.10 Special cases

The LMI conditions Equations 3.86-3.89 are existence conditions, with an entire set of possible solutions. Some particular solutions may in some circumstances be desired. Three particular types are considered: $\mathcal{H}_\infty$ optimal control, and constant gain feedback, and LTI controllers.

The $\mathcal{H}_\infty$ optimal controller may be desired, for example in order to achieve the minimum state cost for a particular control cost. As noted above, any parameter which appears linearly in the feasibility conditions can be considered a matrix variable; that is, can be allowed to vary. The inverse of the $\mathcal{H}_\infty$ bound γ appears along with γ, hence it is not a linear parameter and a direct $\mathcal{H}_\infty$ optimization cannot be performed.

Reduced order controllers are often of interest. Apkarian and Gahinet identify the condition

$$\text{rank}\{I - RS\} \leq k \tag{3.99}$$

as a condition for the existence of a controller of order $k \leq n$ (n is the system order)[77]. This result follows because the reconstructed closed loop Lyapunov function, Equation 3.91, will be of order $k + n$. They note that the condition represented by Equation 3.99 is nonconvex. Nondynamic controllers correspond to solutions R and S for which

$$R = S^{-1} \tag{3.100}$$

In this case $X_{cl} = S$ and the compensator order $k = 0$. This can be achieved with the linear cost constraint

$$J_{CG} = \operatorname{tr}\left\{\begin{bmatrix} R & I \\ I & S \end{bmatrix}\right\} = \operatorname{tr}\{R\} + \operatorname{tr}\{S\} \tag{3.101}$$

which makes all the eigenvalues go to 1, if such a solution lies inside the feasible solution space. Otherwise controller order is minimized. Since this cost is linear in the matrix variables, it is convex. It does not allow the controller order to be directly specified. However, in practice it is noted that control and performance weights can be traded against system order (the tighter the performance bounds, the higher the system order).

Similarly, the cost

$$J_{LTI} = \operatorname{tr}\{J_3\} + \operatorname{tr}\{L_3\} \tag{3.102}$$

will move toward feasible solutions for which

$$J_3 = L_3^{-1} \tag{3.103}$$

and $r_c = 0$, corresponding to a controller which does not measure the parameter variations. This is beneficial because the controller calculation involves a matrix inversion, which is computationally costly.

3.11 Control Design Example

The gain-scheduled design approach is used to control the sample problem discussed in Section 3.2. The stiffness parameter K is taken to be 1. The mass parameter m is variable, representing configuration and payload changes. The mass ranges between 1 and 4. The system natural frequency thus varies by a factor of two, which is representative of the SRMS fully extended compared to fully retracted. Modal damping of 5% is added to prevent imaginary axis poles. The performance outputs are chosen as the position error of the mass, and the control effort: $z = \begin{bmatrix} e & u \end{bmatrix}^T$. The disturbance inputs are the reference command and sensor noise: $w = \begin{bmatrix} r & v \end{bmatrix}^T$. The control input is the command δ. The sensors are the position of the mass, and the reference command: $y = \begin{bmatrix} x & r \end{bmatrix}^T$. The reference command is added to the sensor suite since this allows dramatically higher performance. The state space realization for the design plant model is therefore:

$$A = \begin{bmatrix} 0 & 1 \\ -k/m & -2.05\sqrt{k/m_0} \end{bmatrix} \tag{3.104}$$

$$B_\theta = \sqrt{\delta m}\begin{bmatrix} 0 \\ 1/m_0 \end{bmatrix} \tag{3.105}$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ k/m_0 & 0 \end{bmatrix} \tag{3.106}$$

$$B_2 = \begin{bmatrix} 0 \\ k/m_0 \end{bmatrix} \tag{3.107}$$

$$C_\theta = \sqrt{\delta m}\begin{bmatrix} -k/m_0 & -2.05\sqrt{k/m_0} \end{bmatrix} \tag{3.108}$$

$$D_{\theta\theta} = \delta m/m_0 \tag{3.109}$$

$$D_{\theta 1} = \sqrt{\delta m}\begin{bmatrix} k/m_0 & 0 \end{bmatrix} \tag{3.110}$$

$$D_{\theta 2} = \sqrt{\delta m}\begin{bmatrix} k/m_0 \end{bmatrix} \tag{3.111}$$

$$C_1 = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.112}$$

$$D_{1\theta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.113}$$

$$D_{11} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.114}$$

$$D_{12} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.115}$$

$$C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.116}$$

$$D_{21} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.117}$$

$$D_{22} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.118}$$

$$D_{2\theta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.119}$$

$$\tag{3.120}$$

with the nominal mass $m_0 = 2$ and the maximum variation $\delta m = 1.5$. The uncertainty input/output channels are scaled by the mass variation such that the delta block is bounded by one: $|\theta_1| \leq 1$. Note that in this context, the uncertainty inputs p and q can be given another interpretation: they are "pointing" matrices which pick out certain combinations of the state matrix.

Figure 3-5 plots the error transfer function for the sample problem, along with the frequency dependent performance weight. The solid line represents the nominal sensitivity, while the dotted line with the higher resonant frequency represents the lowest mass case, and the dotted line with lower frequency represents the highest mass case. The performance weight represents the maximum sensitivity allowable in the closed loop system, as a function of frequency.

The desired closed loop behavior is influenced by the choice of performance frequency weightings, such as the performance weight shown above. The frequency weighting may also be a function of the mass. For example, it may be unreasonable to require the same tip position settling time for a fully loaded arm as for the unloaded arm. For space-based manipulators the payload mass to arm mass ratio may approach 1000:1 for some prefabricated elements of the proposed space station[1]. However, for a given payload it is reasonable to require the same settling time throughout the workspace. The advantage is that the response of the controlled arm will be the same at all points in the workspace, reducing operator workload. That is, the weighting function should be a function of the payload mass, but independent of the configuration.

The desired performance is represented by a weight on the sensitivity (the dashed line in Figure 3-5, which shows the inverse of the sensitivity weighting). The inverse

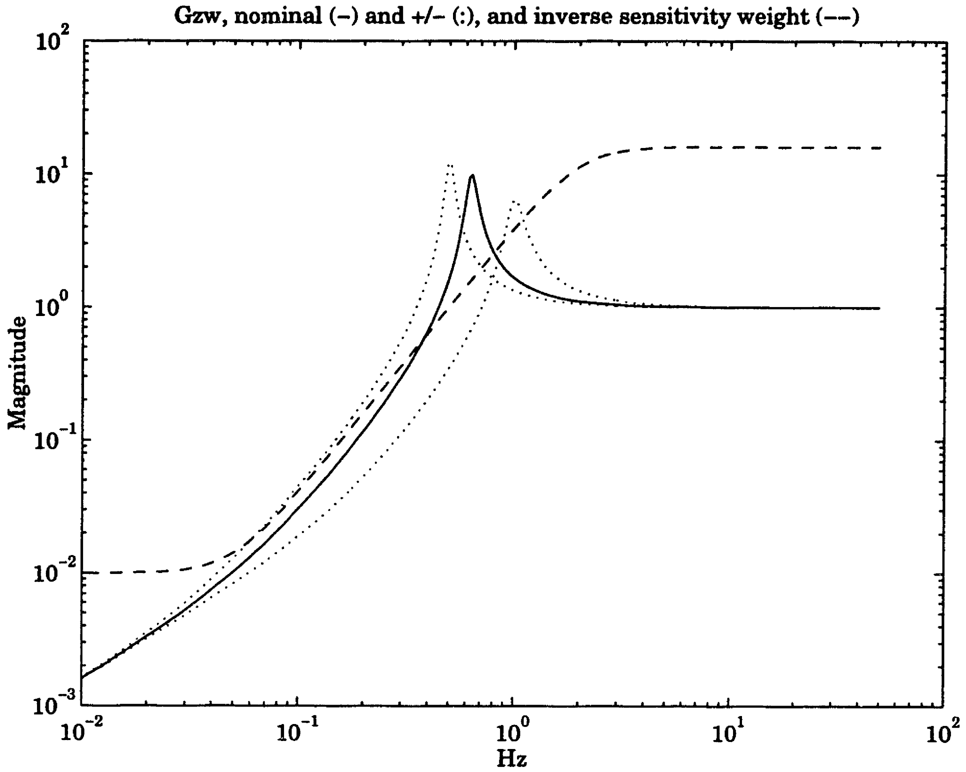

Figure 3-5: Reference input to sensitivity performance output, G_{zw} and weight.

weighting is low at low frequencies, to reduce the error where the majority of operator commands will occur. The weighting is such that the open-loop natural frequency will have to be increased to meet the performance objective. A two-pole rolloff is used to achieve a low bandwidth crossover. Crossover is at 0.5 Hz. The closed loop system should have a first mode at approximately 0.5 Hz. The weighting is allowed to increase above one, since the integral of the sensitivity must remain constant ("push-pop"). However, the smooth curve will ensure that the first mode is not highly resonant.

The bandwidth of the closed loop system can be influenced with a weight on the control. The chosen weight is frequency dependent, increasing at high frequency to penalize high frequency controller dynamics and force the controller to roll off. The two-pole rolloff starts at the desired closed-loop mode at 0.5 Hz. The control input to sensor output transfer function, $G_{yu}(s)$, and the control weight as a function of frequency are shown in Figure 3-6.

The step response of the open loop system is shown in Figure 3-7. The nominal

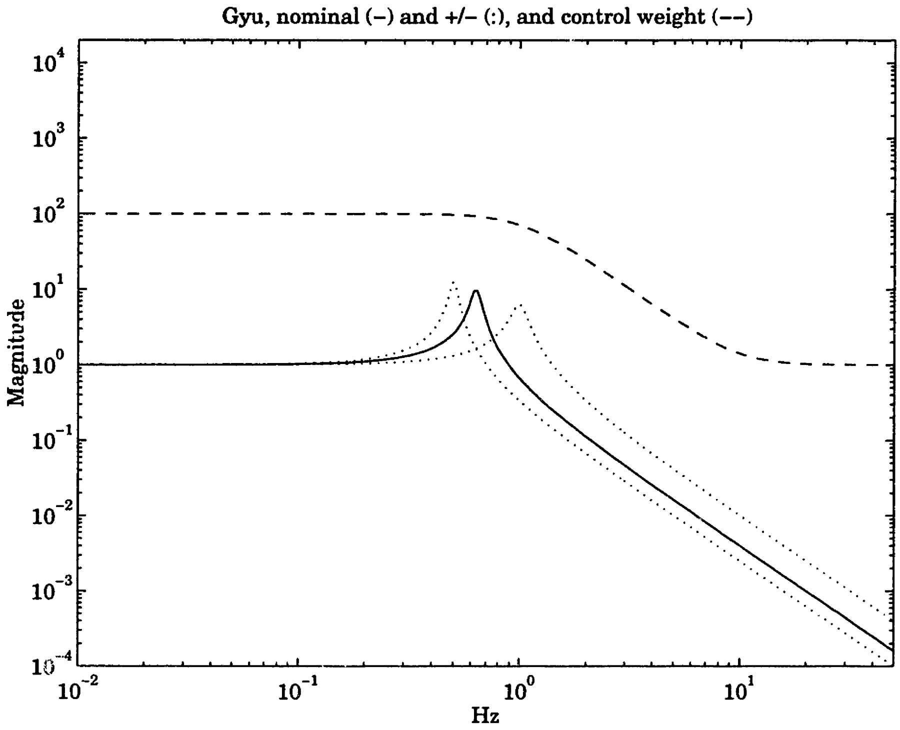

Figure 3-6: Control input to sensor output, G_{uy}, and control weight.

system is shown as a solid line, while the low-mass case is the faster dotted line and the high-mass case is the slower dotted line. Note the highly resonant character of the response, with a long settling time. Use of a manipulator with such a response would adversely impact the productivity of a human crew, since each commanded motion would be followed by a long period of waiting to allow the motion to settle out. Note also that often, crew learn to accommodate such highly resonant responses[49]. For example, they learn to generate a feedforward command which cancels the vibration in the arm while positioning the end-effector as desired. The variable period of vibration in the response as shown would make this more difficult, as the crew have to adapt to a variable frequency.

The gain-scheduled $\mathcal{H}_\infty$ design algorithm is applied and implemented in simulation, as both a continuous time and a discrete time controller. The discrete time controller represents the computer implementation which would be required on the SRMS or the SSRMS.

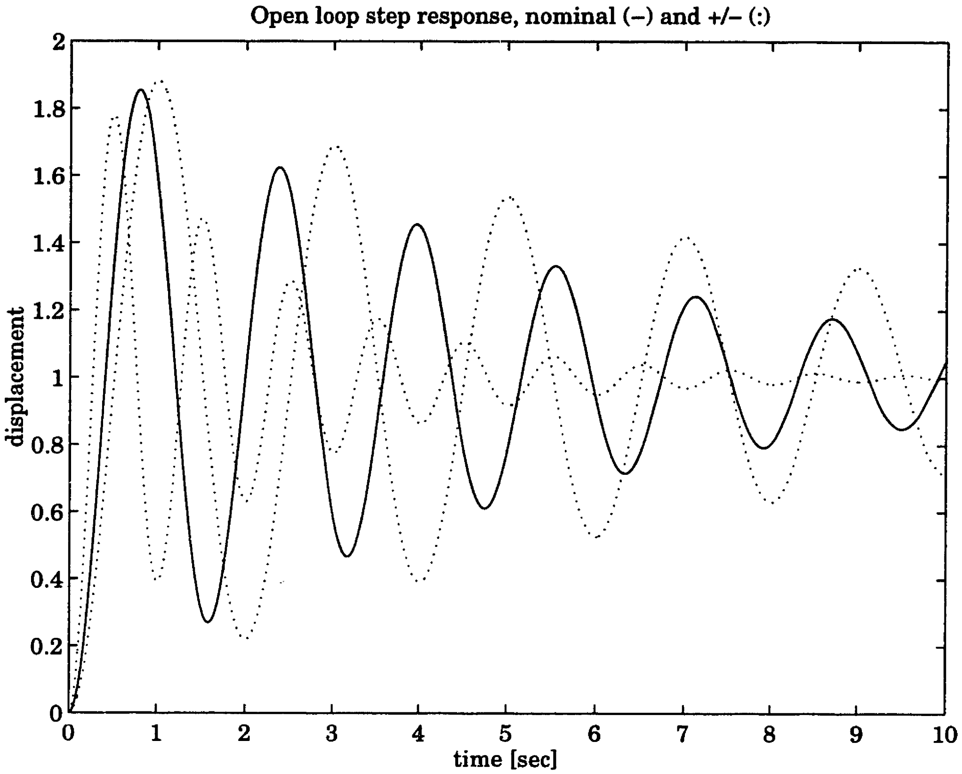

Figure 3-7: Open loop step response.

3.11.1 Reduced Order/LTI Control Design

The controller order reduction cost, J_{LTI}, and constant gain cost, J_{CG}, of Section 3.10 are used to design the sample controller. In order to understand the effect of the minimizations, controllers for the sample problem will be designed using neither cost, the LTI cost, the CG cost, and then both costs. The effect on compensator order and Δ block size will be examined.

The controller order and Δ block size for each of the four cases are shown in Table 3.1. From the left, the table shows the cost function used (a dash denotes that an LMI feasbility problem is solved rather than a cost function minimization subject to LMI feasibility), weighted plant order n, plant Δ block size r, controller order n_c, and controller Δ block size r_c. From the top, the controllers are full order, full delta block; reduced order, full delta block; full order, reduced delta block; and reduced order, reduced delta block. Implicit in the LMI feasibility conditions is the

Table 3.1: Compensator order and Δ block size for combinations of the cost functions.

Design cost	n	r	n_c	r_c
-	6	1	6	1
J_{CG}	6	1	4	1
J_{LTI}	6	1	6	1
$J_{CG} + J_{LTI}$	6	1	4	1

requirement that the $\mathcal{H}_\infty$ gain $\gamma < 1$.

Note that the minimum controller order that satisfies the $\mathcal{H}_\infty$ criteria is 4. Heuristically, the minimum order is a function of the performance and control weights chosen. Tightening the weights increases controller order. Note that the size of the controller delta block cannot be reduced. In this case the effect of the cost is to reduce the norm of the delta block. That is, the compensator variation with respect to Δ are reduced. The controller designed with the J_{CG} cost will still be referred to as a reduced delta block controller, even though the delta block size has not been reduced.

Note also that the controller order can be reduced regardless of whether the delta block reduction cost is used. This does not have to be true - the controller order may be traded off with delta block size in general.

3.11.2 Reduced Order Controller

The reduced order, reduced delta block controller will be examined. The compensator is 4^{th} order. Compensator pole locations, frequency and damping ratios, for the nominal mass, are as shown in Table 3.2. Compensator zeros are shown in Table 3.3. Note that none of the compensator zeros cancel the nominal open loop mode at 0.635 Hz. Such cancellations are typical of $\mathcal{H}_\infty$ optimal controllers, and are nonrobust to modeling errors. Even though the time-varying model used in the robust control design accounts for parameter changes, errors in the model may still result in

Table 3.2: Compensator poles for the typical section sample problem.

s-plane	freq [Hz]	ζ
-0.0171	0.0027	1.0
-10.2198	1.6265	1.0
$-34.6941 \pm 20.1617\jmath$	6.3864	0.8646

Table 3.3: Compensator zeros for the typical section sample problem.

s-plane	freq [Hz]	ζ
$0.5644 \pm 0.9039\jmath$	0.1696	-0.5296
-20.8814	3.3234	1.0000
93.1493	14.8252	-1.0000

Table 3.4: Closed loop poles for the typical section sample problem.

s-plane	freq [Hz]	ζ
-0.3340	0.0532	1.0000
$-2.5465 \pm 3.8586\jmath$	0.7358	0.5508
-5.4629	0.8694	1.0000
$-34.5664 \pm 18.5086\jmath$	6.2404	0.8816

mismodelling of system frequencies for a given value of the parameter. The absence of pole/zero cancellations in the gain-scheduled compensator gives reason to think that such designs may succeed even in the presence of modeling error. Closed loop poles, again for the nominal mass, are given in Table 3.4. Again it is apparent that the frequency and damping of the plant mode has been increased.

The closed loop error is shown in Figure 3-8, for the nominal and maximum and minimum payload cases, along with the inverse of the sensitivity weighting. Note that the error has been pushed below the desired level represented by the weight. This indicates that below 0.03 Hz, the payload positioning error is below 1% for all payloads. Note also the increase in sensitivity around 0.5 Hz. This is a consequence

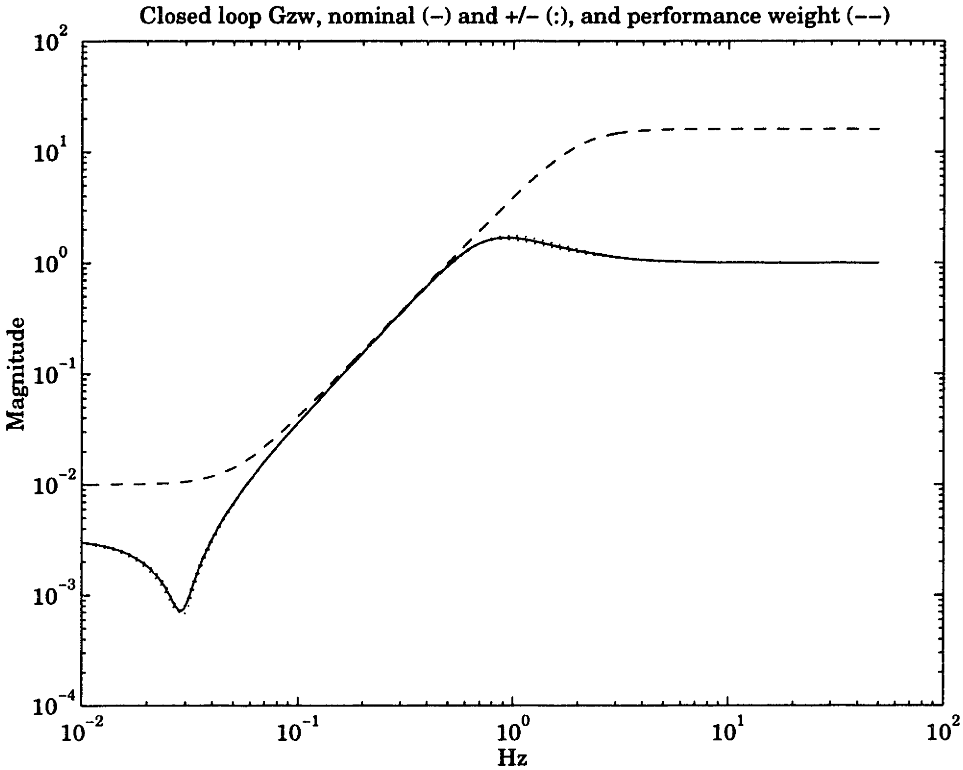

Figure 3-8: Closed loop reference input to sensitivity performance output.

of enforcing small error at low frequency. Finally, note that the closed loop natural frequency has increased over the open loop, and that the frequency is the same for all payloads.

The closed loop step response is shown in Figure 3-9. As desired, the highly resonant response of the open loop system has been replaced with a faster, more highly damped response for all cases. Note also that the response time (settling and rise time) is nearly identical for all payloads. This will allow the operator to more easily input the correct feedforward commands, so as to avoid exciting oscillations.

The compensator FR from sensor to control is shown in Figure 3-10. Note that the compensator is essentially providing rate feedback to the first mode at $0.5 - 1$ Hz. The effect of gain scheduling manifests itself in the variation of the rolloff pole of the compensator, and in the gain at the natural frequency.

The FR from command to control input is plotted in Figure 3-11. One might expect that the compensator might place a zero in the command input channel,

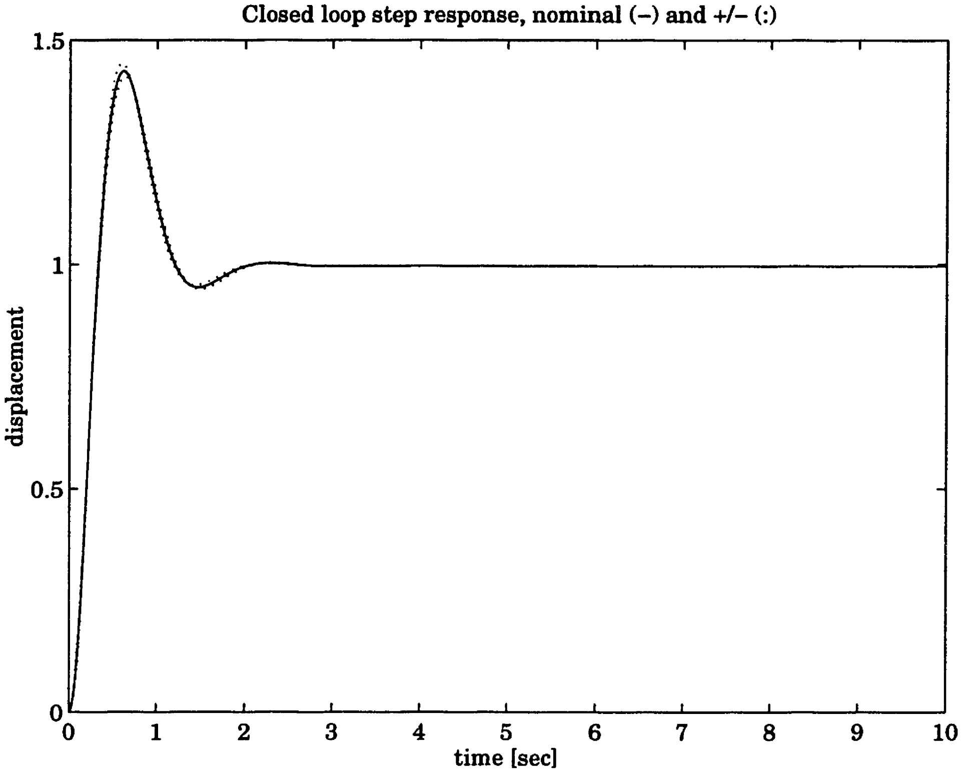

Figure 3-9: Closed loop step response.

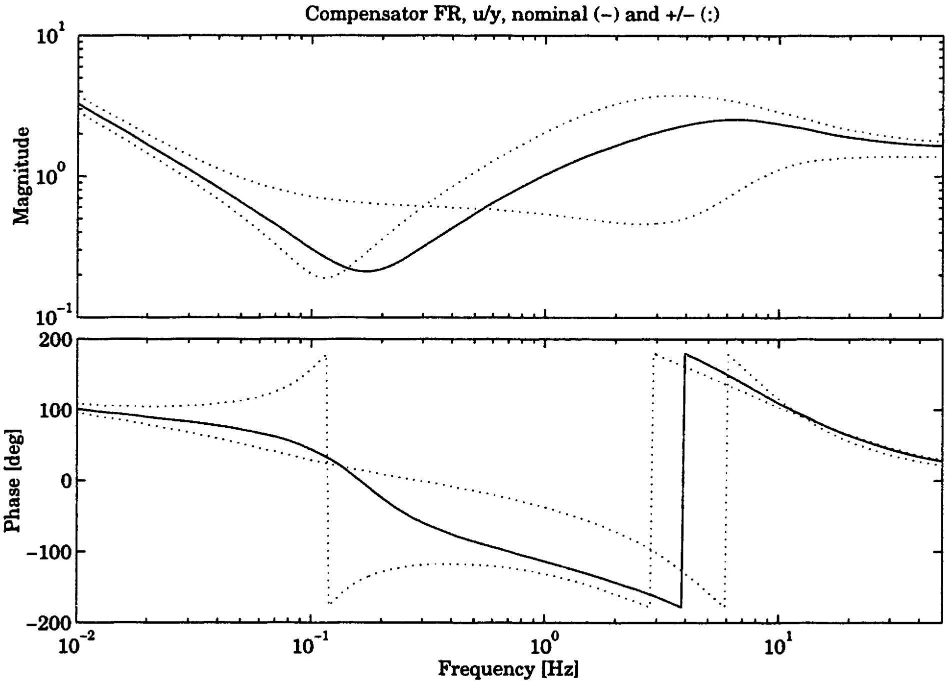

Figure 3-10: Compensator sensor input to control output.

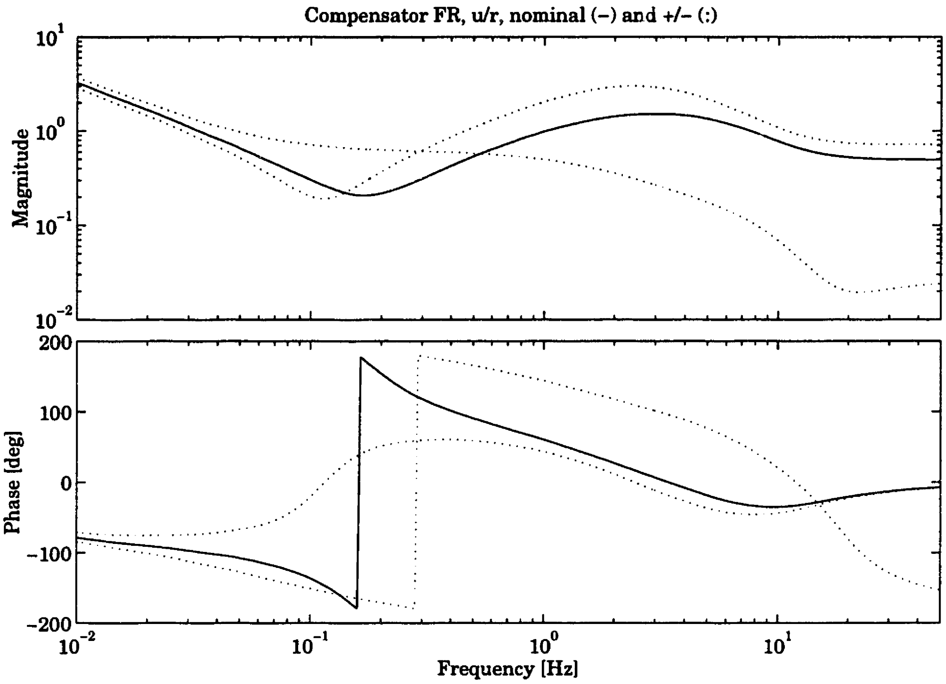

Figure 3-11: Compensator reference input to control output.

canceling the open loop mode. This would explain the higher performance that can be achieved when the reference command is fed to the compensator. However, it is clear that this is not the case.

The control loop FR is shown in Figure 3-12. Note that the loop gain is high at low frequencies, and at the modal frequency of the plant. The low frequency gain is needed to control the error at low frequencies, so as to push as close to the performance bound as possible without exceeding it. The result is to *minimize* the sensitivity improvement to just what was asked for. Due to "push-pop", additional error reduction would have to be made up at high frequencies. This characteristic (of pushing up to the performance bound) is generally associated with the $\mathcal{H}_\infty$ optimal compensator. In this case, it is due to the "minimum rank" controllers produced by the trace-minimization procedure discussed in Section 3.10.

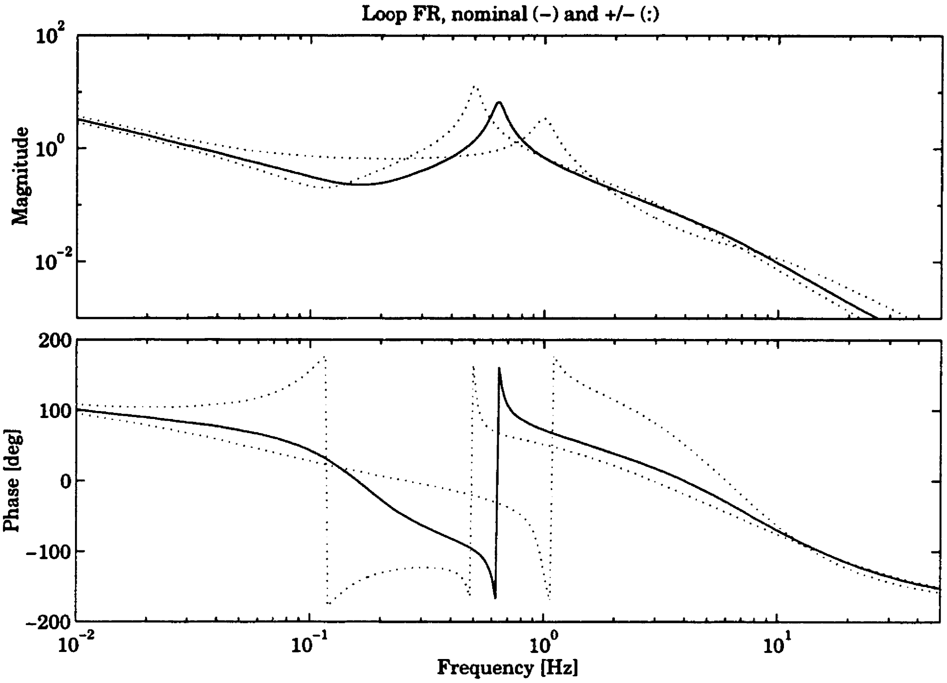

Figure 3-12: Loop transfer function.

3.12 Conclusion

The present chapter has discussed the utility of a parameterized nonlinear model of geometrically nonlinear systems. Such a model would allow the use of modern robust control design techniques for the control of flexible manipulators. As shown in this chapter, appropriate metrics exist in the modern control framework to describe the manipulator performance objective. A specific control design algorithm was presented, which accounts for the time-varying nature of the system. Application of the method to a representative simplified model demonstrated its ability to create desirable performance improvements.

Next, attention must be paid to the creation of a suitably parameterized nonlinear model. The model must be an accurate representation of the true system, while maintaining a parameterization which allows for the use of linear time-varying robust control tools. Motivated by the results of the manipulator typical section control design study, a linear fractional (feedback) representation will be developed. The LFT model will allow the use of the small gain time-varying $\mathcal{H}_\infty$ control technique presented in this chapter.

Chapter 4

Modeling

The proposed control method requires a model of the manipulator, in a specific form. Dynamic modeling of flexible manipulators has been studied in depth. However, the issue of modeling for control has only recently begun to receive attention. For the purpose of this work, modeling for control should be understood as the process of producing a model which is of appropriate accuracy, complexity, and structure for the control design algorithm. Accuracy is required in order that the model-based controller be stable, and meet performance requirements. Accuracy may refer to correct modeling of all dynamic components (gear drives, flexible elements, etc), inclusion of all "strong" forces (friction), and geometries (offsets). Complexity refers to the run-time complexity, so that the controller can be implemented in real time. Complexity may refer to system order, or to the form of the equations which must be solved in real time. Structure refers to the fact that the model should be suitable for use with the desired control algorithms. It is clear that by these metrics the control design algorithm will affect the modeling algorithm.

Accuracy and complexity have been widely studied. Accuracy is constrained by the need for stability and performance on the true system. Complexity is constrained by the computer power available for control. It is the third property, structure, which may deserve more attention than it has received to date.

It is clear that by these metrics the control design algorithm will affect the modeling algorithm. The control algorithms studied in this thesis rely on a small gain approach which is linear-algebra based. As such the system model must be linear. Conventional approaches to linearizing a geometrically nonlinear model can be undesirable, for reasons which will be addressed. This fact motivates the search for a technique which automatically "breaks out" the nonlinear terms into a separate block. The remainder, which is linear, is directly suitable for use in a linear algebra based control framework.

The approach presented in this chapter is intended to build on previous modeling methods, not to replace them. Many modeling approach use a component-level description which separates the multibody system into multiple subsystems. The subsystems are typically then kinematically constrained to remain together at attachment locations (such as joints) by appending the constraints with Lagrange multipliers. The multipliers then represent the constraint forces. The key innovation which is proposed in the current work is the use of an input/output description of the interconnection of the links of a manipulator at each joint. The attachment of two components, such as links, at a joint is then described as a feedback interconnection between the two subsystems. Again uniquely to the current work, time-varying joint rotations are represented as a time-varying feedback gain between the components. As will be shown, the feedback interconnection results in a system realization which leads directly to a Linear Fractional linear system model. This model is shown to be exactly that required for the application of certain types of Time-Varying Robust Control.

The proposed approach builds on previous methods because it is structured so as to allow the component models to be generated using any of the four common modeling approaches: Lagrange's equations, Kane's equations, FEM, or simplified packages.

The chapter is organized as follows: the use of acceleration feedback to model time-varying boundary conditions is motivated by demonstrating its equivalence to mass-coupling. The modeling algorithm is outlined, and the method is presented as a series of "black box" steps which take specific input information from a previous step, operate on it, and output information to the following steps. The elements of

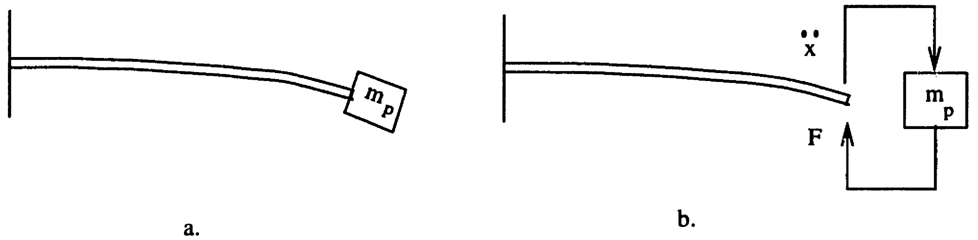

Figure 4-1: Mass-loading a beam is equivalent to feeding back acceleration to force through the static gain m_p.

the algorithm are then examined.

4.1 Proposed Modeling Approach

4.1.1 Acceleration Feedback

The choice of acceleration feedback is a natural one for the configuration nonlinearities inherent in manipulators. These manifest themselves in the system inertia matrix. Adding mass to a structure can be seen as acceleration feedback, since a mass produces a reaction force which is proportional to acceleration.

A natural demonstration takes the form of a cantilevered beam which is loaded with a payload mass m_p (Figure 4-1a). The beam represents the flexible link of a manipulator under joint position control. Taking the beam to be Bernoulli-Euler with uniform mass per unit length ρA and bending stiffness EI equal to one, and unit length, the kinetic energy of the beam with payload is

$$T(t) = \frac{1}{2}\int_0^1 \dot{w}(r,t)^2 \mathrm{d}r + \frac{1}{2} m_p \dot{w}(1,t)^2 \tag{4.1}$$

where r is the distance along the link, and $w(r,t)$ is the transverse velocity. Using a standard assumed modes approach in which the transverse displacement is separated into spatial and temporal components,

$$w(r,t) = \phi(r)^T q(t) \tag{4.2}$$

the kinetic energy becomes

$$T = \frac{1}{2}\dot{q}^T \left(\int_0^1 \phi(r)\phi(r)^T \mathrm{d}r + m_p \phi(1)\phi(1)^T \right) \dot{q} \tag{4.3}$$

$$= \frac{1}{2}\dot{q}^T \left(M_u + m_p\phi(1)\phi(1)^T\right)\dot{q} \tag{4.4}$$

for the unloaded mass matrix M_u. Potential energy is given by

$$P = \frac{1}{2}q^T \int_0^1 \phi(r)\phi(r)^T \mathrm{d}r q = \frac{1}{2}q^T K q \tag{4.5}$$

with K the stiffness matrix. Lagrange's equation then gives the unforced dynamics

$$\left(M_u + m_p\phi(1)\phi(1)^T\right)\ddot{q} + Kq = 0 \tag{4.6}$$

This is compared to acceleration feedback of gain m_p at the tip of the same beam, unloaded, with a tip force (Figure 4-1b):

$$M_u\ddot{q} + Kq = BF \tag{4.7}$$

The feedback is proportional to tip acceleration $a(1)$:

$$F = -m_p a(1) \tag{4.8}$$

The force influence vector is given by

$$B = \phi(1) \tag{4.9}$$

The acceleration is

$$a(1) = \phi(1)^T\ddot{q} \tag{4.10}$$

Therefore the closed loop EOM is

$$M_u\ddot{q} + Kq = -\phi(1)m_p\phi(1)^T\ddot{q} \tag{4.11}$$

or

$$(M_u + m_p\phi(1)\phi(1)^T)\ddot{q} + Kq = 0 \tag{4.12}$$

which is identical to Equation 4.6.

From a conceptual viewpoint, the technique describes inertial reaction forces as acceleration feedback. This is the justification for posing motion boundary conditions as accelerations rather than position or rate. From a mathematical point of view, the "motion" outputs must be invertible with the input forces of the same type and location (type describes the action of the input, i.e.. force or moment). There will be a feedthrough term (D term) in the corresponding transfer function. Simply stated, if there is no feedthrough, the transfer function rolls off with frequency (goes to zero), so the inverse increases with frequency (i.e. goes to infinity). Acceleration outputs contain the necessary feedthrough term.

By extension, the mass can be replaced with a dynamic system, for example another link. The static feedback gain m_p is replaced with a dynamic system. The link example above only imposed a transverse acceleration on the end mass. In the general case, the end component will have different dynamics depending on the orientation between it and the link. This must be captured in a rotating boundary condition.

4.2 System Modeling Algorithm

The proposed modeling approach has the following steps: system configuration definition, component modeling, component input/output inversion, boundary condition definition, and the final collection and computation of the system model. The steps are presented in Figure 4-2. Briefly, the steps are:

(i) Determine the number, type, and arrangement of components.

(ii) Model each component as a free-free body, with force inputs and acceleration outputs at each attachment location.

(iii) Invert the force input and acceleration output of one attachment point per component.

(iv) Define the boundary conditions between attached components.

(v) Assemble the system model.

The same sequence of steps can be used to produce a nonlinear structural model for simulation, and a parameterized linear model for control. Each of the steps will be discussed in the present section. Then, in the following sections the steps will be

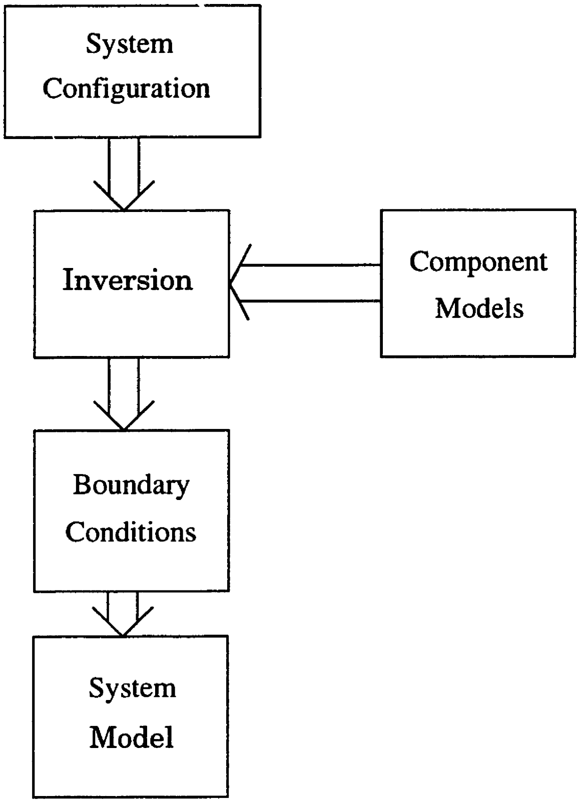

Figure 4-2: Modeling algorithm.

further broken down into the information which must be provided as input, and the information which results as output.

The configuration step consists of defining the number, type, and location of each component (shown in cartoon form in Figure 4-3). Type refers to the physical component - link, gearbox, etc. Location determines which components are attached and where. Configuration definition will also determine the location and type of time-varying boundary conditions. For example, a gearbox will involve a time-varying rotation (the dotted semicircles).

The free-free models of each component are created in the component model step. The component model can be of arbitrary complexity, from linear to nonlinear in rigid body rates, to fully nonlinear in deflections. The essential elements are that all rigid body modes be represented, and that force influence matrices, and acceleration output equations (in terms of the component states) be created at all attachment

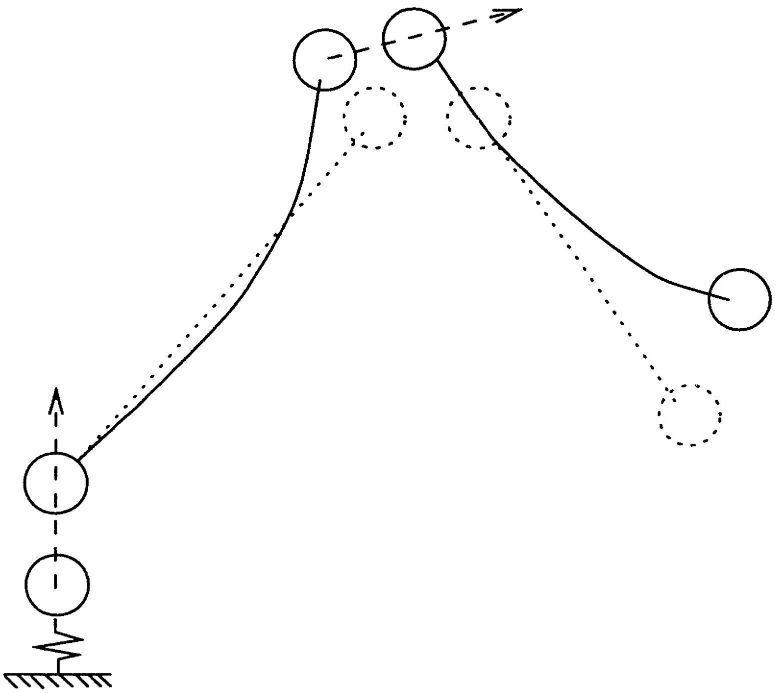

Figure 4-3: Cartoon showing elements of the configuration.

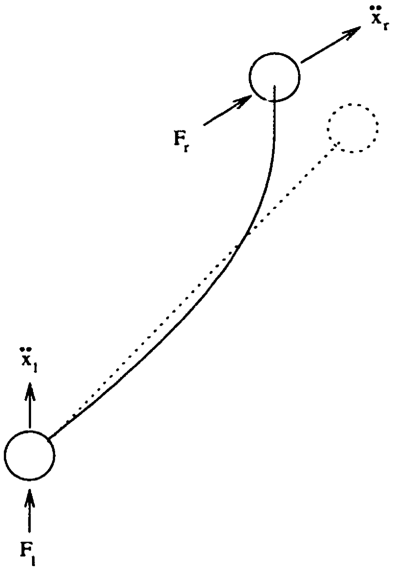

Figure 4-4: Cartoon showing free-free component.

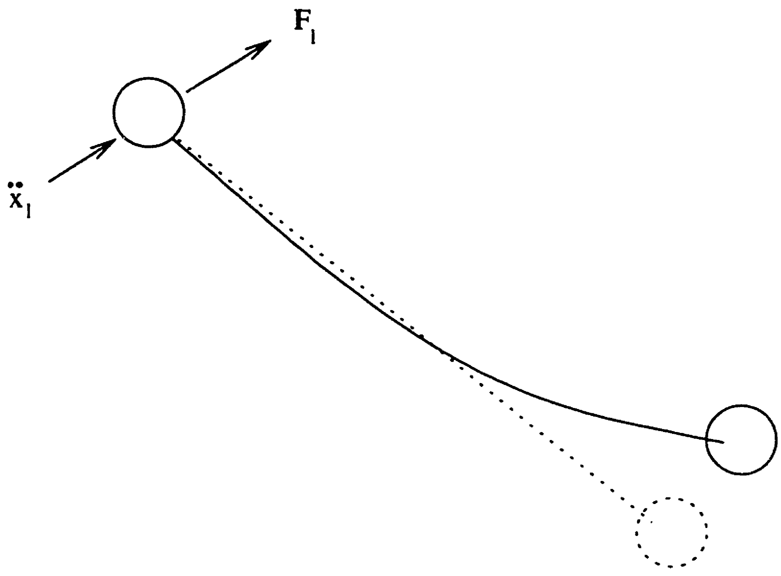

Figure 4-5: Inversion of the interface boundary conditions for one component allows two components to be connected via feedback.

points (Figure 4-4). Note that other physical outputs (for example, the outputs of other sensors such as strain gauges) can be solved for on a component level. In addition, the configuration will flow down to component model requirements. For example, the required component states will be identified. Component attachment locations identify the locations of influence and output equations.

The input/output inversion is the means by which acceleration and force boundary conditions (BCs) between components are equated. The free-free model has all component DOF independent. Forces act on the component and accelerations result. In order to "attach" two components, the forces and accelerations of each must be equated. However, the accelerations are functions of forces, which are in turn functions of acceleration, and so on. One approach to solving for the accelerations and forces in one step is to invert the force/acceleration map of one of the components at the interface (Figure 4-5). Now the component is constrained in acceleration at that location and produces a force in response. Thus the acceleration output of the original component drives the acceleration input of the second link, and the force output of the second link drives the force input of the first (Figure 4-6a). However, the joint angle varies, and the forces which impinge on the second link vary with the

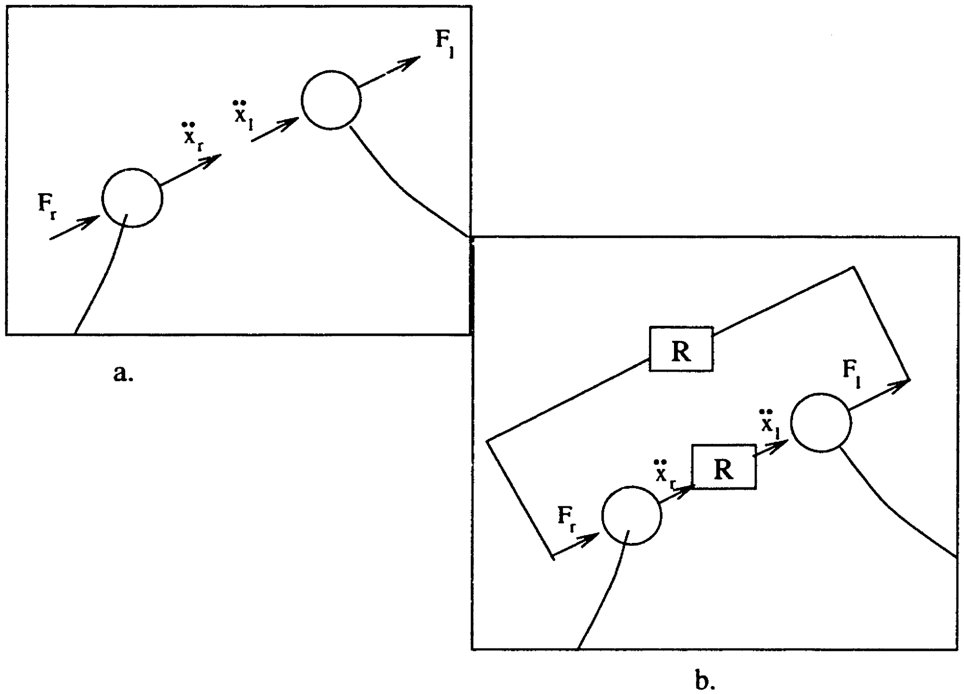

Figure 4-6: Inversion of the interface boundary conditions for one component allows two components to be connected via feedback.

joint angle. It is necessary to rotate the acceleration outputs of one link into the reference frame of the second, and similarly for the forces (Figure 4-6b).

The system model consists of

(i) the homogeneous dynamics of each component

(ii) the boundary condition equations which represent the rotations between components

See Figure 4-7. The homogeneous dynamics are solved first, and used to calculate the contribution of each component to the boundary conditions. Next the rotation boundary condition is formed and solved as a single matrix equation. Finally the contributions of the boundary conditions to the component EOM are calculated. Note that the component models are independent, but the boundary conditions are not. The term "system model" thus refers to the collection of component models and the boundary condition equation, i.e. the coupled system. For the nonlinear system these equations are solved at run time. The parameterized linear model is created

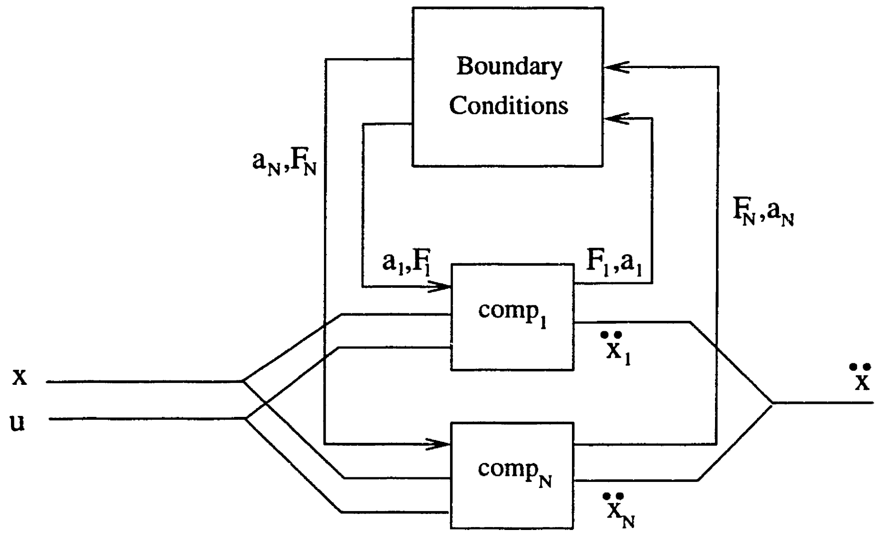

Figure 4-7: The system model consists of N independent component models, and a global boundary condition solution which relates the components via feedback. u are exogenous commands.

by coupling the component models. The boundary condition equation is evaluated separately, as a structured block matrix.

4.3 Configuration

The configuration of the system is set by the physical structure being modeled. The important items of information are the number, type, and location of components. Specifically, the input to the configuration step is the system model to be created, which includes:

(i) global DOF

(ii) physical system parameters

(iii) joint locations

The outputs of the configuration definition are

(i) component model DOF, reference frames, and boundary locations

(ii) boundary condition coupling (rotation) conditions

The output information is used in the component model step and the boundary condition step, respectively.

For the manipulator problems which motivate the current work, two types of components are of interest: *links* and *revolute joints.* A link will refer to a long, thin, flexible member. The link will generally have a length parameter, transverse bending mass and stiffness, and possibly torsional flexibility. A revolute joint will refer to a mechanism for producing a commanded rotation about a single axis. The manipulators under consideration use a geared torque motor to produce rotation. The specific component used to capture this will be called a gearbox. The gearbox will have mass and inertia parameters, a gear reduction, and a commanded torque input. Typically the links will be attached to one or more gearboxes. The systems to be considered will be fully three-dimensional.

These components will be enough to represent the manipulator systems to be investigated. Extensions to these components could include, if necessary; linear joints, which extend and contract; additional flexibility, such as the *ovalization* modes of a hollow link with circular cross-section, in which the cross-section deforms to an oval shape; and arbitrary, possibly flexible, payloads. Each of these could be included in the modeling framework.

4.4 Component Models

The component models are inputs to the modeling algorithm. In order that the modeling algorithm be as general as possible, the component models are input in a standard form which can be created using any of the modeling approaches described in Chapter 1. The succeeding chapter, Chapter 5, will present one approach for deriving component models in a suitable form. For each component, the input information consists of:

(i) The basis for the component DOF.

(ii) Reference frames in which attachment boundary conditions must be expressed.

(iii) Attachment locations where forces and accelerations must be calculated.

The first specifies the rigid body and flexible motions which must be accounted for. For example, planar systems will need only a subset of the full six rigid body DOF. Flexible motions may include transverse vibration, torsion, or shear modes, depending on the kinematic DOF of the system being simulated (planar, etc) as well as the mass and stiffness properties and the bandwidth of the input. The output information produced by the component modeling step consists of

(i) The homogeneous dynamics of each component.

(ii) State-dependent coupling forces.

including the influence of input forces at the attachment locations, and the resulting accelerations at the attachments. It should be noted that typically the reference frames will be body fixed at the attachment locations, so that the attachment locations define the reference frames. State-dependent coupling forces are those which occur between two or more components, as a function of the free component DOF, for example, gearbox stiffnesses or joint friction.

It should be emphasized that the term *location* is a potentially misleading one. The constrained accelerations and free forces may be colocated. For example, the gearbox component model which will be presented in Chapter 5 has no spatial offsets (these are captured in the link model). In this case, recourse must be made to the physics of the attachment. The constraint model is such that a constrained location produces an inertial reaction force. The free force which acts at the same physical location will simply sum into the reaction force. The corresponding force/acceleration input/output pair in the model will have a direct feedthrough, from free location force input to constrained location output force. This is in fact the case for offset constrained/free locations as well, although it is not as apparent.

The general form of the component models will be

$$\begin{aligned} M\ddot{q} &= (-V(q,\dot{q}) + B_u u) + B_c F_c + B_f F_f \qquad (4.13) \\ a_c &= H_c(q,\dot{q}) + G_c\ddot{q} \end{aligned}$$

$$a_f = H_f(q,\dot{q}) + G_f\ddot{q}$$

This would apply to systems with multiple attachment points (tree structures as well as closed chains). For notational reasons, however, each independent attachment location should be broken out separately. As a result, in this work, the above system should be read as having a single constrained end, a single free end, and as a component in an open chain. Again, this is purely notational. Note that the acceleration output functions will typically have terms proportional to the generalized accelerations, as expected. The generalized displacement and rate terms, H_c, and H_f, are essentially centripetal acceleration terms.

4.5 Model Inversion

The component models are in a free-free form. That is, using the link component as an example, the forces at each end are specified, and the link is free to move in response. The corresponding transfer function matrix of the component, from boundary forces to boundary accelerations, is

$$\begin{bmatrix} a_l \\ a_r \end{bmatrix} = \begin{bmatrix} G_{ll}(s) & G_{lr}(s) \\ G_{rl}(s) & G_{rr}(s) \end{bmatrix} \begin{bmatrix} F_l \\ F_r \end{bmatrix} \tag{4.14}$$

with accelerations and forces as defined in previous sections. The component will be attached in a feedback fashion to preceding and succeeding components. To this end, one pair of inputs and outputs are inverted. The resulting component model has the form

$$\begin{bmatrix} F_l \\ a_r \end{bmatrix} = \begin{bmatrix} G_{ll}^{-1} & -G_{ll}^{-1}G_{lr} \\ G_{rl}G_{ll}^{-1} & G_{rr} - G_{rl}G_{ll}^{-1}G_{lr} \end{bmatrix} \begin{bmatrix} a_l \\ F_r \end{bmatrix} \tag{4.15}$$

In this form, the outputs can be fed to an attached component, and the inputs driven by the attached component.

The BC input/output inversion is performed on the component state space system matrices derived above. The inversion involves constraining the motion of the boundary. That is, specific combinations of model DOF will be specified. In effect, these DOF are deleted from the model. The inversion begins by partitioning the states into those which will be constrained, q_c, and those which will remain free, q_f. The free-free model, with force inputs, in second order form is

$$\begin{aligned}
\begin{bmatrix} M_{cc} & M_{cf} \\ M_{cf}^T & M_{ff} \end{bmatrix} \begin{bmatrix} \ddot{q}_c \\ \ddot{q}_f \end{bmatrix} &= - \begin{bmatrix} V_c \\ V_f \end{bmatrix} + \begin{bmatrix} B_{cc} & B_{cf} \\ B_{fc} & B_{ff} \end{bmatrix} \begin{bmatrix} F_c \\ F_f \end{bmatrix} \qquad (4.16) \\
a_c &= H_c + \begin{bmatrix} G_{cc} & G_{cf} \end{bmatrix} \begin{bmatrix} \ddot{q}_c \\ \ddot{q}_f \end{bmatrix} \\
a_f &= H_f + \begin{bmatrix} G_{fc} & G_{ff} \end{bmatrix} \begin{bmatrix} \ddot{q}_c \\ \ddot{q}_f \end{bmatrix}
\end{aligned}$$

The inputs and outputs are partitioned conformably with the constrained locations (subscripted c) and free locations (subscripted f) on the element. By convention, the base (left) end of each link will be constrained, except for the base local y axis. The base y axis is by convention the gearbox shaft axis and is free to rotate.

Note that each force or acceleration is either a driving quantity, if it acts as an input to the component, or an output quantity, if it is a component output. Driving quantities occur in pairs with output quantities. So a driving force is paired with an output acceleration at the same location. A driving acceleration is paired with an output force.

The inversion is performed on the second-order state space system by solving for the constrained state accelerations $\ddot{q}_c$ in terms of the constrained boundary accelerations a_c; using that expression to solve for the constrained force outputs F_c and accelerations a_f; and using $\ddot{q}_c$ and F_c to write the EOM of the free DOF. The resulting dynamics in the states $q = q_f$ can be expressed as

$$\hat{M}\ddot{q} = \hat{V} + \hat{B}_a a_c + \hat{B}_F F_f \qquad (4.17)$$

$$F_c = \hat{H}_F + \hat{G}_{Fa} a_c + \hat{G}_{FF} F_f + \hat{G}_F \ddot{q} \qquad (4.18)$$

$$a_f = \hat{H}_a + \hat{G}_{aa} a_c + \hat{G}_a \ddot{q} \qquad (4.19)$$

with

$$\hat{M} = M_{ff} - B_{fc} B_{cc}^{-1} \left(M_{fc}^T - M_{cc} G_{cc}^{-1} G_{cf} \right) - M_{fc} G_{cc}^{-1} G_{cf}, \qquad (4.20)$$

$$\begin{aligned}
\hat{V} &= -V_f + B_{fc}B_{cc}^{-1}\left(V_c - M_{cc}G_{cc}^{-1}H_c\right) + M_{fc}G_{cc}^{-1}H_c, && (4.21)\\
\hat{B}_a &= B_{fc}B_{cc}^{-1}M_{cc}G_{cc}^{-1} - M_{fc}G_{cc}^{-1}, && (4.22)\\
\hat{B}_F &= B_{ff} - B_{fc}B_{cc}^{-1}B_{cf}, && (4.23)\\
\hat{H}_F &= B_{cc}^{-1}\left(V_c - M_{cc}G_{cc}^{-1}H_c\right), && (4.24)\\
\hat{G}_{Fa} &= B_{cc}^{-1}M_{cc}G_{cc}^{-1}, && (4.25)\\
\hat{G}_{FF} &= -B_{cc}^{-1}B_{cf}, && (4.26)\\
\hat{G}_F &= B_{cc}^{-1}\left(M_{fc}^T - M_{cc}G_{cc}^{-1}G_{cf}\right), && (4.27)\\
\hat{H}_a &= H_f - G_{fc}G_{cc}^{-1}H_c, && (4.28)\\
\hat{G}_{aa} &= G_{fc}G_{cc}^{-1}, && (4.29)\\
\hat{G}_a &= G_{ff} - G_{fc}G_{cc}^{-1}G_{cf} && (4.30)
\end{aligned}$$

where the subscript c refers to an input or output (BC) at the constrained end, and f refers to a BC at the free end, as before, and the new subscripts F and a denote a "force" or "acceleration" BC, respectively.

Note the presence of the B_{cc}^{-1} and G_{cc}^{-1} inverse terms in the constrained model. B_{cc} represents the influence of the forces at the constrained location, on the "constrained" DOF. B_{cc} is invertible whenever the number of force inputs is equal to the number of constrained states (i.e.. is square) and the forces are independent. G_{cc} represents the "observability" of the constrained DOF to the acceleration at the constrained location. Since the attachment forces and accelerations are fixed by the configuration, the modeler must choose the correct "constrained DOF" to ensure the invertability of these two matrices. Generally the correct choice of DOF will be readily apparent.

These equations hold for a general system of the form of Equation 4.16. Intelligent choice of generalized coordinates can reduce the complexity of the constrained system representation. By choice of rigid body DOF, the left end (constrained) forces act directly on the constrained states. In addition, the cantilevered flexible mode shapes have zero displacement and rotation at the root, leading to a zero force influence matrix at the left end. Similarly, the accelerations at the left end of the link are functions only of the rigid body modes. Thus

$$B_{cc} = I, \tag{4.31}$$

$$B_{fc} = 0, \tag{4.32}$$

$$H_c = 0, \tag{4.33}$$

$$G_{cc} = I, \tag{4.34}$$

$$G_{cf} = 0 \tag{4.35}$$

and the standard model representation simplifies to

$$\hat{M} = M_{ff}, \tag{4.36}$$

$$\hat{V} = -V_f, \tag{4.37}$$

$$\hat{B}_a = -M_{fc}, \tag{4.38}$$

$$\hat{B}_F = B_{ff}, \tag{4.39}$$

$$\hat{H}_F = V_c, \tag{4.40}$$

$$\hat{G}_{Fa} = M_{cc}, \tag{4.41}$$

$$\hat{G}_{FF} = -B_{cf}, \tag{4.42}$$

$$\hat{G}_F = M_{fc}^T, \tag{4.43}$$

$$\hat{H}_a = H_f, \tag{4.44}$$

$$\hat{G}_{aa} = G_{fc}, \tag{4.45}$$

$$\hat{G}_a = G_{ff} \tag{4.46}$$

Careful choice of generalized coordinates is therefore necessary, as it has the potential to simplify the modeling and simulation cost. Symbolic inversion of B_{cc} and G_{cc} may be difficult, and each additional matrix expression in the system EOM corresponds to additional real-time computational load.

The expression of the final model will depend on the desired use. The free mass matrix M_{ff} is, in its most general form, a function of the deflection variables (thus time). Inverting it off-line is computationally intensive, and the storage and computation requirements may be as high as those needed to numerically invert it on-line. Hence these equations may be used essentially as presented above.

4.6 Boundary Conditions

The boundary condition definition step consists of assembling the system force and acceleration boundary conditions at the joints. The input information, for each joint, consists of

(i) The relative motion across the joint, between the coupled components.

(ii) The independent and dependent forces and accelerations for the components coupled at the joint.

The first item consists, for example, of the rotation of a revolute joint as a function of the states of the coupled components. The second consists of the output forces and accelerations, and the driving forces and accelerations, for the coupled components. The output consists of

(i) the global coupling condition for the components, which equates the accelerations and forces at the coupling locations

The global coupling condition takes the form of a linear matrix equation.

The boundary conditions equate forces and moments, and linear and angular accelerations across the joint. The boundary conditions which relate generalized forces and accelerations will be developed separately. Then the relations can be used as building blocks to assemble the global coupling matrix.

The rotation across the joint consists of the angular difference between the "outboard" component's rotation around the shaft axis, and the "inboard" component's rotation about the same axis. Note that the shaft axes for both components are constrained to be collinear. Since the rotations of each component, at the attachment location, are expressed in a frame body-fixed to the attachment location, the joint rotation is a function of a single angle. For a given link i, the rotation to the previous link, towards the base of the manipulator, is R_i.

Across each joint, each constrained quantity is paired with a free quantity. The component boundary conditions are developed with the constrained quantity on the

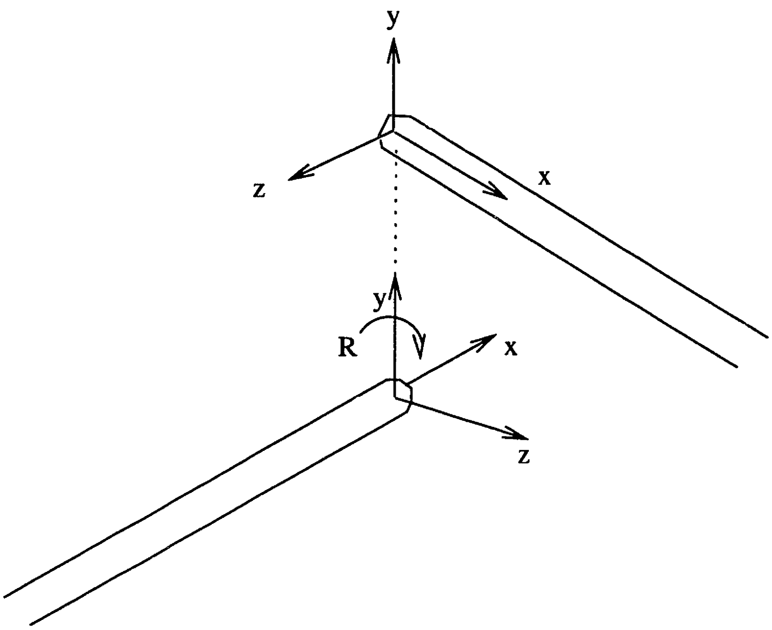

Figure 4-8: The joint rotation between links $i-1$ (left) and i (right). The origins of the frames coincide, and the y axes are in the same direction. Link $i-1$ has force inputs F_{i-1} and acceleration outputs a_{i-1}, link i has inputs a_i, outputs F_i.

left, and the free quantity on the right. The attachment between link a given link i and the link inboard to it, link $i-1$, consists of a generalized force input F_{c_i} equated with a force output F_{f_j}, and a generalized acceleration input a_{c_j} equated with an acceleration output a_{f_i} (see Figure 4-8). In general the constrained accelerations will include linear as well as angular terms. For the purposes of exposition, linear terms will be represented as the vector $\ddot{p}$, and angular terms as $\ddot{\Theta}$.

Linear accelerations are equated with the boundary condition

$$\ddot{p}_{c_i} = R_i \ddot{p}_{f_{i-1}} \tag{4.47}$$

Note that the left hand side, $\ddot{p}_{c_i}$, represents an input to link i. The term on the right, $\ddot{p}_{f_{i-1}}$, represents an output of the previous link.

Angular acceleration boundary conditions are found by equating the angular velocity components at the interface:

$$\dot{\Theta}_{c_i} = R_i \dot{\Theta}_{f_{i-1}} \tag{4.48}$$

From this,

$$\ddot{\Theta}_{c_i} = \dot{R}_i \dot{\Theta}_{f_{i-1}} + R_i \ddot{\Theta}_{f_{i-1}} \tag{4.49}$$

$\dot{\Theta}_j$ is a known function of the state variables of links i and $i-1$. The force condition is

$$F_{f_{i-1}} = -R_i^T F_{c_i} \tag{4.50}$$

Therefore the input BCs are related to the output BCs by

$$\begin{bmatrix} F_{f_{i-1}} \\ a_{c_i} \end{bmatrix} = \begin{bmatrix} 0 & -R_i^T \\ R_i & 0 \end{bmatrix} \begin{bmatrix} a_{f_{i-1}} \\ F_{c_i} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{R}_i \end{bmatrix} \dot{\Theta}_{f_{i-1}} \tag{4.51}$$

Each joint will contribute such a block term to the global coupling matrix.

Note also that the coupling term between any two components is a linear matrix equation in the constrained forces and accelerations. As a result the system coupling matrix will also take the form of a linear matrix equation, as follows. The input BCs for each component can be stacked into the vector u_{BC}, and the outputs into the vector y_{BC}. The rotation matrices are collected into a matrix with the following form:

$$R = \begin{bmatrix} 0 & -R_1^T & \cdots & 0 & 0 \\ R_1 & 0 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 0 & -R_{N-1}^T \\ 0 & 0 & \cdots & R_{N-1} & 0 \end{bmatrix} \tag{4.52}$$

for $N-1$ joints in an N-component open chain body. Collecting the known outputs into the vector H, the joint angular rates into the vector H_r, and the BC-dependent parts into the block diagonal matrix G, the inputs are related to the outputs by

$$u_{BC} = Ry_{BC} + \dot{R}H_r \tag{4.53}$$

Using the global vector representation for the output BCs,

$$y_{BC} = H + Gu_{BC} \tag{4.54}$$

the BC equation, Equation 4.53, becomes

$$u_{BC} = RH + RGu_{BC} + \dot{R}H_r \tag{4.55}$$

which has the solution

$$(I - RG)u_{BC} = RH + \dot{R}H_r \tag{4.56}$$

This is a linear matrix equation which can be solved numerically using readily available algorithms.

4.7 System Model

The system model is created by appending the subsystem equations of motion together, solving for the inter-system constraint forces and accelerations, and summing in the contributions of the constraints to the dynamics. For both the linear and nonlinear model, the inputs are

(i) Component models.

(ii) Component boundary condition equations.

(iii) State dependent coupling terms.

The output of the system modeling step is slightly different for the nonlinear model and for the linear control form model. In each case the output is the system model. However, the simulation form of the model consists of

(i) Second-order state space component models.

(ii) State dependent coupling terms.

(iii) System boundary condition.

The simulation form is assembled at run time to produce the generalized accelerations from the state and inputs. The control form consists of

(i) Coupled first order component EOM.

(ii) The parameterized system boundary condition.

The system model for a particular joint configuration is determined from a linear fractional transformation on the component EOM through the boundary condition at the joint configuration.

The *simulation form* allows the mass matrix and stiffness dependent forces to be time varying. The homogeneous component dynamics are determined independently, the attachment forces determined, and resulting motion summed into the homogeneous dynamics. This sequential solution allows for streamlined calculation. The *control form* is used in conjunction with the robust time varying control design algorithms discussed in Chapter 3. The dynamics must be in the form of a linear time-invariant state space model, with a time-varying linear fractional block. This restriction imposes additional computational burden, compared to the simulation form. For this reason the control form will generally be based on a linearized version of the mass matrix and state dependent forces. The simulation form will be presented first.

4.7.1 Simulation Form

As previously noted, the system model consists of the homogeneous dynamics of each component, plus the boundary condition equations which relate components. Re-writing the component EOM in their general form,

$$\begin{aligned} M_i\ddot{q}_i &= V_i + B_{a_i}a_{c_i} + B_{F_i}F_{f_i} && (4.57)\\ F_{c_i} &= H_{F_i} + G_{Fa_i}a_{c_i} + G_{FF_i}F_{f_i} + G_{F_i}\ddot{q}_i && (4.58)\\ a_{f_i} &= H_{a_i} + G_{aa_i}a_{c_i} + G_{a_i}\ddot{q}_i && (4.59)\end{aligned}$$

for $i = 1..N$, for N components. The boundary condition equations consist of

$$\begin{bmatrix} F_{f_{i-1}} \\ a_{c_i} \end{bmatrix} = \begin{bmatrix} 0 & -R_i^T \\ R_i & 0 \end{bmatrix} \begin{bmatrix} a_{f_{i-1}} \\ F_{c_i} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{R}_i \end{bmatrix} \dot{\Theta}_i \qquad (4.60)$$

for $N-1$ interfaces, assuming an open chain manipulator (that is, each interface connects only two components). It is seen that the EOM consist of a known portion $M^{-1}V$, and boundary-condition dependent contributions. Note also that the state accelerations appear in the output BCs. Since the state accelerations consist of a homogeneous portion and a contribution from the attachment forces, the output BCs also consist of a "homogeneous" portion and a contribution from the attachment forces:

$$F_{c_i} = (H_{F_i} + G_{F_i} M_i^{-1} V_i) + \begin{bmatrix} G_{Fa_i} + G_{F_i} B_{a_i} & G_{FF_i} + G_{F_i} B_{F_i} \end{bmatrix} \begin{bmatrix} a_{F_i} \\ F_{F_i} \end{bmatrix} \tag{4.61}$$

$$a_{f_i} = (H_{a_i} + G_{a_i} M_i^{-1} V_i) + \begin{bmatrix} G_{aa_i} + G_{a_i} B_{a_i} & G_{a_i} B_{F_i} \end{bmatrix} \begin{bmatrix} a_{F_i} \\ F_{F_i} \end{bmatrix} \tag{4.62}$$

where by extension of the terminology for the dynamics, $H_{F_i} + G_{F_i} M_i^{-1} V_i$ and $H_{a_i} + G_{a_i} M_i^{-1} V_i$ are the homogeneous output BCs.

Simulation Form Dynamics Solution Algorithm

The following solution algorithm is used to calculate the system accelerations $\ddot{x}$ at each solution time step.

(i) Calculate forces which depend on the global state x and $\dot{x}$, such as

  (a) gearbox stiffnesses

  (b) friction

  (c) control forces

  Note that global state dependent forces include those which act between any two components, such as gearbox stiffness. In other words, they need not depend on the entire state; only more than a single component.

(ii) For each component:

  (a) Solve the homogeneous dynamics

$$M\ddot{q}_H = V \tag{4.63}$$

  This is the most computationally intensive step of the component dynamics calculations. Note that M is in general a full matrix. It is also in general a time-varying function, so such common steps as diagonalizing the mass matrix via a state transformation are not straightforward. Also note that the homogeneous dynamics V include state-dependent forces calculated earlier.

(b) Compute the homogeneous output BCs

$$H_i = \begin{bmatrix} F_{c_i} \\ a_{f_i} \end{bmatrix} = \begin{bmatrix} H_F \\ H_a \end{bmatrix} + \begin{bmatrix} G_F \\ G_a \end{bmatrix} \ddot{q}_H \tag{4.64}$$

(iii) For the global system, collect the component output vectors:

$$H = \begin{bmatrix} H_{a_1} + G_{a_1}\ddot{q}_{H_1} \\ H_{F_2} + G_{F_2}\ddot{q}_{H_2} \\ H_{a_2} + G_{a_2}\ddot{q}_{H_2} \\ \vdots \\ H_{F_N} + G_{F_N}\ddot{q}_{H_N} \end{bmatrix} \tag{4.65}$$

(iv) Collect the inhomogeneous portion as the block-diagonal matrix

$$G = \begin{bmatrix} G_{aF_1} & 0 & 0 & \cdots & 0 \\ 0 & G_{Fa_2} & G_{FF_2} & & 0 \\ 0 & G_{aa_2} & G_{aF_2} & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & G_{Fa_N} \end{bmatrix} \tag{4.66}$$

(v) Compute the system rotation matrix

$$R = \begin{bmatrix} 0 & -R_1^T & \cdots & 0 & 0 \\ R_1 & 0 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 0 & -R_{N-1}^T \\ 0 & 0 & \cdots & R_{N-1} & 0 \end{bmatrix} \tag{4.67}$$

(vi) Solve the boundary condition equation

$$(I - RG)u = RH + \dot{R}H_r \tag{4.68}$$

(vii) Calculate the BC contribution to the component dynamics via

$$\ddot{q}_i = \ddot{q}_{H_i} + \begin{bmatrix} B_{a_i} & B_{F_i} \end{bmatrix} \begin{bmatrix} a_{c_i} \\ F_{f_i} \end{bmatrix} \tag{4.69}$$

with the BC accelerations and forces a_{c_i} F_{f_i} the appropriate elements of the column vector u.

Notes on the solution procedure:

(i) The BC equation matrix $I - RG$ can often be block-multiplied out to reduce storage requirements.

(ii) A symbolic solution to the BC equation in terms of the inverse of $I - RG$ may reduce computation time. However, due to the I/O size of a typical flexible manipulator, a symbolic inversion is not always possible, and will arguably not reduce storage and computation costs.

(iii) The BC equation will, for open chain bodies, be sparse, since each component interacts with only two others. The largest block size on the diagonal will thus be two times the number of interface DOF. The utility of sparse matrix inversion methods is worthy of exploration.

(iv) The computational efficiency of the component modeling procedure depends on the state size of the components versus the number of interfaces and interface DOF. The relative cost is, roughly, inversion of a single mass matrix with dimension equal to the full state size, versus inversion of a number of smaller mass matrices and the boundary condition. If the components are high order (such as a typical flexible manipulator) with a relatively few interfaces, the set of inversions will be performed faster than inversion of a single large matrix (since inversion algorithms generally have cubic dependence on size). This does not take into account sparse matrix inversion algorithms. Since the component model will generally become sparse, while the global mass matrix will remain nearly full, the component model with sparsity may be significantly less computationally intensive.

4.7.2 Control Form

The linear control form is assembled in a highly structured manner:

(i) Linearize the component models.

(ii) Calculate the second order state dynamics as a function of the linearized models.

(iii) Assemble the component matrices into a global, first order state space model.

(iv) Calculate state dependent coupling terms (friction and gearbox stiffness).

(v) Calculate the system influence matrix from commanded torques.

(vi) Determine any system outputs as a function of component states and boundary forces and accelerations.

The linearized model, for a particular joint configuration, is determined in the following way:

(i) Create the joint rotation matrices corresponding to the current configuration.

(ii) Form the system rotation matrix from the joint rotation matrices.

(iii) Calculate the dynamics by an LFT on the system model through the global rotation matrix.

Note that state dependent coupling terms may depend on configuration. In this case, the global rotation matrix must be augmented with the necessary configuration-dependent terms. The state-dependent coupling terms forces will then become a function also of the boundary conditions.

Note also that the system rotation matrix takes the form of a block skew-symmetric matrix. The control design of Chapter 3 requires a block diagonal form. The approach taken in this work to diagonalize the system rotation matrix will be to factor the matrix into a diagonal matrix and a pair of rectangular "pointing" matrices. The pointing matrices are then incorporated into the system model.

The constrained model is given by

$$\ddot{x} = Ax + Bu + B_a a_c + B_F F_c \tag{4.70}$$

$$F_c = C_F x + D_F u + D_{Fa} a_c + D_{FF} F_c \tag{4.71}$$

$$a_f = C_a x + D_a u + D_{aa} a_c + D_{aF} F_c \tag{4.72}$$

with $x^T = \left[\begin{array}{cc} q^T & \dot{q}^T \end{array}\right]$. Using the notation

$$A = -M^{-1}K \tag{4.73}$$

$$B = M^{-1}B_{mf} \tag{4.74}$$

$$B_a = -M^{-1}M_{cf}^T \tag{4.75}$$

$$B_F = M^{-1}B_{ff} \tag{4.76}$$

$$C_F = M_{cf}A \tag{4.77}$$

$$D_F = -B_{mc} + M_{cf}B \tag{4.78}$$

$$D_{Fa} = M_{cc} + M_{cf}B_a \tag{4.79}$$

$$D_{FF} = -B_{cf} + M_{cf}B_f \tag{4.80}$$

$$C_a = G_{af}A \tag{4.81}$$

$$D_a = G_{af}B \tag{4.82}$$

$$D_{aa} = G_{ac} + G_{af}B_a \tag{4.83}$$

$$D_{aF} = G_{af}B_f \tag{4.84}$$

Note that these equations are in second order form. This is for consistency with the nonlinear model. The rate states will be accounted for later, for the aggregate system.

The system EOM are created by appending the component EOM:

$$\ddot{x} = \overset{block}{diag}\, A_i x + \overset{block}{diag}\, B_{u_i} u + \overset{block}{diag}\, B_{p_i} p \tag{4.85}$$

$$q = \overset{block}{diag}\, C_{q_i} x + \overset{block}{diag}\, D_{uq_i} u + \overset{block}{diag}\, D_{qp_i} p \tag{4.86}$$

$$\tag{4.87}$$

where the states, inputs and outputs are the column-wise aggregation of the component variables. The inputs u contain the commanded torques τ as well as the gearbox torques.

The gearbox stiffness forces are calculated from the gearbox stiffness matrix K_g, which relates base, armature, and shaft angles to torques on those components:

$$\begin{bmatrix} m_B \\ m_A \\ m_S \end{bmatrix} = -K_g \begin{bmatrix} \theta_B \\ \theta_A \\ \theta_S \end{bmatrix} \tag{4.88}$$

The gearbox stiffness torques are incorporated into the system matrix by creating the appropriate observation matrix for the base, armature, and shaft angles:

$$\begin{bmatrix} \theta_B \\ \theta_A \\ \theta_S \end{bmatrix} = C_g x \tag{4.89}$$

where x is the global state vector, and by determining the correct global influence matrix to actuate the global DOF:

$$\dot{x} + = B_g \begin{bmatrix} m_B \\ m_A \\ m_S \end{bmatrix} \tag{4.90}$$

The gearbox stiffness is then accounted for in the global state matrices by incorporating the gearbox torques through the observation and influence matrices defined above:

$$\begin{aligned} \dot{x} &= Ax + B_g \begin{bmatrix} m_B \\ m_A \\ m_S \end{bmatrix} && (4.91) \\ &= Ax - B_g K_g C_g x && (4.92) \\ &= (A - B_g K_g C_g) x && (4.93) \\ &= A_g x && (4.94) \end{aligned}$$

Note that the gearbox stiffnesses take on the form of output feedback torques.

Using Equation 5.127 along with the appropriate observation matrix for joint j, C_j, to pull off the correct angles from the system state vector,

$$\begin{bmatrix} \theta_{B_j} \\ \theta_{A_j} \\ \theta_{S_j} \end{bmatrix} = C_j x \tag{4.95}$$

the gearbox torques can be described:

$$\begin{bmatrix} m_{B_j} \\ m_{A_j} \\ m_{S_j} \end{bmatrix} =^{\text{block}}_{\text{diag}} \left(K_{gear_j} C_j\right) x \tag{4.96}$$

The commanded torques impose a differential force between the motor armature and the base:

$$u_j = \begin{bmatrix} .. & 0 & .. \\ -1 & 0 & .. \\ 0 & 1 & .. \end{bmatrix} \tau = T_j \tau_j \tag{4.97}$$

In other words the commanded torque pushes back on the base component of joint j with equal and opposite force to the armature. These "pointing" matrices T_j, which describe how the command influences the states, are used to map the commanded torque (one for each joint), into the component models (where the torques will act on two components). Stacking the pointing matrices,

$$u = T\tau \tag{4.98}$$

gives the component-level torques from the system input torques.

Combining the spring forces and the commanded torque "pointing" matrix, the linearized EOM become:

$$\dot{x} = Ax + B_u u + B_p p \tag{4.99}$$

$$q = C_q x + D_{qu} u + D_{qp} p \tag{4.100}$$

with

$$A = \begin{bmatrix} 0 & I \\ {}^{\text{block}}_{\text{diag}} A_i & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ {}^{\text{block}}_{\text{diag}} B_{u_i} \end{bmatrix} {}^{\text{block}}_{\text{diag}} K_{gear_i} C_i \tag{4.101}$$

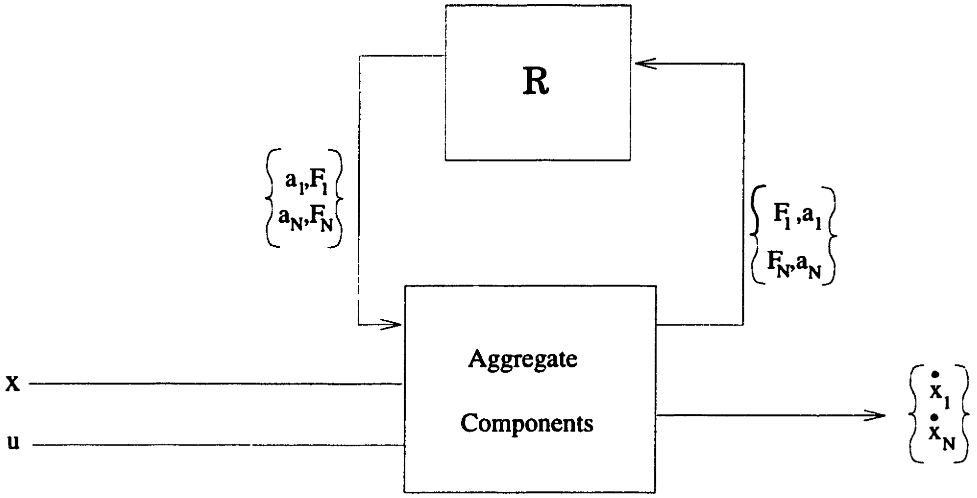

Figure 4-9: Feedback interconnection of the rotation matrices with the linearized system model.

$$B_u = \begin{bmatrix} 0 \\ (\overset{block}{diag} B_{u_i})T) \end{bmatrix} \tag{4.102}$$

$$B_p = \begin{bmatrix} \overset{block}{diag} B_{p_i} \end{bmatrix} \tag{4.103}$$

$$C_q = \begin{bmatrix} \overset{block}{diag} C_{q_i} \end{bmatrix} \tag{4.104}$$

$$D_{qu} = \overset{block}{diag} D_{uq_i} \tag{4.105}$$

$$D_{qp} = \overset{block}{diag} D_{qp_i} \tag{4.106}$$

with the geometric terms (p and q) calculated as

$$p = Rq \tag{4.107}$$

Note that that in effect the linearized system model consists of the aggregate of the linearized component models. That is, they are no longer independent. This is represented in the Figure 4-9

4.7.3 Friction

The resulting manipulator model does not incorporate friction. The scope of the thesis is limited to an accurate and simple representation of configuration-dependent nonlinearities. Friction takes the form of a (possibly nonlinear) function of the joint rates and is not configuration dependent, and as such does noι fall within the bounds of the research. However, this is not to suggest that friction is not of concern to the manipulator controls engineer. In fact, friction may end up being a strong determinant of the ability to close high-authority loops around a robotic system. Thus it is important be aware of the standard *ad hoc* methods for accounting for friction which are available in the literature. Friction can affect two areas: the ability to *model* the system, and the ability to perform *control.*

The ability to model the system may be compromised due to friction. This is partly due to the difficulty in accurately capturing the effects of friction; witness the literature extant on determining linear friction. In addition, the presence of nonlinear friction may distort the transfer function or time domain data needed for identification and model updating to such an extent that the model parameters cannot be determined. Of particular interest are *Coulomb* friction and stiction.

Coulomb friction refers to a friction force which has a constant magnitude, with a sense that resists the motion of the joint:

$$F_c = -c\ sign(\dot{\theta}) \tag{4.108}$$

where c is the magnitude of the friction force and $\dot{\theta}$ is the joint rotation rate. Stiction takes the form of a static friction force which "freezes" the joint in place whenever the joint rate goes to zero and the inertial force is not sufficient to break the joint loose.

Modeling of stiction is currently not well understood, since its effect is to constrain, and then release, a subset of the EOM. In effect the model order varies dynamically. One of the few approaches uses a multiple-time-scale simulation approach to resolve high bandwidth (step function) friction forces, then solve for lower bandwidth vibration and rigid body motions. The work does not address experimental determination of the stiction parameter.

It is assumed that the above model is a representation of a physical system. Further, in keeping with the scope of the thesis, the model must be of sufficient accuracy to allow the use of control. In general, as asserted above, friction will be a strong element of the system dynamics. Thus a friction model may be required; this may further entail experimental determination of the friction model parameters.

Many approaches to identifying the Coulomb friction parameter have been proposed, in the time domain[2] as well as in the frequency domain[79]. The time domain technique involves a pendulum test in which the envelope of the time response is fitted with a friction parameter, using a numerical cost evaluation and finite difference search function. Alternatively, for higher computational cost, a time domain model simulation in response to various inputs can be performed, with a similar numerical cost function created from time domain simulations of the model with friction imposed at the joints and data from the actual system. The frequency domain approach uses transfer function measurements of joint commanded torque and motor angle, along with the first term in the Fourier series of the square wave function $sign(\dot{\theta})$, to derive a numerical cost function representing the error between the modeled and actual friction forces. The parameter is then further used to modify the experimentally determined transfer function, to reduce the distortion caused by friction in the measurement.

Four approaches for implementing control in the presence of nonlinear friction are commonly used. A fifth may be considered, in addition, although it has apparently not been implemented to date:

(i) dither: high frequency torque command which breaks the joint free, while being far enough above the fundamental mode that the structural response is low.

(ii) feedforward of a torque which opposes friction (based on the identified model above).

(iii) joint position servoing, using the property that feedback rejects disturbance forces.

(iv) joint torque servoing, in which strain measurements on the base of the link (which are proportional to moment in the link) are used to servo a commanded torque into the link.

(v) nonlinear feedback of joint rate, via a nonlinear delta block similar to that used for the configuration dependence.

The first option may excite structural modes in the rolloff region, and may thus be unacceptable since it is an additional disturbance source. In addition, the step nature of the friction response to the dither command may excite frequencies beyond the sampling frequency of the active control[49], with resulting aliasing.

The major difficulty of numbers two and five is that an accurate model of the friction is needed. If the feedforward torque is larger than the friction force it opposes, the feedforward torque will be destabilizing (as it is a function of the joint angular rate, thus in fact is a positive feedback term). This is problematic because the magnitude of the friction can vary over a wide range as a function of joint angle, and in fact is not generally repeatable for any given angle (this is due to the bearings not moving in a repeatable way in the ball race).

In addition, feedforward only works for a joint which is being commanded. For a linear system, a torque in for example any pitch joint, will cause some motion at all the other pitch joints, due to being communicated through the structure. This will not occur when the torque carried by the structure is less than the breaking torque of the stiction in the other joints. If the input torque is always less than the breaking torque at the other joints, the robot will respond as if it is a "locked joint" structure.

The joint servoing techniques, options three and four, both rely on the well-known property that feedback rejects disturbances and decrease the effect of model errors (thus obviating the need for an accurate friction model). Difficulty may arise in getting sufficiently high gain, in the presence of strong friction force (or slow commanded movements, in which inertial forces are small compared to friction). Servo bandwidth is limited by loop phase loss, due to motor dynamics time delay. Digital control implementations are particularly of concern. Motor dynamics may appear when the torsional rotation of the motor shaft becomes important. For the torque servo, there is geartrain (or belt reduction) flexibility between the motor input and the sensed torque output. This takes the form of a second order roll-off and is amenable to rate compensation. However, the rolloff frequency changes as the manipulator inertia changes with configuration. Torque servoing is often fairly simple to implement in analog, as the motor is often driven by a voltage controlled current amp. Adding a loop which modulates the voltage input to the amp based on sensed beam moment error is straightforward to do in analog, which bypasses the potential bandwidth restrictions of a digital loop.

The first four approaches may be implemented independently of the high authority control. The final approach may be amenable to the type of control proposed in the following section for the configuration dependent terms in the dynamics. This would require a measurement of the friction force. Alternatively a model-based control approach may be possible in the same framework. This is a topic which requires further research and will not be developed in this thesis.

Chapter 5

Component Modeling

The proposed manipulator modeling approach relies on an existing set of component models. The accuracy of the system model is set by the accuracy of the component models. The component boundary conditions preserve all forces and accelerations across the interconnection, so do not degrade accuracy. Thus, all "important" terms in the EOM must be present, where the importance is a function of the system being modeled, the expected rates of motion, and so on, and must remain in the province of engineering insight. In addition, the representation of the component models can facilitate (or hinder) the system modeling algorithm. In this chapter, the modeling of two important flexible manipulator components is considered: the link, and the gearbox. These two components will be sufficient to capture many flexible manipulators, such as the SRMS. A Lagrange-based derivation will be used. As will be seen, the approach leads to a closed form model for the link. This allows the system model to be written in closed form. That is, an open chain manipulator (of any number of links) can be modeled using the results of the previous chapter combined with this chapter. In addition, the EOM are in a form which requires minimal on-line calculation.

The chapter is laid out is follows: in Section 5.2, a set of generalized coordinates, which leads to a simplified form for the link EOM, is proposed. The proposed coordinates are used to derive the EOM for the link component. Next the gearbox component EOM are derived.

5.1 Modeling Framework

The Lagrangian framework is used to derive the free-free EOM of the link and gearbox components. Lagrangian methods employ the system energy function, or Lagrangian. The energy function is written in terms of a set of assumed degrees of freedom (DOF) collected in the vector q:

$$L(q,\dot{q}) = T(q,\dot{q}) - P(q,\dot{q}) \tag{5.1}$$

where L is the Lagrangian, T is the kinetic energy, and P is the potential energy. Then Lagrange's equations, given by

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = BF \tag{5.2}$$

are evaluated to determine the equations of motion (EOM). The matrix B is the *influence matrix* which maps external forces F to the states.

The general form of the resulting model will be

$$\begin{aligned} M(q)\ddot{q} &= -V(q,\dot{q}) + B(q)F \\ a &= H(q,\dot{q}) + G(q)F \end{aligned} \tag{5.3}$$

where as before, M is a time-varying inertia matrix, V are state dependent forces, and B is the influence matrix (including boundary forces). a are the boundary accelerations, which are functions of the states H and the boundary forces through G. Component models of the above form will be sought for the link and the gearbox.

The local modeling result of Section 5.2 will be used in the derivation of the link model. Some further notational conventions will be made to facilitate the process. First, note that the Lagrangian is linear in the energy functions T and P, thus Lagrange's' equation can be evaluated in terms of a summation of contributions from various sources:

$$T = \sum_{i=0}^{N} T_i, \; P = \sum_{i=0}^{N} P_i \tag{5.4}$$

Multiple contributions can exist; for example, kinetic energy due to translational velocity, and due to angular velocity. Noting that Lagrange's equation is a function of time only, not space, the summation above can be extended to an integral over a body (conceptually, as an infinite sum of differential elements)

$$T_i = \int_0^L T_i' \mathrm{d}r \tag{5.5}$$

where r is a length parameter, the prime denotes differentiation with respect to r, and the differential element is $\mathrm{d}r$. The length parameter could in fact be a vector of lengths, and the integral be a body integral. For notational convenience, the body will be considered to be "long and thin", so that the integration is carried out over a scalar length parameter (e.g. the distance along the link). Note that T_i' is then an energy per unit length.

Note that the EOM which result from evaluating Lagrange's equation for the body can be written

$$M\ddot{q} + V = BF \tag{5.6}$$

where M is the *mass matrix*, V is the vector of *state-dependent forces*, and again B is the influence matrix for the external forces F. Note that if the Lagrangian is broken out into contributions T_i, each T_i will result in corresponding M_i and V_i terms in the EOM. Taking advantage of this fact, the evaluation of Lagrange's equation will proceed as follows: determine the contributions T_i and P_i, apply Lagrange's equations to each contribution, integrate over the link, and sum the resulting mass matrices and state dependent forces:

$$M = \sum_{i=0}^{N} \int_0^L M_i' \mathrm{d}r \tag{5.7}$$

$$V = \sum_{j=0}^{N} \int_0^L V_i' \mathrm{d}r \tag{5.8}$$

Using the results of Section 5.2, the components M_i' and V_i' will fall out of the definition of the link component.

5.2 Local Equations of Motion

One of the keys to a successful model for a nonlinear structure is the correct choice of assumed DOF. It is the aim of this section to present a particular set of DOF

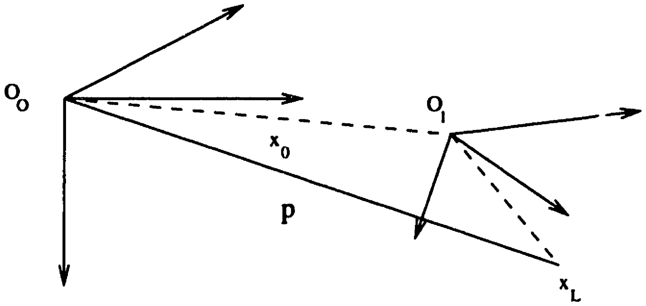

Figure 5-1: The position of the particle is represented in the frame O_1, which is moving with respect to the inertial frame O_0.

which may be said to be "natural" for a flexible link. The argument will be made by using the proposed DOF to derive the EOM for a particular system. The resulting EOM will be compared to a more standard representation to determine if there is an advantage to the former.

The present section discusses the generation of the EOM of a single point. The point may represent a true point mass, or a differential mass which can be integrated over the body to produce the body EOM. The initial step is to define an inertial reference frame O_0, and a moving coordinate frame O_1. O_1 can translate and rotate in the inertial frame. The point of interest can in turn translate and rotate within O_1. O_1 will be referred to as the *local frame.* Vector quantities will be referred to as *local coordinates* when they are expressed in the local frame.

In order to use Lagrange's equation, the inertial velocity of the point is required. Let the velocity of the point in inertial space be denoted $\dot{p}$, which can be written

$$\dot{p} = \dot{x}_0 + R\dot{x}_L \tag{5.9}$$

where $\dot{x}_0$ is the velocity of the local frame, and $\dot{x}_L$ is the velocity of the point *expressed in the local frame*, that is, in local coordinates. R is the rotation from local to global coordinates. Let the assumed DOF which represent motion in the local frame be denoted q_L. The DOF q_L for a manipulator will typically consist of rotations θ and translations w. Partitioning for convenience,

$$q_L = \begin{bmatrix} \theta \\ w \end{bmatrix} \tag{5.10}$$

Assume that a linear relationship between the assumed DOF and the local velocity exists, which is independent of rotations:

$$\dot{x}_L = A(w)\dot{q}_L \tag{5.11}$$

An expression for the EOM in terms of the local velocities $\dot{q}_L$, and the frame velocity $\dot{x}_0$, can be found from Lagrange's equation.

The energy per unit mass of the point is given by

$$\begin{aligned} T' &= \frac{1}{2}\dot{p}^T\dot{p} && (5.12)\\ &= \frac{1}{2}(\dot{x}_0 + R\dot{x}_L)^T(\dot{x}_0 + R\dot{x}_L) && (5.13)\\ &= \frac{1}{2}(\dot{x}_0 + RA\dot{q}_L)^T(\dot{x}_0 + RA\dot{q}_L) && (5.14)\\ &= \frac{1}{2}\begin{bmatrix} \dot{x}_0 \\ \dot{q}_L \end{bmatrix}^T \begin{bmatrix} I & RA \\ A^TR^T & A^TA \end{bmatrix} \begin{bmatrix} \dot{x}_0 \\ \dot{q}_L \end{bmatrix} && (5.15) \end{aligned}$$

Applying Equation 5.2 and using the notation $q = \begin{bmatrix} x_0^T & q_L^T \end{bmatrix}^T$,

$$\begin{aligned} \frac{d}{dt}\frac{\partial T'}{\partial \dot{q}} &= \begin{bmatrix} I & RA \\ A^TR^T & A^TA \end{bmatrix} \begin{bmatrix} \ddot{x}_0 \\ \ddot{q}_L \end{bmatrix} + \\ & \begin{bmatrix} 0 & \dot{R}A + R\dot{A} \\ (\dot{R}A + R\dot{A})^T & \dot{A}^TA + A^T\dot{A} \end{bmatrix} \begin{bmatrix} \dot{x}_0 \\ \dot{q}_L \end{bmatrix} && (5.16)\\ \frac{\partial T'}{\partial \theta_i} &= \frac{1}{2}\begin{bmatrix} \dot{x}_0 \\ \dot{q}_L \end{bmatrix}^T \begin{bmatrix} 0 & R_i'A \\ (R_i'A)^T & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_0 \\ \dot{q}_L \end{bmatrix} && (5.17)\\ \frac{\partial T'}{\partial w_i} &= \frac{1}{2}\begin{bmatrix} \dot{x}_0 \\ \dot{q}_L \end{bmatrix}^T \begin{bmatrix} 0 & RA_i' \\ (RA_i')^T & A_i'^TA + A^TA_i'^T \end{bmatrix} \begin{bmatrix} \dot{x}_0 \\ \dot{q}_L \end{bmatrix} && (5.18) \end{aligned}$$

where the notation $R_i' = \frac{\partial R}{\partial \theta_i}$, θ_i is the i^{th} component of θ, and similarly $A_i' = \frac{\partial A}{\partial w_i}$. These equations are a force balance expressed in the inertial frame. They are functions of orientation, through the rotation matrix R. The dynamics are thus functions of the orientation of the point. This is intuitively over-complicated. A simpler expression is desired, in which the dynamics are independent of orientation.

The EOM can be expressed in the local frame using the substitutions for the frame velocities and accelerations

$$R\dot{q}_0 = \dot{x}_0, R\ddot{q}_0 = \ddot{x}_0 \tag{5.19}$$

This is not a state transformation. Rather, the given vector quantities are being expressed in the local frame. The force balance corresponding to the inertial translation DOF must also be rotated into the local frame, by pre-multiplying by R^T. The EOM become

$$\begin{aligned}\frac{d}{dt}\frac{\partial T'}{\partial \dot{q}} = & \begin{bmatrix} I & A \\ A^T & A^TA \end{bmatrix} \begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_L \end{bmatrix} + \\ & \begin{bmatrix} 0 & R^T\dot{R}A + \dot{A} \\ (R^T\dot{R}A + \dot{A})^T & \dot{A}^TA + A^T\dot{A} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix}\end{aligned} \tag{5.20}$$

$$\frac{\partial T'}{\partial \theta_i} = \frac{1}{2} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix}^T \begin{bmatrix} 0 & R^TR_i'A \\ (R^TR_i'A)^T & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix} \tag{5.21}$$

$$\frac{\partial T'}{\partial w_i} = \frac{1}{2} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix}^T \begin{bmatrix} 0 & A_i' \\ A_i'^T & A_i'^TA + A^TA_i'^T \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix} \tag{5.22}$$

Equations 5.20-5.22 are not of interest in themselves, since the rotation matrix, and its derivatives in t and θ, still appear. However, the following relationship can be used to express the rotation matrix terms in Equation 5.20 in terms of the angular rates $\dot{\theta}$ only:

$$R^T\dot{R}v = \dot{\theta} \times v = \begin{bmatrix} 0 & -\dot{\theta}_z & \dot{\theta}_y \\ \dot{\theta}_z & 0 & -\dot{\theta}_x \\ -\dot{\theta}_y & \dot{\theta}_x & 0 \end{bmatrix} v = \Omega v \tag{5.23}$$

where the elements of Ω are the angular rates of the local frame, expressed in the local frame. Further, the following expansion for Ω can be written:

$$\Omega = R^T\dot{R} = R^T(R_x'\dot{\theta}_x + R_y'\dot{\theta}_y + R_z'\dot{\theta}_z) \tag{5.24}$$

from which the following relation can be determined

$$\Omega_i' = \frac{\partial \Omega}{\partial \theta_i} = R^TR_i' \tag{5.25}$$

which is a constant matrix, consisting of zeros and ones in the appropriate locations. Using these relationships, Equations 5.20-5.22 can be re-written in a form which is independent of orientation:

$$\frac{d}{dt}\frac{\partial T'}{\partial \dot{q}} = \begin{bmatrix} I & A \\ A^T & A^T A \end{bmatrix} \begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_L \end{bmatrix} + \begin{bmatrix} 0 & \Omega A + \dot{A} \\ (\Omega A + \dot{A})^T & \dot{A}^T A + A^T \dot{A} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix} \tag{5.26}$$

$$\frac{\partial T'}{\partial \theta_i} = \frac{1}{2} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix}^T \begin{bmatrix} 0 & \Omega_i' A \\ (\Omega_i' A)^T & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix} \tag{5.27}$$

$$\frac{\partial T'}{\partial w_i} = \frac{1}{2} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix}^T \begin{bmatrix} 0 & A_i' \\ A_i'^T & A_i'^T A + A^T A_i'^T \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix} \tag{5.28}$$

These equations represent the EOM of a particle of unit mass, expressed in the local frame. The rotation to ground, R, does not appear. The implications are that the dynamics are independent of the orientation of the frame. This is intuitively pleasing. A link freely floating in space should have the same dynamics whether it is vertical with respect to the observer, or horizontal. Note that the angular DOF are specified by Equation 5.23. In particular, the rotation rates are referred to a rotating coordinate system. As a result the rotation DOF will not in general be physically meaningful. In many instances this will not matter. In some cases (such as in a gravity field) it may. In this case the inertial orientation is determined as the integral of Equation 5.19. In addition, the relation in Equation 5.11 must hold, so that the mapping $A(w)$ from DOF to inertial velocity does not depend on orientation.

5.3 Link Modeling

Links are modeled as thin, flexible beam elements (Figure 5-2). Transverse and torsional vibrations are supported. Axial stiffness is assumed high enough to neglect axial deformation. A Bernoulli-Euler beam model is used to represent the transverse link stiffness and mass, based on the assumption of thin cross-section. The parameters

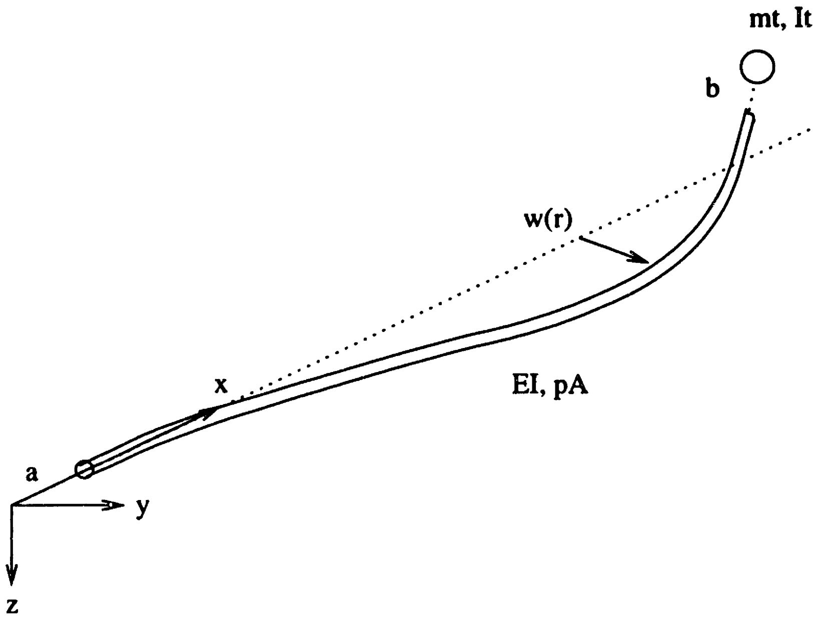

Figure 5-2: Link component overview.

describing the link are length L, bending stiffness EI, torsional stiffness GJ, linear density ρA, and cross-sectional inertias of I_{xp}, I_{yp}, and I_{zp}. The flexible portion of the link is offset from the axis of rotation by a distance a (which may represent the distance from the motor axis to the gearbox case and the link attachment fixture). The link is loaded with an end mass of mass m_t and inertias I_{x_t}, I_{y_t}, and I_{z_t}. The end mass is offset from the end of the bending portion of the link by a vector of offsets $b = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}^T$. The length L refers to the bending length of the link. Thus the total distance from the base of the link to the center of mass of the end mass is $L_t = a + L + b_x$.

In accordance with the local modeling results, the link model states are defined as follows: the link frame of reference is affixed to one end of the beam, referred to as the *base* (Figure 5-3). The link frame translates and rotates with the link, in three dimensions. The x axis of the frame points along the link. The x position variable is r. The translation of the base of the link is captured with the DOF $\begin{bmatrix} x(t) & y(t) & z(t) \end{bmatrix}^T$, expressed in the rotating link frame. The rotation of the link frame (thus the base of the link) is expressed with the DOF $\begin{bmatrix} \theta_x(t) & \theta_y(t) & \theta_z(t) \end{bmatrix}^T$,

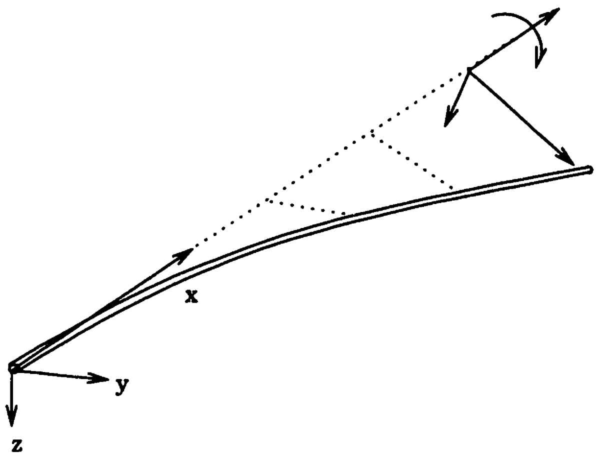

Figure 5-3: Link axis system, affixed to base of link. Deflections w are defined in a coordinate system which rotates about the local x axis due to torsion in the link.

again referenced to the link frame. The bending deflection of the beam is denoted $w = \left[\begin{array}{ccc} w_x(r,t) & w_y(r,t) & w_z(r,t) \end{array}\right]^T$, where $w_x(r,t)$ is in fact a torsional vibration variable, and $w_y(r,t)$ and $w_z(r,t)$ are transverse bending deflections. These can be thought of as place-holders for an assumed modes expansion. Dependence on r and t will be suppressed. Following the notation in the local modeling section, the translational variables are denoted q_0, the rotations as θ, and the local variables as $q_L = \left[\begin{array}{cc} \theta^T & w^T \end{array}\right]^T$.

The link modeling algorithm is to 1. determine the differential mass matrices and state-dependent forces by decomposing the energy terms, and evaluate Lagrange's equation for each, 2. choose a set of assumed modes and integrate over the links, 3. sum the contributions, 4. evaluate the force influence matrix, and 5. determine the boundary accelerations as functions of the model DOF. The link kinetic energy will be decomposed into translational and rotational contributions. The link potential energy will be evaluated due to centripetal forces, and due to material stiffness.

5.3.1 Link Kinetic Energy Terms

The link kinetic energy exists in the form of of transverse and rotational velocity of the differential elements. Denoting these contributions as 1, and 2, the link kinetic energy is

$$T = \int_0^L T_1' \mathrm{d}r + \int_0^L T_2' \mathrm{d}r \tag{5.29}$$

which leads to the following terms in the component dynamics:

$$\int_0^L M_1' \mathrm{d}r, \int_0^L M_2' \mathrm{d}r \tag{5.30}$$

and

$$\int_0^L V_1' \mathrm{d}r, \int_0^L V_2' \mathrm{d}r \tag{5.31}$$

Transverse motion is considered first. A relation between the local variables and the inertial velocity is sought. Using the notation $c_x = \cos(w_x)$, $s_x = \sin(w_x)$ leads to the expression

$$x_L = \begin{bmatrix} r \\ c_x w_y - s_x w_z \\ c_x w_z + s_x w_y \end{bmatrix} = \begin{bmatrix} r \\ \hat{w}_y \\ \hat{w}_z \end{bmatrix} \tag{5.32}$$

where the variables $\hat{w}_y$ and $\hat{w}_z$ are the deformations projected onto the base. The total time derivative of Equation 5.32 in the rotating coordinate frame is found from $d/dt(.) = \frac{\partial}{\partial t}(.) + \omega \times (.)$:

$$\dot{x}_L = \begin{bmatrix} 0 \\ \dot{\hat{w}}_y \\ \dot{\hat{w}}_z \end{bmatrix} + \begin{bmatrix} 0 & -\dot{\theta}_z & \dot{\theta}_y \\ \dot{\theta}_z & 0 & -\dot{\theta}_x \\ -\dot{\theta}_y & \dot{\theta}_x & 0 \end{bmatrix} \begin{bmatrix} r \\ \hat{w}_y \\ \hat{w}_z \end{bmatrix} \tag{5.33}$$

Collecting the rate dependent terms,

$$\dot{x}_L = \begin{bmatrix} 0 & \hat{w}_z & -\hat{w}_y & 0 & 0 & 0 \\ -\hat{w}_z & 0 & r & -\hat{w}_z & c_x & -s_x \\ \hat{w}_y & -r & 0 & \hat{w}_y & s_x & c_x \end{bmatrix} \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \\ \dot{w}_x \\ \dot{w}_y \\ \dot{w}_z \end{bmatrix} \tag{5.34}$$

thus A is determined. Calculating A, $A^T A$, $A_x = \frac{\partial A}{\partial w_x}$, $A_y = \frac{\partial A}{\partial w_y}$, and $A_z = \frac{\partial A}{\partial w_z}$, and their time derivatives, the differential mass matrix M_1 and state dependent forces V_1

are known from Equations 5.26-5.28:

$$M_1' = \begin{bmatrix} I & A \\ A^T & A^T A \end{bmatrix} \tag{5.35}$$

$$V_1' = \begin{bmatrix} 0 & \Omega A + \dot{A} \\ (\Omega A + \dot{A})^T & \dot{A}^T A + A^T \dot{A} \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_L \end{bmatrix} \tag{5.36}$$

These will be high order in the deflection variables (up to 6^{th} order) with trigonometric terms representing rotations. Small angle approximations for the deflection variables will be used throughout this work:

$$\sin(x) = x \ , \ \cos(x) = 1 \tag{5.37}$$

$$\sin(x)^2 = x^2 \ , \ \cos(x) = 1 - x^2 \tag{5.38}$$

which preserves the relationship $\sin(x)^2 + \cos(x)^2 = 1$. Note that the full equations for the link EOM will not be presented in this chapter, for reasons of clarity and brevity. The full EOM, to second order in deflection variables, can be found in Appendix B.

Next the energy in rotation of the differential element is evaluated. Note that the magnitude of the energy in rotation may be much smaller than that in transverse motion, and may be neglected if appropriate. Care must be taken to include at least the torsional rotational energy (about the x axis), otherwise the torsional mass matrix will be zero and the component mass matrix will be non-positive definite. An expression for velocity as a function of local variables is defined, similar to the translational velocity:

$$\dot{\Theta} = \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix} \begin{bmatrix} \dot{w}_x \\ -\dot{w}_z' \\ \dot{w}_y' \end{bmatrix} \tag{5.39}$$

where again $\dot{w}_x$ is torsion rate (already an angular rate). $\dot{w}_y'$ and $\dot{w}_z'$ are differential rotations associated with transverse deflections. Using the notation $w' = \begin{bmatrix} w_x & w_y' & w_z' \end{bmatrix}^T$ and by extension $q_L' = \begin{bmatrix} \theta^T & w'^T \end{bmatrix}^T$, $q' = \begin{bmatrix} q_0^T & q_L'^T \end{bmatrix}^T$, the rotation rate can be written

$$\dot{\Theta} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -s_x & -c_x \\ 0 & 0 & 1 & 0 & c_x & -s_x \end{bmatrix} \dot{q}_L' \tag{5.40}$$

$$= \begin{bmatrix} 0 & A_r \end{bmatrix} \dot{q}' \tag{5.41}$$

where I_r is the matrix of cross-sectional rotational inertias (akin to the linear density ρA):

$$I_r = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{5.42}$$

It is noted in passing that often the inertia coupling terms are zero and I_r is diagonal (this arises when the neutral bending axis coincides with the modulus-weighted centroid of the link). The kinetic energy associated with rotation of the differential element is

$$T_2' = \int_0^L \dot{q}'^T \begin{bmatrix} 0 & A_r \end{bmatrix}^T I_r \begin{bmatrix} 0 & A_r \end{bmatrix} \dot{q}' \mathrm{d}r \tag{5.43}$$

Lagrange's equation applied to Equation 5.43 gives

$$\frac{d}{dt}\frac{\partial T_2'}{\partial \dot{q}_L'} = A_r^T I_r A_r \ddot{q}_L' + (A_r^T \dot{I}_r A_r)\dot{q}_L' \tag{5.44}$$

$$\frac{\partial T_2'}{\partial w_x} = \dot{q}_L'^T \frac{\partial A_r^T I_r A_r}{\partial w_x} \dot{q}_L' \tag{5.45}$$

By inspection,

$$M_2 = \begin{bmatrix} 0 & 0 \\ 0 & A_r^T I_r A_r \end{bmatrix} \tag{5.46}$$

and $V_2 = (A_r^T \dot{I}_r A_r)\dot{q}_L$ plus a contribution in the EOM of the torsional deflection w_x.

5.3.2 Link Potential Energy Terms

The link potential energy includes link bending stiffness and centrifugal stiffening. Again the energy has two components

$$P = \int_0^L P_1' \mathrm{d}r + \int_0^L P_2' \mathrm{d}r \tag{5.47}$$

which lead to the following terms in the link dynamics

$$\int_0^L V_3'\mathrm{d}r, \int_0^L V_4'\mathrm{d}r \tag{5.48}$$

note that the indices start at 3 to align with the kinetic energy contributions.

Centrifugal stiffening is calculated using the work done to move the differential element away from the centerline against the tension exerted by centrifugal force $T(r)$. The distance the force is applied over is equal to the length of the displaced element, ds, minus the length of the undisplaced element, dr:

$$P_1' = T(r)(ds - dr) \tag{5.49}$$

The "stretched" distance ds is related to the distance along the x-axis by the differential rotation (due to bending) of the link:

$$ds^2 = dr^2 + (w'dr)^2 \tag{5.50}$$

or

$$ds = \sqrt{1 + w'^2}dr \tag{5.51}$$

Using the approximation

$$\sqrt{1 + w'^2}dr \approx (1 + \frac{1}{2}w'^2)dr \tag{5.52}$$

allows the distance term in the work equation to be written as a quadratic term:

$$ds - dr \approx (1 + \frac{1}{2}w'^2)dr - dr = \frac{1}{2}w'^2dr \tag{5.53}$$

With two transverse bending axes, w' is the hypotenuse of a triangle with w_y' and w_z' on the sides. By Pythagoras' Theorem,

$$ds - dr \approx \frac{1}{2}(w_y'^2 + w_z'^2)dr \tag{5.54}$$

which is a quadratic term in the deflection DOF.

Tension is created by the radial acceleration, a_r, of the sections of the link outboard of the differential element:

$$T(r) = \rho A \int_r^{L+a} a_r(r)\mathrm{d}r + m_t a_r(L_t)L_t \tag{5.55}$$

where $L_t = a + L + b$ is the distance from the joint axis to the center of mass of the tip mass. Radial acceleration can be found to be (zeroing deflection terms to avoid carrying two spatial integrations)

$$a_r = \ddot{x} - (\dot{\theta}_y^2 + \dot{\theta}_z^2)r \tag{5.56}$$

which leads to the following expression for the tension in the link:

$$T(r) = (\rho A(a + L - r) + m_t L_t)\ddot{x} + (\frac{1}{2}\rho A(r^2 - (a + L)^2) - m_t L_t)(\dot{\theta}_y^2 + \dot{\theta}_z^2) \tag{5.57}$$

Substituting Equation 5.57 and Equation 5.54 into Equation 5.49, and adding in the link stiffness terms, produces the following expression for the link potential energy.

$$\begin{aligned} P_1' = {} & \frac{1}{2}(m_t + \rho A(a + L - r))\ddot{x}(w_y'^2 + w_z'^2) - \\ & (m_t L_t + \frac{1}{2}\rho A((a + L)^2 - r^2))(\dot{\theta}_y^2 + \dot{\theta}_z^2)(w_y'^2 + w_z'^2) \end{aligned} \tag{5.58}$$

With a symmetric cross-section of bending stiffness EI and torsional stiffness GJ. Collecting terms, applying Lagrange's Equations, and linearizing the result in the deflection terms,

$$\frac{\partial P_1'}{\partial w_x} = 0 \tag{5.59}$$

$$\frac{\partial P_1'}{\partial w_y} = \frac{1}{2}\left((m_t + \rho A(a + L))\ddot{x} - \frac{1}{4}(m_t L_t + \rho A(a + L)^2)(\dot{\theta}_y^2 + \dot{\theta}_z^2)\right) w_y' \tag{5.60}$$

$$- \rho A r w_y' \ddot{x} + \frac{1}{4}\rho A(\dot{\theta}_y^2 + \dot{\theta}_z^2)r^2 w_y{}' \tag{5.61}$$

$$\frac{\partial P_1'}{\partial w_z} = \frac{1}{2}\left((m_t + \rho A(a + L))\ddot{x} - \frac{1}{4}(m_t L_t + \rho A(a + L)^2)(\dot{\theta}_y^2 + \dot{\theta}_z^2)\right) w_z' \tag{5.62}$$

$$- \rho A r w_z' \ddot{x} + \frac{1}{4}\rho A(\dot{\theta}_y^2 + \dot{\theta}_z^2)r^2 w_z{}' \tag{5.63}$$

The energy in bending simply consists of the strain energy in the link:

$$P_2' = EI(w_y''^2 + w_z''^2) + GJw_x'^2 \tag{5.64}$$

which leads to

$$\frac{\partial P_2'}{\partial w_x} = GJw_x'^2 \tag{5.65}$$

$$\frac{\partial P_1'}{\partial w_y} = EIw_y'' \tag{5.66}$$

$$\frac{\partial P_1'}{\partial w_z} = EIw_z'' \tag{5.67}$$

these terms all contribute to the state-dependent force on the differential element. There is no inertia matrix contribution from the potential energy.

The ı esult of the evaluation of the decomposed energy functionals is the unforced dynamics of the differential element on the link:

$$M'(q)\ddot{q} + V'(q,\dot{q}) = 0 \tag{5.68}$$

where

$$M' = M_1 + M_2 \tag{5.69}$$

and

$$V' = V_1 + V_2 + V_3 + V_4 \tag{5.70}$$

The next section will present the evaluation of the link unforced dynamics from the dynamics of the differential element.

5.3.3 Assumed Modes and Spatial Integration

As of this point the deflections are in the form of spatially distributed functions $w(r,t) = \begin{bmatrix} w_x(r,t) & w_y(r,t) & w_z(r,t) \end{bmatrix}^T$. A finite order model is desired. An assumed modes approach is used to approximate the flexible deformations. The assumed modes approach factors the deflection into a set of functions of time only and a set of functions of space only:

$$w_x(r,t) = \phi_x(r)^T q_x(t) \tag{5.71}$$

$$w_y(r,t) = \phi_y(r)^T q_y(t) \tag{5.72}$$

$$w_z(r,t) = \phi_z(r)^T q_z(t) \tag{5.73}$$

The assumed modes carry through the application of Lagrange's Equation by the application of the chain rule:

$$\frac{\partial T'}{\partial q_i} = \frac{\partial T'}{\partial w_i}\frac{\partial w_i}{\partial q_i} = \frac{\partial T'}{\partial w_i}\phi_i, \tag{5.74}$$

$$\frac{\partial T'}{\partial \dot{q}_i} = \frac{\partial T'}{\partial \dot{w}_i}\phi_i \tag{5.75}$$

Note that q_i and ϕ_i are vector quantities, while w_i are scalars. The transformation between the EOM in terms of the distributed functions and the assumed modes expansion will involve an increase in order.

The assumed modes expansion can be written as a state transformation on the distributed deflection coordinates:

$$\begin{bmatrix} x \\ y \\ z \\ \theta_z \\ \theta_y \\ \theta_z \\ w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \phi_x^T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi_y^T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \phi_z^T \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \theta_z \\ \theta_y \\ \theta_z \\ q_x \\ q_y \\ q_z \end{bmatrix} \tag{5.76}$$

For the purposes of exposition, the above transformation is defined as the matrix T and used to transform the system state vector. The mass matrix becomes

$$\hat{M} = T^T M T \tag{5.77}$$

the state dependent forces become

$$\hat{V} = T^T V \tag{5.78}$$

the force influence matrix becomes

$$\hat{B} = T^T B \tag{5.79}$$

and the acceleration output function becomes

$$\hat{G} = GT \tag{5.80}$$

Note that the above terminology will not be used in the sequel. The state transformation from distributed coordinates to assumed modal coordinates will be implicit.

Note also that the transformation is rank-deficient. That is, there are more columns than rows. As a result the mass matrix is apparently singular. This is true because for a single point at a location r on the link, the deflection w_y is the same as base motion y. However, the mass matrix is actually full-rank because of the spatial distribution of the deflection mode shapes. Essentially the rigid body shape is a constant deflection along the link, and the flexible shapes can be decomposed into a complete orthogonal set (to each other and the rigid body modes). Hence when the differential mass matrix is integrated over the link (or in fact evaluated at any two points with non-zero mass) the mass matrix will be full rank.

Choice of the mode shapes $\phi_i(r)$ is an important step in creating an accurate model. Oakley[2] has analytically determined transverse vibration mode shapes for a cantilevered BE beam with a tip mass and inertia. She experimentally verifies that the mass-loaded mode shapes produce a more accurate model than unloaded cantilevered beams, which need to capture a non-zero strain distribution at the tip with a zero strain mode shape. Thus accuracy is greater for the same number of mode shapes. Alternatively, fewer modes need be used for the same accuracy. The same mode shapes can be determined numerically (see Appendix A).

Note that the mass loaded mode shape does not obviate the need for engineering insight into the mode shape selection. The correct system loading varies as a function of configuration (as the attached inertia varies). A reasonable average mass must be chosen to approximate the true system throughout the workspace. Such an average may need to account for varying payloads, and should also take into account the particular structure. A "floppy" manipulator may not completely transfer configuration varying moments back to the link, leading to a lower apparent load than simply the total mass attached outboard of that link.

The choice of a cantilevered base is also an approximation. The link will generally be attached to a compliant base, thus $w(0,t)$ and $w'(0,t)$ will not be zero. A non-cantilevered base could be included in the mode shape selection, by changing the base boundary condition. Stiffness loading may be appropriate.

The mode shapes can then be numerically integrated. The integrated mode shapes can then be incorporated into the component M and V. This is facilitated by noting that the differential mass matrix is integrated over the lumped mass at the base of the link m_0, the link with area density ρA, and the tip mass m_t. The resulting mass matrix has the form

$$M = \int_0^L M'(r)\mathrm{d}m = m_0 M(0) + \int_0^L \rho A M(r)\mathrm{d}r + m_t M(L) \tag{5.81}$$

Therefore, for the $(i,j)^{th}$ element of $M'(r)$ which is a function of $\phi_i(r)$,

$$M'_{ij}(r) = \phi_i(r) \tag{5.82}$$

M will contain the term

$$M_{ij} = m_0\phi_i(0) + \int_0^L \rho A\phi_i \mathrm{d}r + m_t\phi_i(L) \tag{5.83}$$

Similarly for products of the mode shapes. In effect the mass matrix and inertia forces can be integrated by inspection (given the integrals of the mode shapes). The mode shape integrations are done off-line.

The result of the mass matrix and state-dependent force integration is an expression for the unforced, free boundary condition dynamics for the link component:

$$M(q)\ddot{q} + V(q,\dot{q}) = 0 \tag{5.84}$$

Note that the mass matrix and the state dependent forces are both functions of the model DOF. $M(q)$ is generally specifically a function of the deflection DOF only. V is generally a function of rigid body DOF as well as flexible DOF. In the next section, the forcing functions needed for the attachment model are calculated.

5.3.4 Boundary Force Influence Matrices

The present section discusses how to determine the force influence matrices on the link. The attachment forces are taken to act at the left, or base end of the link, and the right, or free end. Two types of generalized forces act: linear forces F and moments M. These generalized forces act in distinct ways on the rigid body translations, the rigid body rotations, and the deflections. The force influence matrix

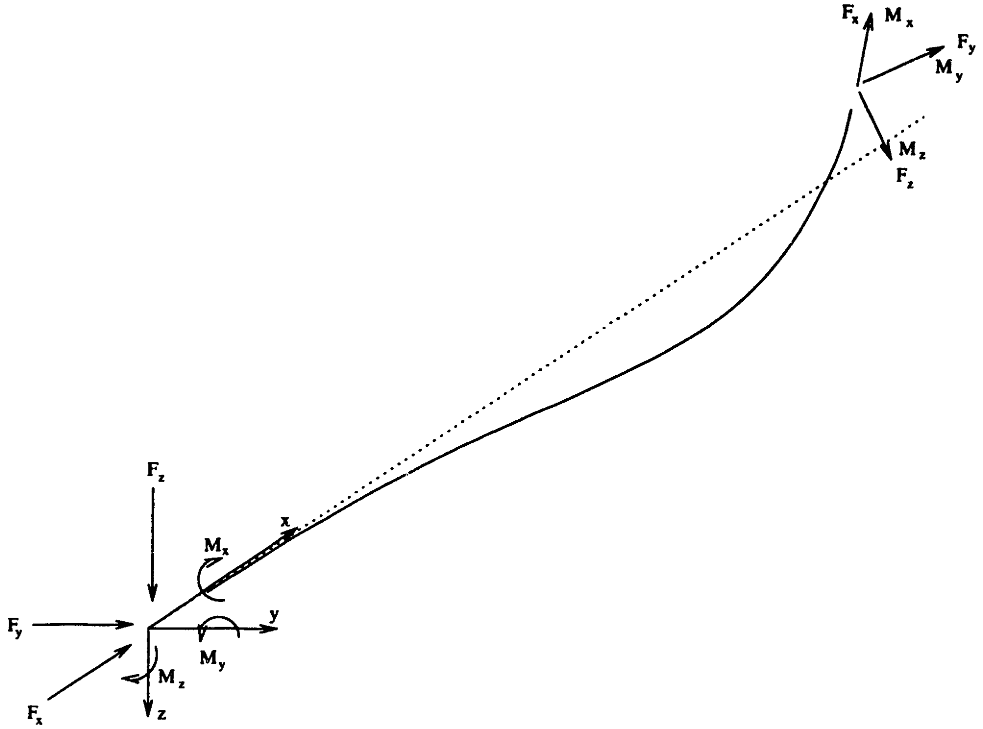

Figure 5-4: Boundary force inputs.

will be determined for each type of force, for each of the three types of generalized coordinates

$$B(r) = \begin{bmatrix} B_{0F} & B_{0M} \\ B_{\theta F} & B_{\theta_M} \\ B_{wF} & B_{wM} \end{bmatrix} \tag{5.85}$$

where again the force influence matrix is a function of the distance along the link. The first subscript, 0, θ, or w, denotes which set of generalized coordinates is affected. The second denotes the type of the generalized force (force or moment).

The attachment boundaries for the link will be at the left and right ends. The generalized forces which act at those locations will be denoted F_l and F_r, respectively, with the force and moment components as follows:

$$F_l = \begin{bmatrix} F_{x_l} & F_{y_l} & F_{z_l} & M_{x_l} & M_{y_l} & M_{z_l} \end{bmatrix}^T \tag{5.86}$$

$$F_r = \begin{bmatrix} F_{x_r} & F_{y_r} & F_{z_r} & M_{x_r} & M_{y_r} & M_{z_r} \end{bmatrix}^T \tag{5.87}$$

The linear components of F_l act along the base axis system, and the moment components act around the same axes (Figure 5-4). The components of F_r act along the axes of a system fixed to the far end of the link, which translates and rotates with respect to the base as a function of the flexible deflection. Thus the influence matrices are functions of the deflections.

The general force influence matrix for a point on the link at a distance r from the base will be established. The components for the translational DOF q_0, rotations q_θ, and flexible DOF q_w, will be determined separately. Beginning with the translational DOF, forces act along the axes fixed to the link at point r. Defining R_w as the rotation from link to base axis systems, the force on q_0 is

$$\begin{aligned} F_0 &= R_w F && (5.88) \\ &= B_{0F} F && (5.89) \end{aligned}$$

Moments do not drive the translational DOF.

Next the influence of the generalized forces on the rotational DOF q_θ are considered. The linear forces produce a moment on the base by acting through a moment arm $d = \begin{bmatrix} r & \hat{w}_y & \hat{w}_z \end{bmatrix}^T$, where the deflections are again rotated back out from the torsionally rotated frame to the base frame. Using the matrix form of the expression $M = d \times F$, with forces rotated into the base frame,

$$\begin{aligned} F_\theta &= \begin{bmatrix} 0 & -\hat{w}_z & \hat{w}_y \\ \hat{w}_z & 0 & -r \\ -\hat{w}_y & r & 0 \end{bmatrix} R_w F && (5.90) \\ &= B_{\theta F} F && (5.91) \end{aligned}$$

The moments at r are rotated into the link frame

$$\begin{aligned} M_\theta &= R_w M && (5.92) \\ &= B_{\theta M} M && (5.93) \end{aligned}$$

where they directly act on the rotational DOF.

Finally, the deflection influence matrices are considered. The forces act directly on the flexible DOF. The force influence matrix is

$$F_w = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \phi_y & 0 \\ 0 & 0 & \phi_z \end{bmatrix} F \tag{5.94}$$

$$= B_{wF}F \tag{5.95}$$

The moment influence matrix acts proportional to the slope of the deflection:

$$M_w = \begin{bmatrix} \phi_x & 0 & 0 \\ 0 & 0 & \phi'_y \\ 0 & -\phi'_z & 0 \end{bmatrix} M \tag{5.96}$$

$$= B_{wM}M \tag{5.97}$$

The general expression for the force influence matrix $B(r)$ is found by collecting all the above matrices. Before this can be done, the deflection into the base frame (R_w) must be known.

The rotation of the end of the link is a function of torsion w_x and of the slopes of the transverse deflections w'_y and w'_z. The rotation can be decomposed into a sequence of three rotations about each axis; for example, rotation about the local z, the (now rotated) axis y, and the torsion axis x:

$$R_w = R_x R_y R_z \tag{5.98}$$

with

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(w_x) & \sin(w_x) \\ 0 & \sin(w_x) & \cos(w_x) \end{bmatrix} \tag{5.99}$$

$$R_y = \begin{bmatrix} \cos(-w'_z) & 0 & -\sin(-w'_z) \\ 0 & 1 & 0 \\ \sin(-w'_z) & 0 & \cos(-w'_z) \end{bmatrix} \tag{5.100}$$

$$R_z = \begin{bmatrix} \cos(w'_y) & -\sin(w'_y) & 0 \\ \sin(w'_y) & \cos(w'_y) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.101}$$

However, this rotation is not consistent with the definition of the deflection DOF (in which the transverse deflections are independent). In effect the above choice for R_w defines the deflection DOF in terms of Euler angles for the point on the link.

This difficulty can be addressed by assuming infinitesimal deflection rotations. Small angle assumptions hold; further, products of deflections vanish, leaving only the linear terms in flexible deformations. In this case, the deflection rotation matrix Equation 5.98 becomes

$$R_w = \begin{bmatrix} 1 & -w_y' & w_z' \\ w_y' & 1 & -w_x \\ -w_z' & w_x & 1 \end{bmatrix} \tag{5.102}$$

As can be seen, for infinitesimal rotations, the order of the sequence of rotations does not matter.

The force influence matrix as a function of r can now be assembled:

$$B(r) = \begin{bmatrix} B_{0F} & 0 \\ B_{\theta F} & B_{\theta_M} \\ B_{wF} & B_{wM} \end{bmatrix} \tag{5.103}$$

where the block elements are as described above. The dependence on r arises from an explicit dependence, for example as a moment arm of a linear force on the rigid body rotations. An implicit dependence also arises from the dependence on the deflections w. The left and right influence matrices are then evaluated at $r = 0$ and $r = L_t$ respectively.

5.3.5 Boundary Acceleration Functions

The outputs used in the system model are the linear and rotational accelerations at both ends of the link. These can be easily calculated in terms of the velocity functions defined during the unforced modeling process. The translational velocity as a function of r, expressed in the base frame, is

$$\dot{p} = \begin{bmatrix} I & A(w) \end{bmatrix} \dot{q} \tag{5.104}$$

and the angular velocity is

$$\dot{\Theta} = \begin{bmatrix} 0 & A_r(w) \end{bmatrix} \dot{q} \tag{5.105}$$

Expressing the velocities in the frame which lies on the rotated link produces the following expression:

$$\begin{bmatrix} \dot{p} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} R_w^T & R_w^T A \\ 0 & R_w^T A_r \end{bmatrix} \dot{q} \tag{5.106}$$

Differentiating,

$$\begin{bmatrix} \ddot{p} \\ \ddot{\Theta} \end{bmatrix} = \begin{bmatrix} R_w^T & R_w^T A \\ 0 & R_w^T A_r \end{bmatrix} \ddot{q} + \begin{bmatrix} 0 & (\dot{R_w^T A}) + \Omega A \\ 0 & (\dot{R_w^T A_r}) \end{bmatrix} \dot{q} \tag{5.107}$$

which has the form

$$a = H(q, \dot{q}) + G(q, \dot{q})\ddot{q} \tag{5.108}$$

Substituting $r = 0$ for the left end, and $r = L_t$ for the right end, gives the corresponding accelerations:

$$a_l = H_l + G_l \ddot{q} \tag{5.109}$$

$$a_r = H_r + G_r \ddot{q} \tag{5.110}$$

These are the *acceleration BCs* of the link.

5.3.6 Link Component Model

Collecting the link homogeneous dynamics, the forcing matrices, and the output observation functions, the free-free component model for the link can be written

$$\begin{aligned} M\ddot{q} &= (-V(q, \dot{q}) + B_u u) + B_l F_l + B_r F_r \\ a_l &= H_l(q, \dot{q}) + G_l \ddot{q} \\ a_r &= H_r(q, \dot{q}) + G_r \ddot{q} \end{aligned} \tag{5.111}$$

with the matrix elements known from the above equations. Note that the control forces u have been lumped into the homogeneous dynamics of the component since they are independent of the BCs. In fact, in the system model these torques may be

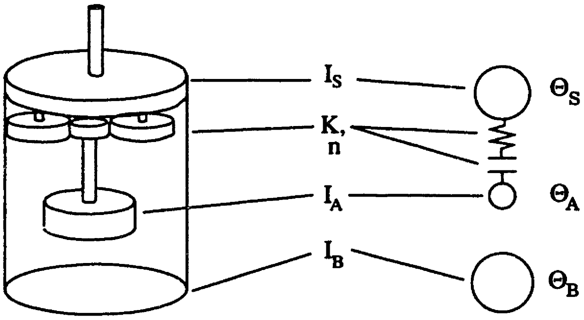

Figure 5-5: Joint model consisting of case, armature, gear ratio, lumped stiffness, and output shaft.

functions of the system states; the salient feature is that they are known in advance (either being externally imposed, or by being known functions of the states in the same way that V is k:1own).

5.4 Gearbox Component Model

Joint motors are modeled with a single shaft degree of freedom, in a unit which will be termed a "gearbox". Multiple articulations (such as the pitch and yaw at the base of the SRMS) are modeled with two separate gearboxes attached together. The gearbox is modeled in three sections: a case, an armature, and an output plate. The case is mounted to a supporting structure, either the end of a link, or the base of the structure. The armature is the free-spinning portion of the motor to which commanded torques are applied. The armature is connected through a gear reduction unit to the output shaft. The output shaft is in turn connected to the base of a link member, and applies torque to it. The gearset will typically have flexibility all through the load path, due to deflection of the gear teeth, and torsion of the gear shaft. For the present work, the gearbox flexibility will be reflected outboard of the gearbox. Note that a gearbox has a single shaft degree of rotational freedom. In all translation axes, and in the other two rotations, the gearbox reacts as a lumped mass.

The joint Lagrangian has two components: kinetic energy from an armature degree of freedom, and a potential energy term arising from strain in the gearbox stiffness due to the motion of the armature, the joint casing, and the joint output plate. The potential energy term will cause torques on the joint degree of freedom and on the links to which the gearbox is attached. Note that an independent component model is desired. The stiffness forces depend on other components, thus they will be represented separately. The gearbox model will consist only of the inertial forces introduced by gearbox elements. The stiffness forces will be introduced as a state-dependent feedback in the system model. The parameters in the gearbox model are case mass and inertia, armature mass and inertia, output plate mass and inertia, gear ratio n, and stiffness K.

5.4.1 Gearbox Kinetic Energy

The DOF of the free gearbox are translational displacements of the center of mass (C.M.) $\dot{x}, \dot{y}, \dot{z}$, rotations about the gearbox x and z axes $\dot{\theta}_x$ and $\dot{\theta}_z$, and rotations of the base, armature, and output shaft $\dot{\theta}_B, \dot{\theta}_A, \dot{\theta}_S$. By convention the y axis is the shaft axis. The kinetic energy function for the gearbox is

$$T_g = \frac{1}{2}(m_g\dot{x}^2 + m_g\dot{y}^2 + m_g\dot{z}^2 + I_B\dot{\theta}_B^2 + I_A\dot{\theta}_A^2 + I_S\dot{\theta}_S^2 + J_{xx}\dot{\theta}_x^2 + J_{zz}\dot{\theta}_z^2) \tag{5.112}$$

with m_g the total mass of the case, armature, and output, J_{xx} and J_{zz} the angular inertia about the center of mass, and I_B, I_A, I_S are the base, armature, and shaft inertias.

Because the gearbox translates as a unit (independent of configuration), the terms in the EOM which arise from translational velocity of the gearbox C.M. are best dealt with by associating them with the EOM of the base link (that is, the link to which the base of the gearbox is attached). The C.M. inertial velocities are given by

$$\dot{p} = \dot{x}_0 + R_x(\dot{x}_L + \Omega_w \times b) \tag{5.113}$$

where Ω_w is the rotation rate of the base link endpoint due to deflection:

$$\Omega_w = \begin{bmatrix} \dot{w}_x \\ -\dot{w}'_z \\ \dot{w}'_y \end{bmatrix} \tag{5.114}$$

and $b = \begin{bmatrix} b_x & b_y & b_z \end{bmatrix}$ is the offset of the gearbox center of mass. The last term in Equation 5.113 can be re-written in terms of an offset matrix B_x and deflections $w = \begin{bmatrix} w_x & w_y & w_z \end{bmatrix}^T$, as before:

$$\Omega_w \times b = B_x \dot{w} \tag{5.115}$$

where the offset matrix is the cross-product matrix formed from the elements of b and the differential operators which act on w_y and w_z:

$$B_x = \begin{bmatrix} 0 & b_z & -b_y \\ -b_z & 0 & b_x \\ b_y & -b_x & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -\frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial x} & 0 \end{bmatrix} \tag{5.116}$$

The C.M. inertial velocity vector is then written

$$A_g = A(L) + \begin{bmatrix} 0 & B_x \end{bmatrix} \tag{5.117}$$

where $A(L)$ is given by Equation 5.34 evaluated at the link endpoint. The velocity term A_g is then evaluated in Equations 5.26-5.28.

Considering the rotational DOF, the Lagrange's equation gives the gearbox EOM as

$$\begin{aligned} \begin{bmatrix} J_B & & & & \\ & J_A & & & \\ & & J_S & & \\ & & & J_x & 0 \\ & & & & I_z \end{bmatrix} \begin{bmatrix} \ddot{\theta}_B \\ \ddot{\theta}_A \\ \ddot{\theta}_S \\ \ddot{\theta}_x \\ \ddot{\theta}_z \end{bmatrix} = & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}_L + \\ & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix}_R \end{aligned} \tag{5.118}$$

$$a_l = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_B \\ \ddot{\theta}_A \\ \ddot{\theta}_S \\ \ddot{\theta}_x \\ \ddot{\theta}_z \end{bmatrix} \tag{5.119}$$

$$a_r = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_B \\ \ddot{\theta}_A \\ \ddot{\theta}_S \\ \ddot{\theta}_x \\ \ddot{\theta}_z \end{bmatrix} \tag{5.120}$$

where the boundary accelerations are simply the states. There is no state-dependent component to the outputs. Note that the "right" and "left" designations R and L are not physically meaningful, as they are for the link, since the forces act at the same physical location. The notation is used for consistency, and does represent distinct forces. In the parlance of the link component modeling section, the generalized forces and accelerations represent the force and acceleration BCs of the gearbox model. Note also that the rotational component model has the same form as the link model, Equation 5.111.

5.4.2 Gearbox Potential Energy

Next, the torques exerted by the gearbox stiffness on the base, shaft, and armature are determined. The torques exerted by the spring are found by evaluating Lagrange's equation using the potential energy stored in the spring. The potential energy is formulated by defining the intermediate variables θ_1' and θ_2' which are the rotation of the armature and the output shaft, θ_A and θ_S respectively, with respect to the casing θ_B:

$$\theta_1' = \theta_A - \theta_B \,, \theta_2' = \theta_S - \theta_B \tag{5.121}$$

The potential energy is a function of the armature motion, θ'_1, reflected through the gear ratio, and the output motion:

$$P = \frac{1}{2}\left(\theta'_2 - \frac{1}{n}\theta'_1\right)^2 \tag{5.122}$$

$$= \frac{1}{2}\left((\theta_S - \theta_B) - \frac{1}{n}(\theta_A - \theta_B)\right)^2 \tag{5.123}$$

Calculating the contributions of base, armature, and output rotation to the respective balance of forces,

$$\frac{\partial P}{\partial \theta_B} = \frac{(n-1)^2}{n^2}K\theta_B + \frac{n-1}{n^2}K\theta_A + \frac{1-n}{n}K\theta_S \tag{5.124}$$

$$\frac{\partial P}{\partial \theta_A} = \frac{n-1}{n^2}K\theta_B + \frac{1}{n^2}K\theta_A - \frac{1}{n}K\theta_S \tag{5.125}$$

$$\frac{\partial P}{\partial \theta_S} = \frac{1-n}{n}K\theta_B - \frac{1}{n}K\theta_A + K\theta_S \tag{5.126}$$

Collecting all the terms, moving them to the left hand side of the EOM, and solving for the forces exerted on the gearbox base M_B, armature M_A, and output shaft M_S,

$$\begin{bmatrix} M_B \\ M_A \\ M_S \end{bmatrix} = K \begin{bmatrix} \frac{(n-1)^2}{n^2} & \frac{n-1}{n^2} & \frac{1-n}{n} \\ \frac{n-1}{n^2} & \frac{1}{n^2} & -\frac{1}{n} \\ \frac{1-n}{n} & -\frac{1}{n} & 1 \end{bmatrix} \begin{bmatrix} \theta_B \\ \theta_A \\ \theta_S \end{bmatrix} \tag{5.127}$$

The gearbox torques are functions of the base link tip angle θ_B, armature DOF θ_A, and output shaft angle θ_S. As a result they are functions of the states of several components. They can be calculated as a state-dependent feedback matrix, where the state vector is the global, system state, and lumped into the appropriate state force vectors V.

5.5 Conclusion

The preceding chapter has developed the component EOM for a flexible link and a gearbox with flexible drive train. The EOM are found in closed form. These two components can be used in the modeling framework of Chapter 4 to capture many interesting geometrically nonlinear flexible structures. One of specific interest is the

SRMS. The following chapter will develop a model of the SRMS using the component modeling approach. The model will be compared to a model developed using standard tools (Lagrange's equation applied to the entire manipulator). Next the component model will be used to develop a gain-scheduled controller for the SRMS, which will be evaluated on the Lagrange SRMS model.

Chapter 6

Verification

The modeling algorithm is verified against a standard Lagrange modeling approach. The verification has two separate purposes:

(i) Confirm that the model is of comparable accuracy to the standard approach, and compare on the basis of modeling effort and run-time computational requirements.

(ii) Use the model to create a controller, to be implemented in simulation on the Lagrange truth model.

The Lagrangian SRMS model developed for Chapter 2 is used as the truth model for the first aspect of verification. In the second stage of verification, the SRMS truth model plays the role of a "testbed" for application of control using the linearized component model, in conjunction with the design algorithm of Chapter 3.

The chapter road map is as follows: first the SRMS is described. The physical characteristics of the model, along with the parameters used in the truth model, are used to develop the component model configuration. The component models are described. The boundary conditions are derived. Finally, the nonlinear simulation form, and linearized control form are assembled. Next, the models are verified on the basis of frequency and time domain metrics. Finally, a gain-scheduled controller is designed and implemented on the truth model.

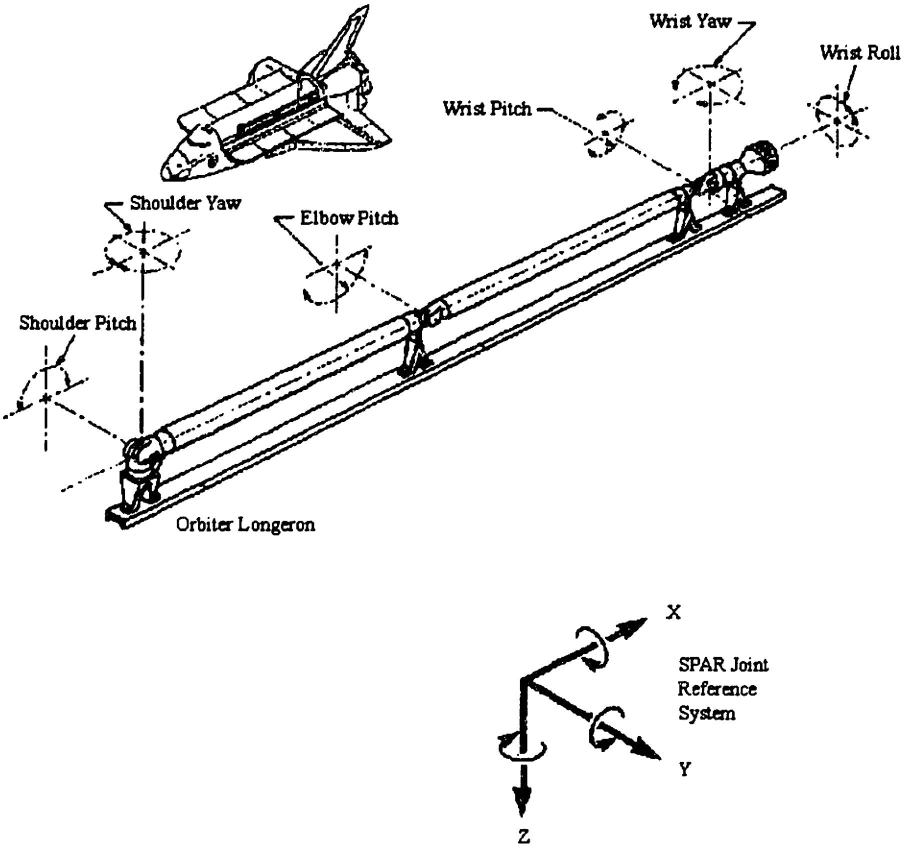

Figure 6-1: Space Shuttle Remote Manipulator System.

6.1 SRMS Component Model

6.1.1 SRMS Overview

This section gives an overview of the Shuttle Remote Manipulator System. The Shuttle Manipulator System is a seven joint, two link flexible manipulator[80] (see Figure 6-1). Its length at full extension is 50 feet. Links are carbon fiber tubes. Joints consist of servo motors with planetary gear reducers. Tachometers are used as feedback sensors to accurately control each joint's rate independently[67]. Each joint is also fitted with a mechanical brake. A locking end-effector is mounted at the end of the arm, to deploy, capture, and maneuver payloads. Movable cameras at the elbow and end-effector assist in positioning the end-effector.

Following standard robotics terminology, the *proximal* link is the link closest to the Shuttle. The second link is referred to as the *distal* link. Turning to joint nomenclature, the *base swingout* joint deploys the arm from its stowed configuration along the Shuttle cargo bay wall. Once the arm is deployed, the swingout joint is locked in position. The next joint along the arm is the *shoulder yaw* joint, which actuates the arm around the shuttle local vertical axis. The *shoulder pitch* joint is attached to the shoulder yaw joint, and actuates pitch motion of the first link. The *elbow pitch* joint is located at the elbow, between the proximal and distal links, and actuates the pitch of the distal link. The *wrist roll, pitch, and yaw* joints are located at the end of the distal link, and control the roll, pitch, and yaw rotations, respectively, of the end-effector. The end-effector consists of a grappling fixture with a rigidizing mechanism that locks the end-effector to the payload grapple fixture.

A number of simplifications are made at the system level to reduce the modeling effort for this verification model. The Shuttle is assumed to be inertially fixed. The swingout joint is locked in the deployed position. A suitable base compliance is chosen to represent both the compliance of the swingout joint as well as the Shuttle wall. The wrist joints are also assumed locked and rigid (that is, no wrist rotation or flexibility is incorporated into the model). Therefore, three joint degrees of freedom are to be included in the model. Each joint is modeled with a flexible gearbox, whose parameters are an armature inertia, a reduction ratio, and a single lumped stiffness. The payload is assumed to be a rigid, lumped mass at the tip of the end-effector. The SRMS as it is to be modeled is shown in Figure 6-2.

The next section will evaluate the simplified SRMS to determine the parameters necessary for the component modeling method.

6.1.2 Configuration

The simplified SRMS presented above must be reduced to a set of parameters suitable for the component modeling algorithm. Some conventions and terms will be defined before proceeding. The joints are referred to with the indices 1, 2 and 3, starting at the Shuttle and moving outwards. The nominal configuration for the system,$(\theta_1, \theta_2, \theta_3) = (0, 0, 0)$, will be fully extended, with the links parallel to the Shuttle wall. In the

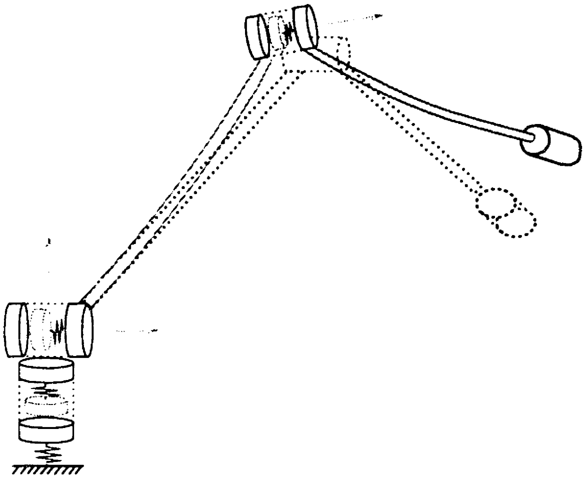

Figure 6-2: Simplified 3-joint model of the SRMS, used to verify the modeling algorithm. Arrows mark rotation axes of the joints.

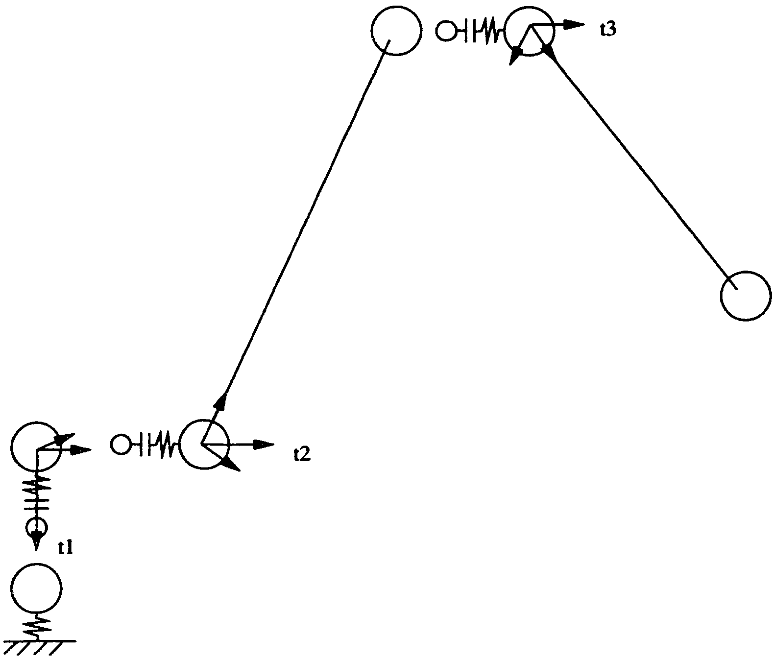

Figure 6-3: Mass and flexibility distribution of the 3D SRMS model. Joint rotations are shown by the arrows, labeled θ_1, θ_2, and θ_3.

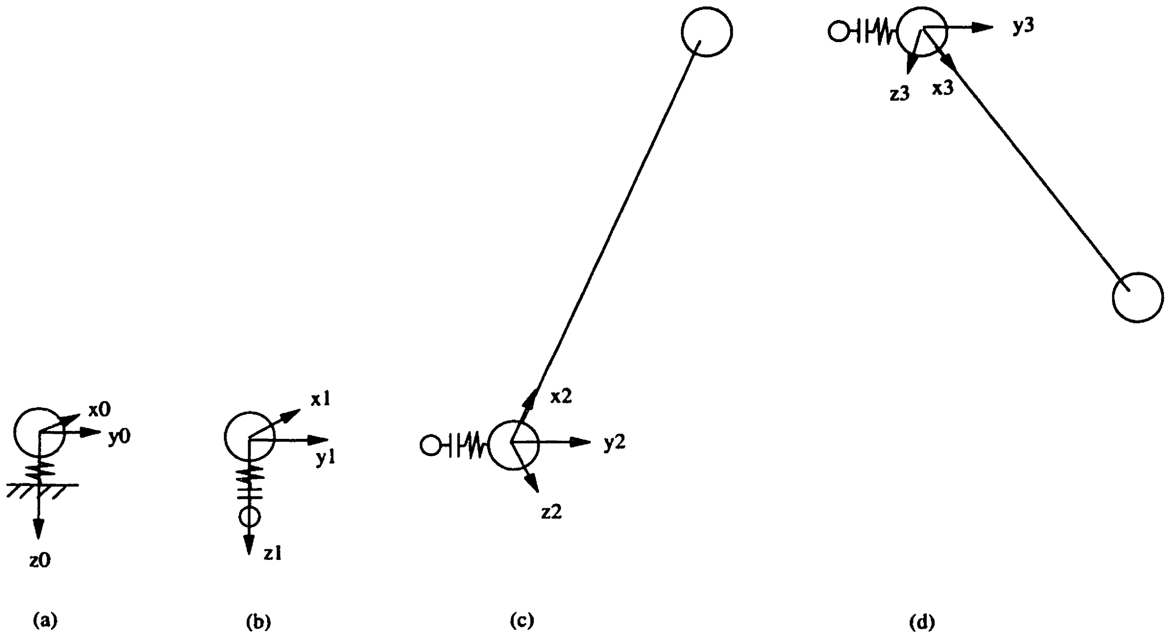

Figure 6-4: SRMS component models, and associated axis systems: (a) base swingout joint, (b) shoulder yaw joint, (c) proximal link, (d) distal link.

nominal configuration, all component axis systems have the x axis pointing along the manipulator, and the z axis downwards. The y axis completes a right-handed coordinate system. This definition is consistent with the Shuttle body-referenced coordinate system[80]. The *left* or *base* end of either link is the end nearest the Shuttle. The *right* or *free* end is nearest the end-effector.

The SRMS system model is shown in Figure 6-3. Each circle denotes a lumped mass. Jagged lines denote a lumped stiffness. Parallel lines denote a gear reduction. Single lines represent flexible links. Axis systems are positioned at all joint rotations, with the axis of rotation for joint i denoted with the symbol θ_i. Note that the orientation of each axis system varies with the motion of the joint to which it is attached. For example, the elbow pitch axis orientation is a function of the proximal link rigid body rotations and deflections. Commands to each joint of the manipulator, u_i, enter as torque couples between the base and the armature of each gearbox. The task of the configuration definition process is to realize the system model as a set of independent components, and a set of coupling rotations.

The components which comprise the system model are determined first. The components must be suitable for independent modeling. Because the joint rotations require coupling between two systems, components generally will be comprised of all elements between any two joints. In the case of the SRMS model, each axis system is associated with a component. Noting that the swingout joint DOF are defined in the Shuttle inertial frame, the components which make up the SRMS model are shown in Figure 6-4. From left to right, the first component is the base swingout joint, defined in the inertial 0 reference frame. The second component is the shoulder yaw joint, modeled with a gearbox in the 1 frame. The third component consists of the proximal link and the lumped masses and inertias of the shoulder pitch joint and the elbow pitch gearbox base. The proximal link is modeled with a mass-loaded flexible link, in the 2 frame. The fourth and final component consists of the armature and shaft inertia of the elbow pitch joint, the distal link, and the payload mass and inertia. The distal link is modeled in the 3 frame.

It must be emphasized again that the gearbox stiffnesses act to couple component models, as shown in Chapter 5. In order to preserve the independence of each component model, the stiffnesses will be treated at a system level, not at the component level. The component models will not include the stiffness terms. As a result, the gearbox armature dynamics are not coupled to the links as shown in Figure 6-4. Considering the armature DOF to be independent components will modestly reduce the computational requirements of the system dynamics (by splitting a single mass matrix into two smaller matrices) at the cost of some additional notational cost for carrying extra component models.

Next the free and constrained DOF associated with each component are identified:

(i) swingout joint: all DOF are free.

(ii) shoulder yaw gearbox: the free, z_1 axis of rotation is aligned vertically. The x_1 and y_1 axes are constrained (representing the rigid case of the gearbox).

(iii) proximal link: free DOF consist of the armature rotation, link rigid body y_2

rotation, and flexible deformations. The constrained DOF are rigid body x_2 and z_2 rotations.

(iv) distal link: The free DOF of the distal link are, again, armature rotation, link y_3 rotation, and flexible deformations. The constrained DOF are x_3, y_3, and z_3 translations, and x_3 and z_3 rotations.

Consistent with Chapter 5, the flexible deformations are represented with assumed modes DOF. The proximal link flexible DOF are q_{x2} in torsion, q_{y2} in y_2 translation, and q_{z2} in z_2 translation. Similarly, deformations of the distal link are represented with q_{x3}, q_{y3}, and q_{z3}. For consistency with the Lagrangian truth model, one assumed modes DOF is used to capture torsional vibration, and two each for translational vibration. The free DOF are listed in Table 6.1. Note that the notation $a : b$ refers to a range of indices from a to b, i.e. $(a, a+1, \cdots, b)$

Note that armature inertias for the shoulder pitch and elbow pitch armatures are split out of the proximal link and distal link components, respectively, for modest computational gains. In contrast the shoulder yaw gearbox component includes the motor armature inertia. This is a matter of notational convenience in capturing the rigid body rotations of the shoulder yaw gearbox.

The constrained DOF are listed in Table 6.2. Each set of constrained DOF, for each component, is assigned a vector-valued variable a_{ci} which denotes an input to that component at the *constrained* location. The index denotes the component number. The a indicates that the quantity is an acceleration input. Note also that each of the constrained rotations has an associated reaction force, which is also listed in the table. Again, the reaction force is assigned a variable F_{ci} representing the force BC output, at the constrained location, for component i.

Next the free acceleration outputs and force inputs, for each component, are specified. This is done using knowledge of the attachment locations between components. Beginning at the shoulder, the swingout joint couples to the shoulder yaw gearbox. The shoulder yaw x_1 and z_1 rotations must be driven by the swingout joint. Since the swingout joint z_1 axis is coincident with the shoulder yaw rotation axis y_1, the swingout joint driving axes are the x_0 and y_0 axes. The swingout joint component has corresponding angular acceleration outputs, as listed in Table 6.3. The component BC outputs are denoted as a_{f0}, referring to the *free* location acceleration outputs from component 0. The swingout joint must also accept the reaction forces from the gearbox. The BC force inputs to the swingout joint are also listed in the table, and denoted F_{f0}.

Moving to the proximal link, the constrained x_2 and z rotations are driven by the shoulder yaw gearbox x_1 and y_1 rotations. Finally, the distal link x_3, y_3, z_3 translations and x_3 and z_3 rotations are driven by the proximal link.

The required information from the configuration definition is now available. The component models can be derived in suitable fashion, knowing the necessary DOF, and the force inputs and acceleration outputs. The model inversion can be completed knowing the constrained DOF. The component boundary conditions can be defined in terms of joint rotation matrices, and used to assemble the global boundary condition equation.

6.1.3 Component Models

The components to be captured include the swingout joint, three gearboxes, and two links. The swingout joint is a simple spring-mass system. The gearboxes and links are captured with the models presented in Chapter 5. Any suitable modeling method can be used, which returns:

(i) the component models in terms of inertially referenced DOF, which include the free DOF in Table 6.1 and the constrained DOF Table 6.2. The component exogenous inputs are also identified.

(ii) the reaction forces in Table 6.2 and the free forces in Table 6.3, as force inputs to the model, *in the frame attached to the link.*

(iii) the acceleration output functions corresponding to the constrained DOF in Table 6.2 and the free accelerations in Table 6.3, again in the frame attached to the link.

Table 6.1: SRMS model states

Component	DOF(s) #	Name	Physical
Swingout joint	1	θ_{x0}	base x rotation
	2	θ_{y0}	base y rotation
	3	θ_{z0}	base z rotation
Shoulder yaw	4	θ_{A_1}	shoulder yaw armature
	5	θ_{z_1}	shoulder yaw output
Shoulder pitch	6	θ_{A_2}	shoulder pitch armature
Proximal Link	7	θ_{y_2}	rigid body y rotation
	8	q_{x_2}	torsional deflection
	9:10	q_{y_2}	y transverse deflection
	11:12	q_{z_2}	z transverse deflection
Elbow pitch	13	θ_{A_3}	elbow armature
Distal Link	14	θ_{y_3}	rigid body y rotation
	15	q_{x_3}	torsional deflection
	16:17	q_{y_3}	transverse y deflection
	18:19	q_{z_3}	transverse z deflection

Table 6.2: SRMS model constrained DOF

Component	Name	Physical	BC Input	Reaction force	BC Output
Shoulder pitch	θ_{x_1}	x rot.	a_{c_1}	M_{x_1}	F_{c_1}
	θ_{y_1}	y rot.		M_{y_1}	
Proximal Link	θ_{x_2}	RB x rot.	a_{c_2}	M_{x_2}	F_{c_2}
	θ_{z_2}	RB z rot.		M_{z_2}	
Distal Link	x_3	RB x disp.		F_{x_3}	
	y_3	RB y disp.		F_{y_3}	
	z_3	RB z disp.	a_{c_3}	F_{z_3}	F_{c_3}
	θ_{x_3}	RB x rot.		M_{x_3}	
	θ_{z_3}	RB z rot.		M_{z_3}	

Table 6.3: SRMS model free forces and accelerations

Component	Acceleration	BC Output	Force	BC Input
Swingout joint	θ_{x_0}	a_{f_0}	M_{x_0}	F_{f_0}
	θ_{z_0}		M_{z_0}	
Shoulder pitch	θ_{x_1}	a_{f_1}	M_{x_1}	F_{f_1}
	θ_{z_1}		M_{z_1}	
Proximal Link	x_3	a_{f_2}	F_{x_2}	F_{f_2}
	y_3		F_{y_2}	
	z_3		F_{z_2}	
	θ_{x_3}		M_{x_2}	
	θ_{z_3}		M_{z_2}	

The restriction on the frame in which inputs and outputs are expressed is necessary in order that rigid body rotations and translations are expressed correctly at the attachment location. In addition, the *exogenous* inputs are generalized forces which act on the free degrees of freedom. When the system model is assembled, some of these forces will contain state-dependent terms (such as gearbox stiffness terms). In addition the commands will be generated by a feedback controller. So the term "exogenous" describes forces at the component level, but not necessarily at the system level.

Base Swingout Joint

The swingout joint takes the SRMS from its stowed configuration along the side of the payload bay, to its deployed configuration. The SRMS model simplifies the swingout joint to a deployed and locked configuration. The flexibility parameters model both the swingout joint and the Shuttle cargo bay wall. The x and y inertia parameters include the inertia of the shoulder yaw gearbox. The swingout joint exogenous input consists of the reaction torque from the shoulder yaw command, $-u_1$, and the base torque from the shoulder yaw gearbox, m_{B1}. Both torques act around the z axis.

The swingout joint EOM are

$$M_0 \begin{bmatrix} \ddot{\theta}_{x0} \\ \ddot{\theta}_{y0} \\ \ddot{\theta}_{z0} \end{bmatrix} = V_0 + B_{F_0} F_{c_0} \tag{6.1}$$

$$a_{f_0} = G_{aF_0} F_{c_0} + G_{a_0} \begin{bmatrix} \ddot{\theta}_{x0} \\ \ddot{\theta}_{y0} \\ \ddot{\theta}_{z0} \end{bmatrix} \tag{6.2}$$

The swingout joint parameters are given in Table 6.4. The notation is chosen consistent with the SRMS Lagrangian model. Note that the swingout joint EOM are given in the component model standard form, Equation 4.57:

$$M_0 = \begin{bmatrix} I_0 & 0 & 0 \\ 0 & I_{41} & 0 \\ 0 & 0 & I_1 \end{bmatrix} \tag{6.3}$$

$$V_0 = \begin{bmatrix} -K_0 \theta_{x_0} \\ -K_{31} \theta_{y_0} \\ -K_1 \theta_{z_0} - u_1 + m_{B_1} \end{bmatrix} \tag{6.4}$$

$$B_{F_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.5}$$

$$H_{a_0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.6}$$

$$G_{aF_0} = \begin{bmatrix} \frac{1}{I_0} & 0 \\ 0 & \frac{1}{I_{41}} \end{bmatrix} \tag{6.7}$$

$$G_{a_0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{6.8}$$

Terms in the acceleration input (B_a, G_{aa}, etc.) are zero since the swingout joint is inertially fixed. Reaction force outputs (H_F, ...) are likewise zero.

Table 6.4: Base Swingout Joint Parameters

Parameter	Value	Units	Description	Reference
I_0	4.0	slug-ft^2	base x inertia	[81]
K_0	161510	ft-lb/rad	base x stiffness	
I_{41}	4.0	slug-ft^2	base y inertia	
K_{31}	493280	ft-lb/rad	base y stiffness	
I_1	1.0	slug-ft^2	base z inertia	
K_1	443320	ft-lb/rad	base z stiffness	

Shoulder Yaw Gearbox

The shoulder yaw gearbox actuates around a vertically oriented axis. The gearbox is modeled as a motor armature inertia, an output shaft inertia, lumped flexibility reflected to the output side of the reduction, and a gear ratio. The gearbox mass is ignored since the shoulder joint does not translate. The inertias orthogonal to the shaft are lumped into the swingout joint parameters.

The z axis exogenous torques are the shoulder yaw command, u_1, and the shoulder yaw gearbox torques m_{A_1} and m_{S_1} on the armature and output shaft, respectively. The y axis inputs consist of the shoulder pitch command reaction torque, $-u_2$, and the shoulder pitch gearbox base torque m_{B_2}.

The component model form for the EOM in standard form is

$$M_1 \begin{bmatrix} \ddot{\theta}_{A1} \\ \ddot{\theta}_{S1} \end{bmatrix} = V_1 + B_{a_1} a_{c_1} + B_{F_1} F_{c_1} \tag{6.9}$$

$$F_{f_1} = H_{F_1} + G_{Fa_1} a_{c_1} + G_{FF_1} F_{c_1} + G_{F_1} \begin{bmatrix} \ddot{\theta}_{A1} \\ \ddot{\theta}_{S1} \end{bmatrix} \tag{6.10}$$

$$a_{f_1} = G_{aa_1} a_{c_1} + G_{a_1} \begin{bmatrix} \ddot{\theta}_{A1} \\ \ddot{\theta}_{S1} \end{bmatrix} \tag{6.11}$$

with

$$M_1 = \begin{bmatrix} I_{A_1} & 0 \\ 0 & I_{S_1} \end{bmatrix} \tag{6.12}$$

Table 6.5: Base Yaw Parameters

Parameter	Value	Units	Description	Reference
I_2	0.000271	slug-ft^2	armature inertia	[81]
I_3	0.9092447917	slug-ft^2	output inertia	
K_3	1180000.0	ft-lb/rad	stiffness	
n_1	1842.0		gear ratio	

$$V_1 = \begin{bmatrix} u_1 + m_{A_1} \\ m_{S_1} \end{bmatrix} \tag{6.13}$$

$$B_{a_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.14}$$

$$B_{F_1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.15}$$

$$H_{F_1} = \begin{bmatrix} 0 \\ -u_2 + m_{B_2} \end{bmatrix} \tag{6.16}$$

$$G_{Fa_1} = \begin{bmatrix} I_x & 0 \\ 0 & I_y \end{bmatrix} \tag{6.17}$$

$$G_{FF_1} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.18}$$

$$G_{F_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.19}$$

$$G_{aa_1} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.20}$$

$$G_{a_1} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.21}$$

Note the feedthrough of the x axis torque in G_{FF_1}. The rigid case acts to carry the input torque directly to the base. Similarly, the x axis rotation output is the x rotation of the swingout joint, indicated by the feedthrough in G_{aa_1}.

Table 6.6: Base Pitch Parameters

Parameter	Value	Units	Description	Reference
I_4	0.000271	slug-ft^2	armature inertia	[81]
I_5	0.9092447917	slug-ft^2	output inertia	
K_3	2110000.0	ft-lb/rad	stiffness	
n_2	1843.0		gear ratio	

Base Pitch Gearbox

The base pitch gearbox model consists of an armature inertia, output shaft inertia, lumped stiffness, and gear reduction. The orthogonal base inertias are captured in the swingout joint model.

The exogenous forces on the base pitch gearbox include the armature command, u_2, and the stiffness torques m_{A_2} and m_{S_2}. Note the base reaction force has already been incorporated into the base yaw joint model.

Since the gearbox shaft is rigidly fixed to the proximal link, the output inertias will be captured in the proximal link model. In addition, exogenous forces will act on the proximal link. The base pitch model therefore consists only of the armature DOF:

$$I_4 \ddot{\theta}_{A_2} = u_2 + m_{A_2} \tag{6.22}$$

The component model standard form is

$$M_{A_2} = I_4 \tag{6.23}$$

$$V_{A_2} = u_2 + m_{A_2} \tag{6.24}$$

Note however that the model does not need to be written in standard form, since none of the rotational boundary conditions acts directly on the armature. Instead, torques with joint angle dependence act through the gearbox stiffness torque m_{A_2}.

Elbow Pitch Gearbox

The elbow pitch gearbox is modeled with three lumped masses, representing the gearbox casing, the armature, and the output shaft. The geartrain is represented with a stiffness and a gear reduction. Exogenous forces arise from the elbow pitch command, u_3, and the gearbox torques m_{B_3}, m_{A_3} and m_{S_3}.

Similarly to the shoulder pitch gearbox, the elbow gearbox base is rigidly fixed to the proximal link. Therefore, the base inertias will be captured as lumped tip masses on the proximal link. The output shaft is fixed to the distal link, so the output inertias will be modeled as lumped masses at the base of the distal link. The base torque acts directly on the proximal link, and the shaft torque acts directly on the distal link.

As a result the gearbox dynamics are again independent of joint angle:

$$I_{y6}\ddot{\theta}_{A_3} = u_3 + m_{A_3} \tag{6.25}$$

so that a standard form representation is not needed.

Payload

A moderately massive payload was desired to represent operational use of the SRMS in space construction. The Shuttle Pallet Satellite 01 (SPAS-1) was chosen due to the availability of inertia parameters for the payload, and the presence of SPAS-1 payload parameters in the DRS database[81], which allowed the Lagrange model to be validated. Since the end-effector is modeled as a rigid link, the payload mass and inertia can be modeled as a lumped tip mass on the end-effector.

Proximal Link

The proximal link is modeled as a Bernoulli-Euler beam with finite cross-sectional rotational inertias. The bending portion of the link is offset from the base axis of rotation due to the radius of the gearbox case and the link attachment fixture. The elbow center of mass is offset from the tip of the bending portion of the link, due

Table 6.7: Elbow Pitch Parameters

Parameter	Value	Units	Description	Reference
I_{x6}	0.15703125	slug-ft^2	base x inertia	[81]
I_{y6}	0.31406250	slug-ft^2	base y inertia	
I_{z6}	0.15703125	slug-ft^2	base z inertia	
m_6	2.5000	slug	base mass	
I_{x7}	0.0002590025	slug-ft^2	armature x inertia	
I_{y7}	0.0002905	slug-ft^2	armature y inertia	
I_{z7}	0.0002590025	slug-ft^2	armature z inertia	
m_7	0.2	slug	armature mass	
I_{x8}	0.2180989583	slug-ft^2	shaft x inertia	
I_{y8}	0.4361979166	slug-ft^2	shaft y inertia	
I_{z8}	0.2180989583	slug-ft^2	shaft z inertia	
m_8	2.5	slug	shaft mass	
K_4	1900000.	ft-lb/rad	stiffness	
n_3	1260.		gear ratio	

to the radius of the elbow case and the elbow attachment fixture. The offsets, mass properties, and distributed stiffness parameters are given in Table 6.9.

The EOM are

$$M_2 \begin{bmatrix} \ddot{\theta}_{y2} \\ \ddot{q}_{x2} \\ \ddot{q}_{y2} \\ \ddot{q}_{z2} \end{bmatrix} = V_2 + B_{a_2} a_{c_2} + B_{F_2} F_{c_2} \tag{6.26}$$

$$F_{f_2} = H_{F_2} + G_{Fa_2} a_{c_2} + G_{FF_2} F_{c_2} + G_{F_2} \begin{bmatrix} \ddot{\theta}_{y2} \\ \ddot{q}_{x2} \\ \ddot{q}_{y2} \\ \ddot{q}_{z2} \end{bmatrix} \tag{6.27}$$

Table 6.8: Payload Parameters

Parameter	Value	Units	Description	Reference
I_{x9}	1928.	slug-ft^2	payload x inertia	[82]
I_{y9}	7063.5	slug-ft^2	payload y inertia	
I_{z9}	7843.5	slug-ft^2	payload z inertia	
m_9	238.66	slug	payload mass	

Table 6.9: Proximal Link Parameters

Parameter	Value	Units	Description	Reference
a_1	0.5	ft	base offset	[81]
b_1	0.5	ft	tip offset	
L_1	20.0	ft	bending length	
EI_1	5365100.0	lb-ft^2	bending stiffness	
GJ_1	2727400.0	lb-ft^2	torsional stiffness	
ρA_1	0.4	slug/ft	linear density	
ρI_{x1}	0.0013046821802	slug-ft^2	x linear inertia	
ρI_{y1}	0.0006523410901	slug-ft^2	y linear inertia	
ρI_{z1}	0.0006523410901	slug-ft^2	z linear inertia	

$$a_{f_2} = H_{a_2} + G_{aa_2} a_{c_2} + G_{aF_2} F_{c_2} + G_{a_2} \begin{bmatrix} \ddot{\theta}_{y2} \\ \ddot{q}_{x2} \\ \ddot{q}_{y2} \\ \ddot{q}_{z2} \end{bmatrix} \qquad (6.28)$$

Note again that the state accelerations appear in the output equations.

In addition, the parameters necessary to determine the mass-loaded assumed mode shapes are required. The beam length, linear density, and bending stiffness are as above. The mass and inertia loading are taken to be the total mass of the elbow and the total y axis inertia (base, armature, and output) for the transverse z mode shapes. The transverse y mode shape is calculated with the total elbow mass and the z axis inertia of the elbow. In both cases the offset is the x displacement of the

Table 6.10: Proximal Link Transverse Mode Shape Parameters

Parameter	Value	Units
L	20	ft
ρA	0.4	slugs/ft
EI	5365100	lb-ft^2
m_t	$m_6 + m_7 + m_8$	slugs
b_x	0.5	ft
I_{yy}	$I_{z6} + I_{z7} + I_{z8}$	slug-ft^2
I_{zz}	I_{y6}	slug-ft^2

elbow center of mass from the end of the link. These parameters are summarized in Table 6.10.

Note that the mass of the distal link and payload (and their inertias) are ignored in the mode shape calculation. Incorporation of the distal link would preclude a numerical solution for the mode shape parameters, as the mode shape would be a trigonometric function of the joint angle. However, the numerical mode shapes will not accurately capture the strain distribution as the elbow angle varies. The standard method for compensating for the fixed mode shapes, and the approach taken in this work, is to "over-parameterize" the model, using more mode shapes than are of interest, in order to provide additional degrees of freedom to capture the system mode shapes.

The parameters for the torsional mode shapes are as listed in Table 6.11. The sum of the elbow x inertias are used as the tip inertia parameter.

Distal Link

The distal link is also modeled as a Bernoulli-Euler beam with finite cross-sectional rotational inertias. As for the proximal link, link model consists of a base offset, bending section, and tip offset. Offsets, inertia properties, and distributed flexibilities are given in Table 6.12.

Table 6.11: Proximal Link Torsional Mode Shape Parameters

Parameter	Value	Units
L	20	ft
ρA	0.4	slugs/ft
GJ	2727400	lb-ft^2
m_t	$m_6 + m_7 + m_8$	slugs
b_y	0	ft
I_{xx}	$I_{x6} + I_{x7} + I_{x8}$	slug-ft^2

The parameters used in the distal link mode shape determination are given in Tables 6.13 and 6.14 for the transverse and torsional modes, respectively. The EOM are

$$M_3 \begin{bmatrix} \ddot{\theta}_{y3} \\ \ddot{q}_{x3} \\ \ddot{q}_{y3} \\ \ddot{q}_{z3} \end{bmatrix} = V_3 + B_{a_3} a_{c_3} \tag{6.29}$$

$$F_{f_2} = H_{F_3} + G_{Fa_3} a_{c_3} + G_{F_3} \begin{bmatrix} \ddot{\theta}_{y3} \\ \ddot{q}_{x3} \\ \ddot{q}_{y3} \\ \ddot{q}_{z3} \end{bmatrix} \tag{6.30}$$

(6.31)

where the free end is not loaded, hence forces are zero.

6.1.4 Model Inversion

The model inversion is carried out following Equations 4.36-4.46, presented in the previous chapter. Note that by choice of rigid body generalized coordinates (coincident with the constrained location), the terms in the inversion become trivial. Specifically, $B_{cc} = I$, $B_{fc} = 0$, $H_c = 0$, $G_{cc} = I$, $G_{cf} = 0$. As a result the inversion equations can be evaluated off line, as inversion of symbolic matrices is not required.

Table 6.12: Distal Link Parameters

Parameter	Value	Units	Description	Reference
a_2	0.5	ft	base offset	[81]
b_2	4.8	ft	tip offset	
L_2	23.0	ft	bending length	
EI_2	3980100.0	lb-ft^2	bending stiffness	
GJ_2	2107300.0	lb-ft^2	torsional stiffness	
ρA_2	0.1522	slug/ft	linear density	
ρI_{x2}	0.0005771695478	slug-ft^2	x linear inertia	
ρI_{y2}	0.0002885847739	slug-ft^2	y linear inertia	
ρI_{z2}	0.0002885847739	slug-ft^2	z linear inertia	

Table 6.13: Distal Link Transverse Mode Shape Parameters

Parameter	Value	Units
L	23	ft
ρA	0.1522	slugs/ft
EI	3980100	lb-ft^2
m_t	m_9	slugs
b_x	4.8	ft
I_{yy}	I_{y9}	slug-ft^2
I_{zz}	I_{z9}	slug-ft^2

6.1.5 Boundary Conditions

The boundary conditions are determined using the axis systems defined in the configuration section. The joint angle is expressed in terms of model DOF. The rotation matrices relating component BC inputs to component BC outputs, for each joint in turn, are found in terms of the joint rotation.

Beginning with the shoulder yaw joint, the rotation around the joint axis, θ_1, is equal to the difference between the output shaft inertial angle θ_{z_1} and the base z

Table 6.14: Distal Link Torsional Mode Shape Parameters

Parameter	Value	Units
L	23	ft
ρA	0.1522	slugs/ft
GJ	2107300	lb-ft^2
m_t	m_9	slugs
b_y	0	ft
I_{xx}	I_{x9}	slug-ft^2

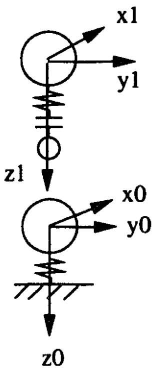

Figure 6-5: Shoulder yaw rotation matrix.

inertial angle θ_{z_0} (Figure 6-5):

$$\theta_1 = \theta_{z_1} - \theta_{z_0} \tag{6.32}$$

The rotation matrix to go from the base frame to the yaw axis frame, in terms of the joint angle, is

$$\begin{bmatrix} \hat{x}_1 \\ \hat{z}_1 \end{bmatrix} = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix} \begin{bmatrix} \hat{x}_0 \\ \hat{y}_0 \end{bmatrix} = R_1 \begin{bmatrix} \hat{x}_0 \\ \hat{y}_0 \end{bmatrix} \tag{6.33}$$

with $c_1 = \cos(\theta_1)$ and $s_1 = \sin(\theta_1)$. The shoulder yaw input angular accelerations consist of the swingout joint output accelerations, rotated up into the shoulder yaw frame:

$$a_{c_1} = R_1 a_{f_0} + \dot{R}_1 r_{f_0} \tag{6.34}$$

where the angular rates for the base swingout joint, r_{f_0}, arise as shown in Section 4.6. The angular rates are functions of the component state (there is no dependence on

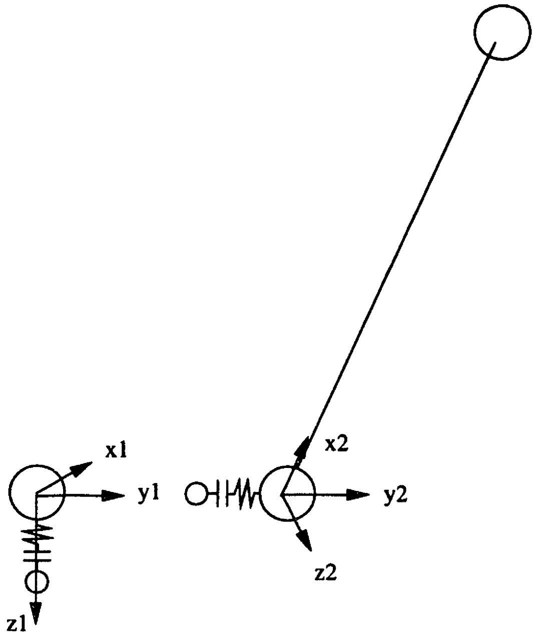

Figure 6-6: Shoulder pitch rotation matrix.

accelerations):

$$r_{f_0} = H_{r_0}(q, \dot{q}) \tag{6.35}$$

The swingout joint input moments are related to the shoulder yaw outputs by

$$F_{c_0} = -R^T F_{f_1} \tag{6.36}$$

where the negative sign arises from "sum of forces is zero" and the transposition of the rotation arises due to reflecting the moments in the 1 frame down into the 0 frame.

The shoulder pitch angle θ_2 is the difference between the proximal link y rotation and the shoulder yaw gearbox y angle(Figure 6-6):

$$\theta_2 = \theta_{y_2} - \theta_{y_1} \tag{6.37}$$

Note that θ_{y_1} is a function of the orientation of the base *and* shoulder yaw angle. Indeed, it cannot be reconstructed from the base angle and the shoulder yaw angle, as it depends on past values of those variables. In fact, θ_{y_1} is the time integral of a rate which is a rotated version of the base rotation rates. The practical consequence is that the rotated base rates must be tracked and integrated to determine the correct rate.

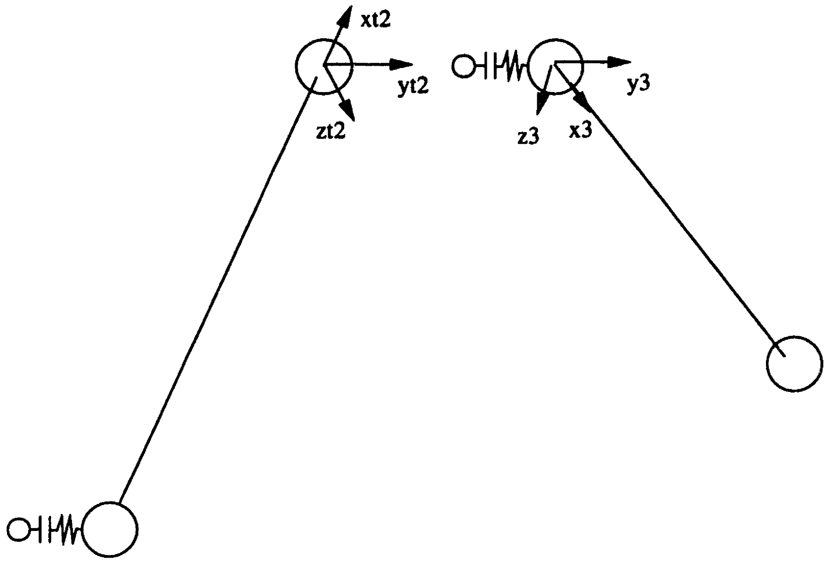

Figure 6-7: Elbow pitch rotation matrix.

Note that θ_{y_1} is not a free DOF. It is one of the constrained DOF. The implication is that the angle must be explicitly tracked with a "placeholder" DOF:

$$\dot{\theta}_{B_2} = \dot{\theta}_{y_1} \tag{6.38}$$

$\dot{\theta}_{y_1}$ is calculated as an element of r_{f_1}.

Given the shoulder pitch angle, the rotation between the shoulder gearbox and proximal link can be found:

$$\begin{bmatrix} \hat{x}_2 \\ \hat{z}_2 \end{bmatrix} = \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \end{bmatrix} = R_2 \begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \end{bmatrix} \tag{6.39}$$

with $c_1 = \cos(\theta_1)$ and $s_1 = \sin(\theta_1)$. The proximal link base inputs and outputs are related to the shoulder gearbox inputs and outputs by

$$a_{c_2} = R_2 a_{f_1} + \dot{R}_2 r_{f_1}, \quad F_{c_1} = -R_2^T F_{f_2} \tag{6.40}$$

where again the angular rates of the shoulder yaw gearbox are required:

$$r_{f_1} = H_{r_1}(q, \dot{q}) \tag{6.41}$$

Note that the vector function H_{r_1} is a function of the base swingout joint rates.

The elbow pitch rotation angle is defined in terms of a frame attached to the end of the proximal link, and the distal link frame (Figure 6-7):

$$\theta_3 = \theta_{y_3} - \theta_{B_3} \tag{6.42}$$

The rotation matrix between the axis systems is

$$\begin{bmatrix} \hat{x}_3 \\ \hat{y}_3 \\ \hat{z}_3 \end{bmatrix} = \begin{bmatrix} c_3 & 0 & s_3 \\ 0 & 1 & 0 \\ -s_3 & 0 & c_3 \end{bmatrix} \begin{bmatrix} \hat{x}_{2,t} \\ \hat{y}_{2,t} \\ \hat{z}_{2,t} \end{bmatrix} = R_3 \begin{bmatrix} \hat{x}_{2,t} \\ \hat{y}_{2,t} \\ \hat{z}_{2,t} \end{bmatrix} \tag{6.43}$$

Additionally, the angular acceleration BCs require the rotation matrix

$$\begin{bmatrix} \hat{x}_3 \\ \hat{z}_3 \end{bmatrix} = \begin{bmatrix} c_3 & s_3 \\ -s_3 & c_3 \end{bmatrix} \begin{bmatrix} \hat{x}_{2,t} \\ \hat{z}_{2,t} \end{bmatrix} = R_{3\theta} \begin{bmatrix} \hat{x}_{2,t} \\ \hat{z}_{2,t} \end{bmatrix} \tag{6.44}$$

The frame attached to the end of the proximal link is, in fact, the frame in which the boundary accelerations of the link are expressed. Again, the y rotation is not in the free DOF, so a placeholder DOF is required to track the joint base rotation:

$$\ddot{\theta}_{B_3} = \ddot{\theta}_{y2t} \tag{6.45}$$

which is integrated along with the free DOF to determine the angle as a function of time.

The force and acceleration relationships between the proximal and distal links are then

$$a_{c_3} = \begin{bmatrix} R_3 & 0 \\ 0 & R_{3\theta} \end{bmatrix} a_{f_2} + \begin{bmatrix} 0 \\ \dot{R}_{3\theta} \end{bmatrix} r_{f_2}, \quad F_{c_2} = \begin{bmatrix} -R_3^T & 0 \\ 0 & -R_{3\theta}^T \end{bmatrix} F_{f_3} \tag{6.46}$$

and

$$r_{f_2} = \begin{bmatrix} 0 \\ H_{r_2}(q, \dot{q}) \end{bmatrix} \tag{6.47}$$

This completes the component boundary conditions.

The component boundary conditions are assembled into the global condition:

$$\begin{bmatrix}F_{c_0}\\ a_{c_1}\\ F_{c_1}\\ a_{c_2}\\ F_{c_2}\\ a_{c_3}\end{bmatrix} = \begin{bmatrix} 0 & -R_1^T & & & & \\ R_1 & 0 & & & & \\ & & 0 & -R_2^T & & \\ & & R_2 & 0 & & \\ & & & & 0 & \begin{bmatrix}-R_3^T & 0\\ 0 & -R_{3\theta}^T\end{bmatrix} \\ & & & & \begin{bmatrix}R_3 & 0\\ 0 & R_{3\theta}\end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix}a_{f0}\\ F_{f1}\\ a_{f1}\\ F_{f2}\\ a_{f2}\\ F_{f3}\end{bmatrix} \tag{6.48}$$

which is written

$$u_{BC} = Ry_{BC} \tag{6.49}$$

Note that u_{BC} are BC inputs to the components. y_{BC} are BC outputs from the system. This completes the boundary condition definition.

6.1.6 System Model

The system model consists of the dynamics of the components coupled by the rotational boundary conditions. The inputs to the system model are the current state (generalized coordinates and their rates), and any external commands. The outputs are the accelerations of the generalized coordinates. The system model can be treated as an Ordinary Differential Equation (ODE). State trajectories can be determined using standard numerical integration techniques[83].

The state consists of the free DOF and the placeholder DOF needed in the evaluation of the dynamics:

$$q_F = \begin{bmatrix} \theta_{x_0} \\ \theta_{y_0} \\ \theta_{z_0} \\ \theta_{A_1} \\ \theta_{z_1} \\ \theta_{A_2} \\ \theta_{y_2} \\ q_{x_2} \\ q_{y_2} \\ q_{z_2} \\ \theta_{A_3} \\ \theta_{y_3} \\ q_{x_3} \\ q_{y_3} \\ q_{z_3} \end{bmatrix}, \quad q_p = \begin{bmatrix} \theta_{B_1} \\ \theta_{B_2} \\ \theta_{x_1} \\ \theta_{y_1} \\ \theta_{x_2} \\ \theta_{z_2} \\ x_3 \\ y_3 \\ z_3 \\ \theta_{x_3} \\ \theta_{z_3} \end{bmatrix} \tag{6.50}$$

The system model returns the generalized accelerations of both sets of coordinates.

The SRMS system model consists of the independent component models, the coupling forces arising from joint stiffnesses, and the boundary condition equation. The accelerations for a given state are determined using the following algorithm.

The notation "+=" is an operator which acts like the C language operator: the right hand side is added to the left hand side, and the result is stored in the left hand side. The expression "a += b" is equivalent to "a = a + b". The operands may be matrix valued. The operator is used in the evaluation of the homogeneous component dynamics: the (possibly time dependent) system matrices in standard form, as presented above, are evaluated first. Then the contribution from the homogeneous accelerations are added in.

Evaluation of the EOM

(i) Solve for the gearbox stiffness torques:

$$\begin{bmatrix} m_{B_1} \\ m_{A_1} \\ m_{S_1} \end{bmatrix} = K \begin{bmatrix} \frac{(n_1-1)^2}{n_1^2} & \frac{n_1-1}{n_1^2} & \frac{1-n_1}{n_1} \\ \frac{n_1-1}{n_1^2} & \frac{1}{n_1^2} & -\frac{1}{n_1} \\ \frac{1-n_1}{n_1} & -\frac{1}{n_1} & 1 \end{bmatrix} \begin{bmatrix} \theta_{z_0} \\ \theta_{A_1} \\ \theta_{z_1} \end{bmatrix} \tag{6.51}$$

$$\begin{bmatrix} m_{B_2} \\ m_{A_2} \\ m_{S_2} \end{bmatrix} = K \begin{bmatrix} \frac{(n_2-1)^2}{n_2^2} & \frac{n_2-1}{n_2^2} & \frac{1-n_2}{n_2} \\ \frac{n_2-1}{n_2^2} & \frac{1}{n_2^2} & -\frac{1}{n_2} \\ \frac{1-n_2}{n_2} & -\frac{1}{n_2} & 1 \end{bmatrix} \begin{bmatrix} \theta_{B_2} \\ \theta_{A_2} \\ \theta_{y_2} \end{bmatrix} \tag{6.52}$$

$$\begin{bmatrix} m_{B_3} \\ m_{A_3} \\ m_{S_3} \end{bmatrix} = K \begin{bmatrix} \frac{(n_3-1)^2}{n_3^2} & \frac{n_3-1}{n_3^2} & \frac{1-n_3}{n_3} \\ \frac{n_3-1}{n_3^2} & \frac{1}{n_3^2} & -\frac{1}{n_3} \\ \frac{1-n_3}{n_3} & -\frac{1}{n_3} & 1 \end{bmatrix} \begin{bmatrix} \theta_{B_3} \\ \theta_{A_3} \\ \theta_{y_3} \end{bmatrix} \tag{6.53}$$

(ii) Solve for the homogeneous portion of the component dynamics:

(a) swingout joint: evaluate M_0, V_0, B_{F0}, and G_{a0}, then compute

$$M_0 \ddot{q}_{H_0} = V_0 \tag{6.54}$$

$$B_{F_0} = M_0^{-1} B_{F_0} \tag{6.55}$$

$$H_{a_0} = G_{a_0} \ddot{q}_{H_0} \tag{6.56}$$

$$G_{aF_0} = G_{a_0} B_{F_0} \tag{6.57}$$

(b) shoulder yaw gearbox: evaluate M_1, V_1, B_{a_1}, B_{F_1}, H_{F_1}, G_{F_1}, G_{Fa_1}, G_{FF_1}, G_{aa_1}, and G_{a_1}, then compute

$$M_1 \ddot{q}_{H_1} = V_1 \tag{6.58}$$

$$B_{a_1} = M_1^{-1} B_{a_1} \tag{6.59}$$

$$B_{F_1} = M_1^{-1} B_{F_1} \tag{6.60}$$

$$H_{F_1} \mathrel{+}= G_{F_1} \ddot{q}_{H_1} \tag{6.61}$$

$$G_{Fa_i} \mathrel{+}= G_{F_1} B_{a_1} \tag{6.62}$$

$$G_{FF_1} \mathrel{+}= G_{F_1} B_{F_1} \tag{6.63}$$

$$H_{a_1} = \ddot{q}_{H_1} \tag{6.64}$$

$$G_{aa_1} \mathrel{+}= G_{a_1} B_{a_1} \tag{6.65}$$

$$G_{aF_1} = G_{a_1} B_{F_1} \tag{6.66}$$

(c) shoulder pitch gearbox:

$$I_4 \ddot{\theta}_{A_2} = u_2 + m_{A_2} \tag{6.67}$$

(d) proximal link: evaluate M_2, V_2, B_{a_2}, B_{F_2}, H_{F_2}, G_{Fa_2}, G_{FF_2}, G_{F_2}, H_{a_2}, G_{aa_2}, G_{aF_2}, and G_{a_2}, then compute:

$$M_2 \ddot{q}_{H_2} = V_2 \tag{6.68}$$

$$B_{a_2} = M_2^{-1} B_{a_2} \tag{6.69}$$

$$B_{F_2} = M_2^{-1} B_{F_2} \tag{6.70}$$

$$H_{F_2} \mathrel{+}= G_{F_2} \ddot{q}_{H_2} \tag{6.71}$$

$$G_{Fa_2} \mathrel{+}= G_{F_2} B_{a_2} \tag{6.72}$$

$$G_{FF_2} \mathrel{+}= G_{F_2} B_{F_2} \tag{6.73}$$

$$H_{a_2} = G_{a_2} \ddot{q}_{H_2} \tag{6.74}$$

$$G_{aa_2} \mathrel{+}= G_{a_2} B_{a_2} \tag{6.75}$$

$$G_{aF_2} \mathrel{+}= G_{a_2} B_{F_2} \tag{6.76}$$

(e) elbow pitch gearbox:

$$I_{y6} \ddot{\theta}_{A_3} = u_3 + m_{A_3} \tag{6.77}$$

(f) distal link: create the time dependent system matrices M_3, V_3, B_{a3}, H_{F3}, G_{Fa3}, and G_{F3}, then calculate the homogeneous dynamics:

$$M_3 \ddot{q}_{H_3} = V_3 \tag{6.78}$$

$$B_{a_3} = M_3^{-1} B_{a_3} \tag{6.79}$$

$$H_{F_3} \mathrel{+}= G_{F_3} \ddot{q}_{H_3} \tag{6.80}$$

$$G_{Fa_3} \mathrel{+}= G_{F_3} B_{a_3} \tag{6.81}$$

(iii) assemble the global output matrices

$$H = \begin{bmatrix} H_{a_0} \\ H_{F_1} \\ H_{a_1} \\ H_{F_2} \\ H_{a_2} \\ H_{F_3} \end{bmatrix}, \quad H_r = \begin{bmatrix} H_{r_0} \\ 0 \\ H_{r_1} \\ 0 \\ H_{r_2} \\ 0 \end{bmatrix} \tag{6.82}$$

$$G = \begin{bmatrix} G_{aF_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & G_{Fa_1} & G_{FF_1} & 0 & 0 & 0 \\ 0 & G_{aa_1} & G_{aF_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & G_{Fa_2} & G_{FF_2} & 0 \\ 0 & 0 & 0 & G_{aa_2} & G_{aF_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & G_{Fa_3} \end{bmatrix} \tag{6.83}$$

where $y_{BC} = H + Gu_{BC}$

(iv) create the rotation matrices R_1, R_2, R_3, and the global rotation matrix:

$$R = \begin{bmatrix} 0 & -R_1^T & & & & \\ R_1 & 0 & & & & \\ & & 0 & -R_2^T & & \\ & & R_2 & 0 & & \\ & & & & 0 & \begin{bmatrix} -R_3^T & 0 \\ 0 & -R_{3\theta}^T \end{bmatrix} \\ & & & & \begin{bmatrix} R_3 & 0 \\ 0 & R_{3\theta} \end{bmatrix} & 0 \end{bmatrix} \tag{6.84}$$

(v) Solve the boundary condition equation $u_{BC} = Ry_{BC}$ for u_{BC}:

$$(I - RG)\,u_{BC} = RH \tag{6.85}$$

(vi) update the component dynamics:

(a) swingout joint:

$$\begin{bmatrix} \ddot{\theta}_{x0} \\ \ddot{\theta}_{y0} \\ \ddot{\theta}_{z0} \end{bmatrix} = \ddot{q}_{H_0} + B_{F_0} F_{c_0} \tag{6.86}$$

(b) shoulder yaw gearbox

$$\begin{bmatrix} \ddot{\theta}_{A1} \\ \ddot{\theta}_{S1} \end{bmatrix} = \ddot{q}_{H_1} + B_{a_1} a_{c_1} + B_{F_1} F_{c_1} \tag{6.87}$$

(c) proximal link

$$\begin{bmatrix} \ddot{\theta}_{y2} \\ \ddot{q}_{x2} \\ \ddot{q}_{y2} \\ \ddot{q}_{z2} \end{bmatrix} = \ddot{q}_{H_2} + B_{a_2} F_{c_2} + B_{F_2} F_{c_2} \tag{6.88}$$

(d) distal link

$$\begin{bmatrix} \ddot{\theta}_{y3} \\ \ddot{q}_{x3} \\ \ddot{q}_{y3} \\ \ddot{q}_{z3} \end{bmatrix} = \ddot{q}_{H_3} + B_{a_3} a_{c_3} \tag{6.89}$$

(vii) calculate the derivatives of the placeholder variables, where the subscript $[a : b]$ denotes the elements in rows a to b of the vector:

(a) shoulder pitch base:

$$\dot{\theta}_{B_2} = \dot{\theta}_{y_1} \tag{6.90}$$

(b) elbow pitch base:

$$\dot{\theta}_{B_3} = \ddot{\theta}_{yt_2} \tag{6.91}$$

(c) proximal link accelerations:

$$\begin{bmatrix} \ddot{\theta}_{x_2} \\ \ddot{\theta}_{z_2} \end{bmatrix} = u_{BC_{[7:8]}} \tag{6.92}$$

(d) distal link accelerations:

$$\begin{bmatrix} \ddot{x}_3 \\ \ddot{y}_3 \\ \ddot{z}_3 \\ \ddot{\theta}_{x3} \\ \ddot{\theta}_{z3} \end{bmatrix} = u_{BC_{[14:18]}} \tag{6.93}$$

This completes the calculation of the system accelerations.

6.1.7 Numerical Solution of the Dynamics

The system dynamics are evaluated using time simulations. As a result the accuracy of any analysis is dependent on the accuracy of the numerical integration procedure, as well as that of the EOM. A useful metric for determining the overall accuracy of the simulation is to track the internal energy of the system, for example:

$$E(t) = \dot{q}^T M \dot{q} + q^T K q \tag{6.94}$$

The non-dissipative nature of the manipulator EOM will require that the energy $E(t)$ remain constant over time. Note that in general the time-integrated system will not be perfectly conservative. If the change in energy becomes substantial over the length of the simulation, a smaller stepsize (or tolerance, when using an adaptive stepsize routine) must be used.

6.1.8 Linear Fractional Control Design Model

The linear fractional control design model is the input data to the control design model in Chapter 3. The major steps in creating the model are linearizing the component models, appending the linearized systems, and adding gearbox stiffness terms. The global rotation matrix becomes the structured uncertainty (Δ) block. The control design approach requires a diagonal uncertainty structure, while the rotation matrix has a block-diagonal structure. An additional step is required to diagonalize the Δ block. The block is factored into a diagonal part and a pair of "pointing" matrices.

These matrices are then folded into the plant to create the control design model. To summarize, creation of the linear fractional model proceeds as follows:

(i) linearize the component models

(ii) append the component models

(iii) incorporate the gearbox stiffnesses

(iv) incorporate the Δ block factorization

The resulting model can be examined at a particular joint configuration by fixing the Δ block.

6.1.9 Component Model Linearization

The component models will be linearized about $q_0 = 0$, $\dot{q}_0 = 0$. The notation $F|_0$ will denote the vector- or matrix-valued quantity, $F(q, \dot{q})$, evaluated at $q = 0, \dot{q} = 0$. For space reasons the dependence on q and $\dot{q}$ will be suppressed. Note that this will zero the placeholder DOF q_p, which only appear as quadratic terms:

$$H_{F_i}|_0 = 0, \quad H_{a_i}|_0 = 0 \tag{6.95}$$

The state size will reduce to the size of the free DOF (19 in this case). The state dependent forces, $V(q, \dot{q})$ will reduce to:

$$V(q_i, \dot{q}_i)_i|_0 = K_i q_i + B_i u_i \tag{6.96}$$

where K is the stiffness matrix. Note also that linearized homogeneous dynamics can be written:

$$\ddot{q}_{H_i} = K_i q_i + B_{u_i} u_i \tag{6.97}$$

which can be substituted into the homogeneous boundary condition expressions, as will be shown.

(i) swingout joint: the state matrices are already linear in the states. Breaking out the free DOF input influence matrix, the linearized component model is

$$\begin{aligned} \ddot{q}_0 &= A_0 q_0 + B_{u_0}\tau_0 + B_{F_0}F_{c_0} \qquad (6.98) \\ a_{f_0} &= C_0 q_0 + D_{au_0}\tau_0 + D_{aF_0}F_{c_0} \qquad (6.99) \end{aligned}$$

with $\tau_0 = \left[\begin{array}{c} -u_1 + m_{B_1} \end{array}\right]$ and

$$\begin{aligned} A_0 &= M_0^{-1}K_0 \qquad (6.100) \\ B_{u_0} &= M_0^{-1}B_0 \qquad (6.101) \\ B_{F_0} &= M_0^{-1}B_{F_0} \qquad (6.102) \\ C_0 &= G_{a_0}A_0 \qquad (6.103) \\ D_{au_0} &= G_{a_0}B_{u_0} \qquad (6.104) \\ D_{aF_0} &= G_{a_0}B_{F_0} \qquad (6.105) \end{aligned}$$

(ii) shoulder yaw gearbox:

$$\begin{aligned} \ddot{q}_1 &= A_1 q_1 + B_{u_1}\tau_1 + B_{a_1}a_{c_1} + B_{F_1}F_{f_1} \qquad (6.106) \\ F_{c_1} &= C_{F_1}q_1 + D_{Fu_1}\tau_1 + D_{Fa_1}a_{c_1} + D_{FF_1}F_{f_1} \qquad (6.107) \\ a_{f_1} &= C_{a_1}q_1 + D_{au_1}\tau_1 + D_{aa_1}a_{c_1} + D_{aF_1}F_{f_1} \qquad (6.108) \end{aligned}$$

with $\tau_1 = \left[\begin{array}{ccc} u_1 + m_{A_1} & m_{S_1} & m_{B_2} \end{array}\right]^T$ and

$$\begin{aligned} A_1 &= (M_1|_0)^{-1}K_1 \qquad (6.109) \\ B_{u_1} &= (M_1|_0)^{-1}B_1 \qquad (6.110) \\ B_{a_1} &= (M_1|_0)^{-1}B_{a_1}|_0 \qquad (6.111) \\ B_{F_1} &= (M_1|_0)^{-1}B_{F_1}|_0 \qquad (6.112) \\ C_{F_1} &= G_{F_1}|_0 A_1 \qquad (6.113) \\ D_{Fu_1} &= G_{F_1}|_0 B_{u_1} \qquad (6.114) \\ D_{Fa_1} &= G_{Fa_1}|_0 + G_{F_1}|_0 B_{a_1} \qquad (6.115) \\ D_{FF_1} &= G_{FF_1}|_0 + G_{F_1}|_0 B_{F_1} \qquad (6.116) \end{aligned}$$

$$C_{a_1} = G_{a_1}|_0 A_1 \tag{6.117}$$

$$D_{au_1} = G_{a_1}|_0 B_{u_1} \tag{6.118}$$

$$D_{aa_1} = G_{aa_1}|_0 + G_{a_1}|_0 B_{a_1} \tag{6.119}$$

$$D_{aF_1} = G_{aF_1}|_0 + G_{a_1}|_0 B_{F_1} \tag{6.120}$$

$$\tag{6.121}$$

(iii) shoulder pitch gearbox:

$$\ddot{\theta}_{A_2} = B_{A_2}\tau_{A_2} \tag{6.122}$$

with $\tau_{A_2} = \left[\begin{array}{c} u_2 + m_{A_2} \end{array}\right]$ and

$$B_{A_2} = \frac{1}{I_4} \tag{6.123}$$

(iv) proximal link:

$$\ddot{q}_2 = A_2 q_2 + B_{u_2}\tau_2 + B_{a_2}a_{c_2} + B_{F_2}F_{f_2} \tag{6.124}$$

$$F_{c_2} = C_{F_2}q_2 + D_{Fu_2}\tau_2 + D_{Fa_2}a_{c_2} + D_{FF_2}F_{f_2} \tag{6.125}$$

$$a_{f_2} = C_{a_2}q_2 + D_{au_2}\tau_2 + D_{aa_2}a_{c_2} + D_{aF_2}F_{f_2} \tag{6.126}$$

with $\tau_2 = \left[\begin{array}{cc} m_{S_2} & -u_3 + m_{B_3} \end{array}\right]^T$ and

$$A_2 = (M_2|_0)^{-1}K_2 \tag{6.127}$$

$$B_{u_2} = (M_2|_0)^{-1}B_2 \tag{6.128}$$

$$B_{a_2} = (M_2|_0)^{-1}B_{a_2}|_0 \tag{6.129}$$

$$B_{F_2} = (M_2|_0)^{-1}B_{F_2}|_0 \tag{6.130}$$

$$C_{F_2} = G_{F_2}|_0 A_2 \tag{6.131}$$

$$D_{Fu_2} = G_{F_2}|_0 B_{u_2} \tag{6.132}$$

$$D_{Fa_2} = G_{Fa_2}|_0 + G_{F_2}|_0 B_{a_2} \tag{6.133}$$

$$D_{FF_2} = G_{FF_2}|_0 + G_{F_2}|_0 B_{F_2} \tag{6.134}$$

$$C_{a_2} = G_{a_2}|_0 A_2 \tag{6.135}$$

$$D_{au_2} = G_{a_2}|_0 B_{u_2} \tag{6.136}$$

$$D_{aa_2} = G_{aa_2}|_0 + G_{a_2}|_0 B_{a_2} \tag{6.137}$$

$$D_{aF_2} = G_{aF_2}|_0 + G_{a_2}|_0 B_{F_2} \tag{6.138}$$

$$\tag{6.139}$$

(v) elbow pitch gearbox:

$$\ddot{\theta}_{A_3} = B_{A_3}\tau_{A_3} \tag{6.140}$$

with $\tau_{A_3} = \left[\begin{array}{c} u_3 + m_{A_3} \end{array}\right]$ and

$$B_{A_2} = \frac{1}{I_{y6}} \tag{6.141}$$

(vi) distal link:

$$\ddot{q}_3 = A_3 q_3 + B_{u_3}\tau_3 + B_{a_3}a_{c_3} \tag{6.142}$$

$$F_{c_3} = C_{F_3} q_3 + D_{Fu_3}\tau_3 + D_{Fa_3}a_{c_3} \tag{6.143}$$

$$\tag{6.144}$$

with $\tau_3 = \left[\begin{array}{cc} m_{S_3} & 0 \end{array}\right]^T$ and

$$A_3 = (M_3|_0)^{-1}K_3 \tag{6.145}$$

$$B_{u_3} = (M_3|_0)^{-1}B_3 \tag{6.146}$$

$$B_{a_3} = (M_3|_0)^{-1}B_{a_3}|_0 \tag{6.147}$$

$$C_{F_3} = G_{F_3}|_0 A_3 \tag{6.148}$$

$$D_{Fu_3} = G_{F_3}|_0 B_{u_3} \tag{6.149}$$

$$D_{Fa_3} = G_{Fa_3}|_0 + G_{F_3}|_0 B_{a_3} \tag{6.150}$$

$$\tag{6.151}$$

6.1.10 System Model Form

For consistency with the control design chapter, the global state will be denoted $x = \left[\begin{array}{cc} q^T & \dot{q}^T \end{array}\right]^T$. The appended dynamics have the form:

$$\dot{x} = Ax + B_u u + B_1 u_{BC} \tag{6.152}$$

$$y_{BC} = Cx + D_{yu}u + D_{y1}u_{BC} \tag{6.153}$$

with the state matrices, partitioned conformably with the component models, having the form:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \\ A_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{A_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{A_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_u = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ B_{u0} & 0 & 0 & 0 & 0 & 0 \\ 0 & B_{u_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & B_{ua_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{u_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{ua_3} & 0 \\ 0 & 0 & 0 & 0 & 0 & B_{u_3} \end{bmatrix}, \; B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ B_{F_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & B_{a_1} & B_{F_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{a_2} & B_{F_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & B_{a_3} \end{bmatrix}$$

$$C = \begin{bmatrix} C_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{F_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{a_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{F_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{a_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{F_3} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_{yu} = \begin{bmatrix} D_{au_0} & 0 & 0 & 0 \\ 0 & D_{Fu_1} & 0 & 0 \\ 0 & D_{au_1} & 0 & 0 \\ 0 & 0 & D_{Fu_2} & 0 \\ 0 & 0 & D_{au_2} & 0 \\ 0 & 0 & 0 & D_{au_3} \end{bmatrix}, \; D_{y_1} = \begin{bmatrix} D_{aF_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & D_{Fa_1} & D_{FF_1} & 0 & 0 & 0 \\ 0 & D_{aa_1} & D_{aF_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & D_{Fa_2} & D_{FF_2} & 0 \\ 0 & 0 & 0 & D_{aa_2} & D_{aF_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{Fa_3} \end{bmatrix}$$

Note that these block-diagonal matrices are simply created by placing the component matrices along the diagonal of the system matrix. In its current form, the component models are independent.

6.1.11 Gearbox Stiffness

The observation and influence matrices are as follows. The shoulder yaw, shoulder pitch, and elbow pitch gearbox rotations are, respectively,

$$\begin{bmatrix} \theta_{z_0} \\ \theta_{A_1} \\ \theta_{z_1} \end{bmatrix}, \begin{bmatrix} \theta_{B_2} \\ \theta_{A_2} \\ \theta_{y_2} \end{bmatrix}, \begin{bmatrix} \theta_{B_3} \\ \theta_{A_3} \\ \theta_{y_3} \end{bmatrix} \tag{6.154}$$

The observation matrices for these angles are

$$C_{g_1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0_{3x19} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix} \tag{6.155}$$

$$C_{g_2} = \begin{bmatrix} -s_1 & c_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0_{3x19} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \end{bmatrix} \tag{6.156}$$

$$C_{g_3} = \left[\begin{array}{cccccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \phi_z'^T & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0_{3x19} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \end{array}\right] \tag{6.157}$$

where 0_{mxn} is a zero matrix of dimension $m \times n$.

Note that the base pitch motion of the shoulder pitch gearbox is a function of the yaw angle. The configuration dependent terms must be broken out into a constant factor times a sub-block of the Δ block. This will be done at a later time.

The influence matrices for the gearbox stiffness torques are found in terms of the global torque inputs:

$$B_{g_1} = B_{u,[1:3]},\ B_{g_2} = B_{u,[4:6]},\ B_{g_3} = B_{u,[7:9]} \tag{6.158}$$

where the subscripted ranges denote columns of B_u. The system matrices are then

$$\dot{x} = A_g x + B_u u + B_1 u_{BC} \tag{6.159}$$

$$y_{BC} = C_g x + D_{yu} u + D_{y1} u_{BC} \tag{6.160}$$

with

$$A_g = A - B_{g_1} K_{g_1} C_{g_1} - B_{g_2} K_{g_2} C_{g_2} - B_{g_3} K_{g_3} C_{g_3} \tag{6.161}$$

$$C_g = C - D_{g_1} K_{g_1} C_{g_1} - D_{g_2} K_{g_2} C_{g_2} - D_{g_3} K_{g_3} C_{g_3} \tag{6.162}$$

with the gearbox stiffness matrices as given in Equations 6.51-6.53.

6.1.12 Control inputs and outputs

The control problem requires sensor and performance outputs, and control inputs. Since the states of the model are created in a physical basis, the state observation matrices can be created based on knowledge of the component states.

The output vectors which will be used for the control problem include the performance sensors, which will be the end-effector rate in end-effector coordinates, and the control sensors which will consist of joint motor encoders, joint output encoders, and end-effector accelerometers. The control inputs will consist of joint torques.

The angles measured by the joint encoders are the difference between the motor angle and the respective base angle:

$$\begin{bmatrix} \theta_{m_1} \\ \theta_{m_2} \\ \theta_{m_2} \end{bmatrix} = \begin{bmatrix} \theta_{A_1} \\ \theta_{A_2} \\ \theta_{A_3} \end{bmatrix} - \begin{bmatrix} \theta_{z_0} \\ \theta_{B_2} \\ \theta_{B_3} \end{bmatrix} \tag{6.163}$$

The corresponding observation matrix is

$$C_m = \left[\begin{array}{cccccccccccccccc} 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ s_1 & -c_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0_{3x19} \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -\phi_z'^T & 1 & 0 & 0 & 0 & 0 & \end{array} \right] \tag{6.164}$$

Similarly, the output encoders measure the difference between the shaft angle and the base:

$$\begin{bmatrix} \theta_{s_1} \\ \theta_{s_2} \\ \theta_{s_2} \end{bmatrix} = \begin{bmatrix} \theta_{y_1} \\ \theta_{y_2} \\ \theta_{y_3} \end{bmatrix} - \begin{bmatrix} \theta_{z_0} \\ \theta_{B_2} \\ \theta_{B_3} \end{bmatrix} \tag{6.165}$$

with the observation matrix:

$$C_s = \left[\begin{array}{cccccccccccccccc} 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ s_1 & -c_1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0_{3x19} \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -\phi_z'^T & 0 & 1 & 0 & 0 & 0 & \end{array} \right] \tag{6.166}$$

The feedthrough matrices for these outputs are zero: $D_{mu} = 0$, $D_{su} = 0$.

The tip acceleration observation matrix is found from the linearized tip acceleration boundary condition for the distal link:

$$C_a = \left[\begin{array}{cccccc|c} 0 & 0 & 0 & 0 & 0 & C_{a_3} & 0_{3x19} \end{array} \right] \tag{6.167}$$

with $C_{a_3} = G_{a_3}|_0 A_3$. Note that the acceleration will have a feedthrough term from the free moment input:

$$D_{au} = D_{au_3} = G_{F_3}|_0 B_{u_3} \tag{6.168}$$

and also from the boundary acceleration input:

$$D_{aa_3} = G_{aa_3} + G_{a_3} B_{a_3} \tag{6.169}$$

The tip acceleration is expressed in the end-effector frame.

The system control inputs enter as a torque couple between each motor base and armature. Defining the pointing matrix T_u from command inputs to system torques,

$$T_u = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.170}$$

The system inputs are given by

$$B_u = B_u T_u, \; D_{yu} = D_{yu} T_u, \; D_{qu} = D_{qu} T_u \tag{6.171}$$

6.1.13 Δ Block Factorization

Now attention is turned to the rotational boundary condition:

$$R = \begin{bmatrix} 0 & -R_1^T & & & & \\ R_1 & 0 & & & & \\ & & 0 & -R_2^T & & \\ & & R_2 & 0 & & \\ & & & & 0 & \begin{bmatrix} -R_3^T & 0 \\ 0 & -R_{3\theta}^T \end{bmatrix} \\ & & & & \begin{bmatrix} R_3 & 0 \\ 0 & R_{3\theta} \end{bmatrix} & 0 \end{bmatrix} \tag{6.172}$$

with

$$R_1 = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix} \tag{6.173}$$

$$R_2 = \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix} \tag{6.174}$$

$$R_3 = \begin{bmatrix} c_3 & 0 & s_3 \\ 0 & 1 & 0 \\ -s_3 & 0 & c_3 \end{bmatrix} \tag{6.175}$$

$$R_{3\theta} = \begin{bmatrix} c_3 & s_3 \\ -s_3 & c_3 \end{bmatrix} \tag{6.176}$$

This is a block diagonal matrix (in fact it is skew symmetric). Note that the third rotation has an element equal to unity. The corresponding elements of the input/output BCs are not functions of orientation. This cannot be represented in the factorization for Δ, and indeed is not desired to be included since it increases the input/output order. Since it is a known function, it can be solved for off-line and included in the system matrix. Splitting the BCs into known portions, u_k and y_k, and leaving the unknown portions in u_{BC} and y_{BC}, the known portions are solved for:

$$\dot{x} = A_g x + B_u u + B_1 u_{BC} + B_k u_k \tag{6.177}$$

$$y_{BC} = C_g x + D_{yu} u + D_{y1} u_{BC} + D_{yk} u_k \tag{6.178}$$

$$y_k = C_k x + D_{ku} u + D_{k1} u_{BC} + D_{kk} u_k \tag{6.179}$$

With $u_k = y_k$, the system dynamics become:

$$\dot{x} = Ax + B_u u + B_p u_{BC} \tag{6.180}$$

$$y_{BC} = C_q x + D_{qu} u + D_{qp} u_{BC} \tag{6.181}$$

and

$$A = A_g + B_k (I - D_{kk})^{-1} C_k \tag{6.182}$$

$$B_u = B_u + B_k (I - D_{kk})^{-1} D_{ku} \tag{6.183}$$

$$B_p = (B_1 + B_k (I - D_{kk})^{-1} D_{k1}) \tag{6.184}$$

$$C_q = C_g + D_{yk} (I - D_{kk})^{-1} C_k \tag{6.185}$$

$$D_{qu} = D_{yu} + D_{yk} (I - D_{kk})^{-1} D_{ku} \tag{6.186}$$

$$D_{qp} = D_{y1} + D_{yk} (I - D_{kk})^{-1} D_{k1} \tag{6.187}$$

Next the time-varying elements of the Δ block are captured.

The skew symmetric block matrix

$$\begin{bmatrix} 0 & -R^T \\ R & 0 \end{bmatrix}, \; R = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{6.188}$$

can be factored as

$$S_l \begin{bmatrix} \cos(\theta) I_{4x4} & \\ & \sin(\theta) I_{4x4} \end{bmatrix} S_r \tag{6.189}$$

with

$$S_l = \begin{bmatrix} 0 & -I & 0 & J \\ I & 0 & J & 0 \end{bmatrix}, \; S_r = \begin{bmatrix} I & 0 \\ 0 & I \\ I & 0 \\ 0 & I \end{bmatrix} \tag{6.190}$$

where the matrix $J = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, and the zero matrices are 2×2. Using the square symmetric transformation

$$F = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{6.191}$$

the lower left block of the rotation matrix can be put into the block form of Equation 6.188:

$$\begin{bmatrix} 0 & -R_3^T & 0 & 0 \\ R_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -R_{3\theta}^T \\ 0 & 0 & R_3 & 0 \end{bmatrix} = F \begin{bmatrix} 0 & 0 & -R_3^T & 0 \\ 0 & 0 & 0 & -R_{3\theta}^T \\ R_3 & 0 & 0 & 0 \\ 0 & R_3 & 0 & 0 \end{bmatrix} F \tag{6.192}$$

Since the F transformation is square, it does not change the block size.

Using Equation 6.188 as a building block, the Δ block can be factored into

$$S_L \begin{bmatrix} \cos(\theta_1)I_{4x4} & & & & & \\ & \sin(\theta_1)I_{4x4} & & & & \\ & & \cos(\theta_2)I_{4x4} & & & \\ & & & \sin(\theta_2)I_{4x4} & & \\ & & & & \cos(\theta_3)I_{4x4} & \\ & & & & & \sin(\theta_3)I_{4x4} \end{bmatrix} S_R \tag{6.193}$$

with

$$S_L = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & F \end{bmatrix} \begin{bmatrix} S_l & 0 & 0 \\ 0 & S_l & 0 \\ 0 & 0 & \begin{bmatrix} S_l & 0 \\ 0 & S_l \end{bmatrix} \end{bmatrix} \tag{6.194}$$

and

$$S_R = \begin{bmatrix} S_r & 0 & C \\ 0 & S_r & 0 \\ 0 & 0 & \begin{bmatrix} S_r & 0 \\ 0 & S_r \end{bmatrix} \end{bmatrix} \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & F \end{bmatrix} \tag{6.195}$$

Note that the diagonalization increases the size of the Δ block.

Noting that

$$u_{BC} = Ry_{BC} = S_L \Delta S_R y_{BC} \tag{6.196}$$

the boundary conditions are redefined to include the pointing matrices:

$$\hat{y}_{BC} = S_R y_{BC},\ u_{BC} = S_L \hat{u}_{BC} \tag{6.197}$$

The pointing matrices are then folded into the system matrices:

$$\dot{x} = Ax + B_u u + B_p S_L \hat{u}_{BC} \tag{6.198}$$

$$\hat{y}_{BC} = S_R C_q x + S_R D_{qu} u + S_R D_{qp} S_L \hat{u}_{BC} \tag{6.199}$$

The new BC inputs and outputs, $\hat{u}_{BC}$ and $\hat{y}_{BC}$, become the parameter feedback channels of the LFT system model. In the notation of Chapter 3:

$$\dot{x} = Ax + B_u u + B_p p \tag{6.200}$$

$$q = C_q x + D_{qu} u + D_{qp} p \tag{6.201}$$

Note that the performance outputs z, sensor outputs y, and disturbance inputs w, have not been defined. These channels are defined based on the desired performance goals. Once they are known, the appropriate observation and feedthrough matrices can be determined based on the linear system modes. That is, the system model can be developed without carrying through the performance and sensor channels.

This concludes the SRMS model development.

6.2 Linear Fractional Model Comparison

In this section, the Linear Fractional control design models derived from the Lagrangian truth model, and from the component model, are compared to determine relative modeling effort, run-time computation requirements, and system/uncertainty realization order. To this end, an LFT realization of the Lagrange model must be created.

Following the standard practice in the flight controls/robust control literature, the LFT model is realized by creating polynomial fits to parameter-varying elements of the system matrix S:

$$s_{ij}(\theta) \sum_{k=0}^{n} a_{ijk}\theta^k \tag{6.202}$$

with s_{ij} the i, j^{th} element of the system matrix. The scheduling parameters are taken to be the joint angles. The system matrices are sampled across the parameter space, and Least-Squares polynomial fits to each element in the system are created. The order of the polynomial fit is varied until the error is less than a tolerance.

Once polynomial representations in the joint angles are created for each element, a heuristic reduction procedure is carried out. The polynomial representations are compared to identify elements which have the same trajectory. Care must be taken to compare only the elements which multiply the same states:

$$\dot{x}_i = s_{ij} x_j = \sum_{k=0}^{n} a_{ijk}\theta^k x_j \tag{6.203}$$

because the state x_j will be carried through the uncertainty realization. That is, only elements in the same column j are compared.

Finally, the polynomial fit is realized as a LFT. The polynomial consists of a constant term which becomes an element of the nominal system matrix, and an observation matrix on the uncertainty:

$$\dot{x}_i = a_{ij0}x_j + \begin{bmatrix} a_{ij1} & a_{ij2} & \cdots \end{bmatrix} \begin{bmatrix} \theta \\ \theta^2 \\ \vdots \end{bmatrix} x_j = a_{ij0}x_j + c_{ij}^T v_{ij} \tag{6.204}$$

with c_{ij} the vector of coefficients and v_{ij} the vector of polynomials times x_j. The uncertainty itself is realized as a repeated scalar feedback through an augmented system feedthrough matrix:

$$u_{ij} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdots \end{bmatrix} x_j + \begin{bmatrix} 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & \\ 0 & 1 & 0 & \\ \cdots & & & \end{bmatrix} v_{ij} \tag{6.205}$$

with the driving term in the above equation a feedback term:

$$v_{ij} = \theta u_{ij} \tag{6.206}$$

The constant terms in Equation 6.205 are folded into the system matrix, as an observation matrix on the states and a feedthrough matrix. The polynomial coefficients c_{ij} are folded in as an influence matrix. The feedback terms in θ are collected into a single block, with θ repeated along the diagonal, which becomes the structured uncertainty Δ block.

The polynomial realization is carried out on the 2D Lagrange SRMS model. The fit tolerance is set to 1%. This leads to polynomial terms up to 5^{th} order. The system observation matrix is independent of θ, so that the only elements which contribute to the uncertainty realization are from the system A and B matrices.

A comparison of the truth model and the component model is shown in Table 6.15. The first column showing development time represents a gauge of modeling effort. The

Table 6.15: Analytical and numerical efficiency metrics for the planar Lagrange and component models.

	Development time[mo]	Code Size [Kb]	Run time [sec/sec]	Delta block size
Lagrange	12	40	10	191
Component	3	17	2	18

next, code size, gives relative compactness of the two forms of the EOM. Run time, given as CPU time (on a Sparc 20) needed to get one second of simulated time, shows the run-time computational effort of the two forms. Note that the time needed will vary as a function of the forces exerted on the manipulator model. Delta block size gives the number of additional input/output channels needed to capture parametric variations. The table shows that the component form of the EOM is easier to develop, more compact, and runs faster.

Note also that the run time for the component model approaches real time. This capability would enable an observer-based controller[27, 24].

6.3 Model Comparison

The component model is compared to the Lagrangian model on the basis of frequency domain metrics and time domain metrics.

6.3.1 Frequency Domain Comparison

Frequency domain metrics consist of natural mode frequency errors between the component model and the Lagrangian model, and root mean square (rms) errors between the transfer functions from specified torque inputs to sensor outputs. The latter metric accounts for mode shape errors in the Lagrangian model. The models are linearized about a configuration of $(0, 45°, -60°)$, corresponding to a "low hover" position of the payload over the payload bay.

Table 6.16: Normal mode frequencies for the component model, compared to the Lagrangian truth model.

Component Freq [Hz]	Lagrangian Freq [Hz]	Error %
0.6126	0.6126	$3.5510e-08$
1.0833	1.0833	$5.4701e-08$
1.5981	1.5981	$6.0242e-08$
2.0079	2.0079	$3.9907e-09$
2.6868	2.6868	$1.0650e-09$
3.0658	3.0658	$2.6584e-08$
4.7254	4.7254	$2.6225e-08$
16.3091	16.3091	$3.0375e-08$
16.6829	16.6829	$4.6683e-09$
33.2310	33.2310	$6.6544e-07$
53.0269	53.0269	$3.8667e-08$
54.4273	54.4273	$1.6630e-08$
71.5410	71.5410	$9.9241e-09$
76.6820	76.6820	$8.5766e-09$
146.4416	146.4416	$3.8786e-11$
221.0153	221.0153	$6.7193e-11$

The component model normal mode frequencies are compared to the Lagrangian truth model frequencies in Table 6.16. The component model frequencies are given in the first column, the Lagrangian frequencies are given in the second, and the difference in frequencies, given as a percentage difference to the Lagrange model, are shown in the last column. Note that the largest percentage error is on the order of 10^{-7}. This is to be expected as the physical parameters for both models are identical.

The accuracy of the component model mode shapes, compared to the Lagrangian model, are evaluated by calculating the *rms* error between the transfer functions

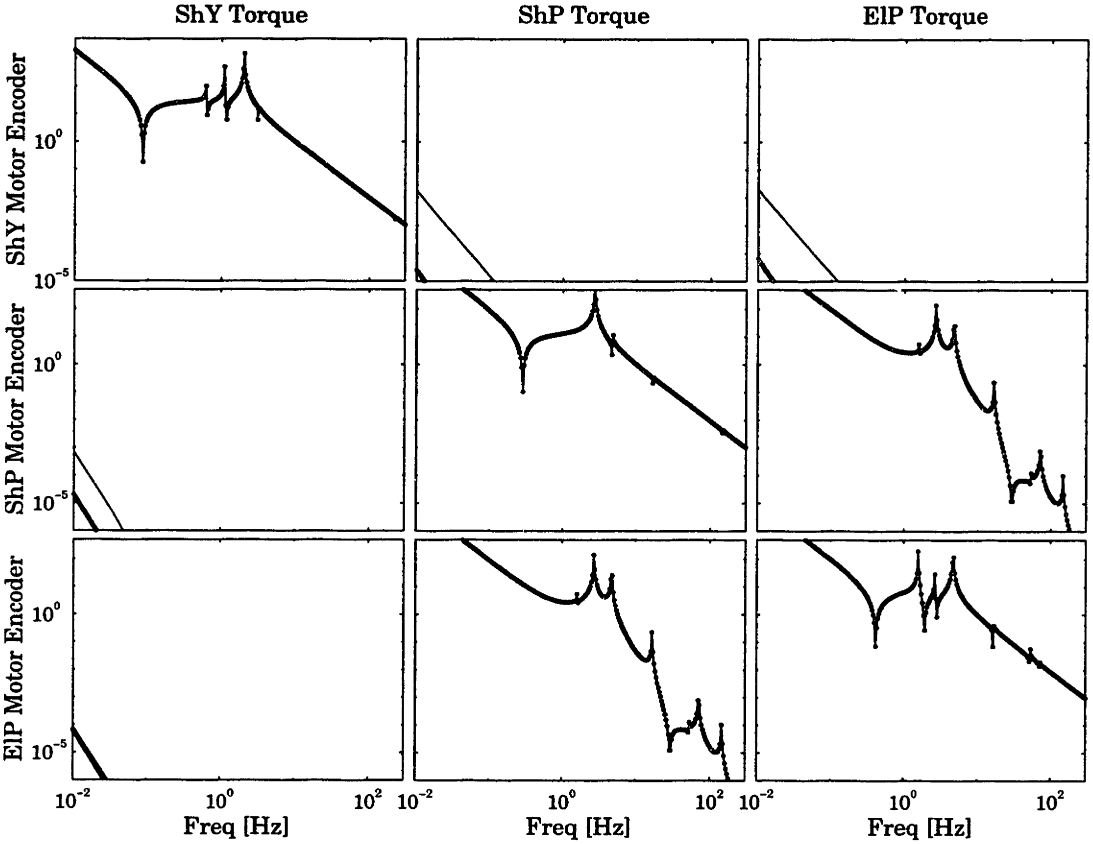

Figure 6-8: Transfer functions from joint torques to motor positions for the Lagrangian truth model (solid) and the component model (dotted).

from the shoulder yaw motor (ShY), shoulder pitch motor (ShP), and elbow pitch motor (ElP) to: ShY, ShP, and ElP motor position (the *motor encoder* outputs), ShY, ShP, and ElP output shaft position (*output encoders*), and end-effector X, Y, and Z accelerations (in the end-effector frame). These transfer functions will be presented first. Then the *rms* errors will be examined.

The transfer functions from motor inputs to motor encoders are given in Figure 6-8. The first column contains transfer functions from the ShY motor input, the second column shows the transfer functions from the ShP input, and the third shows transfer functions from the ElP input. Likewise the first row shows transfer functions to the ShY encoder, the second row transfer functions are to the ShP encoder, and the third row is to the ElP encoder. In each plot, the transfer function is shown with a solid line for the Lagrange truth model, and as a dotted line for the component model.

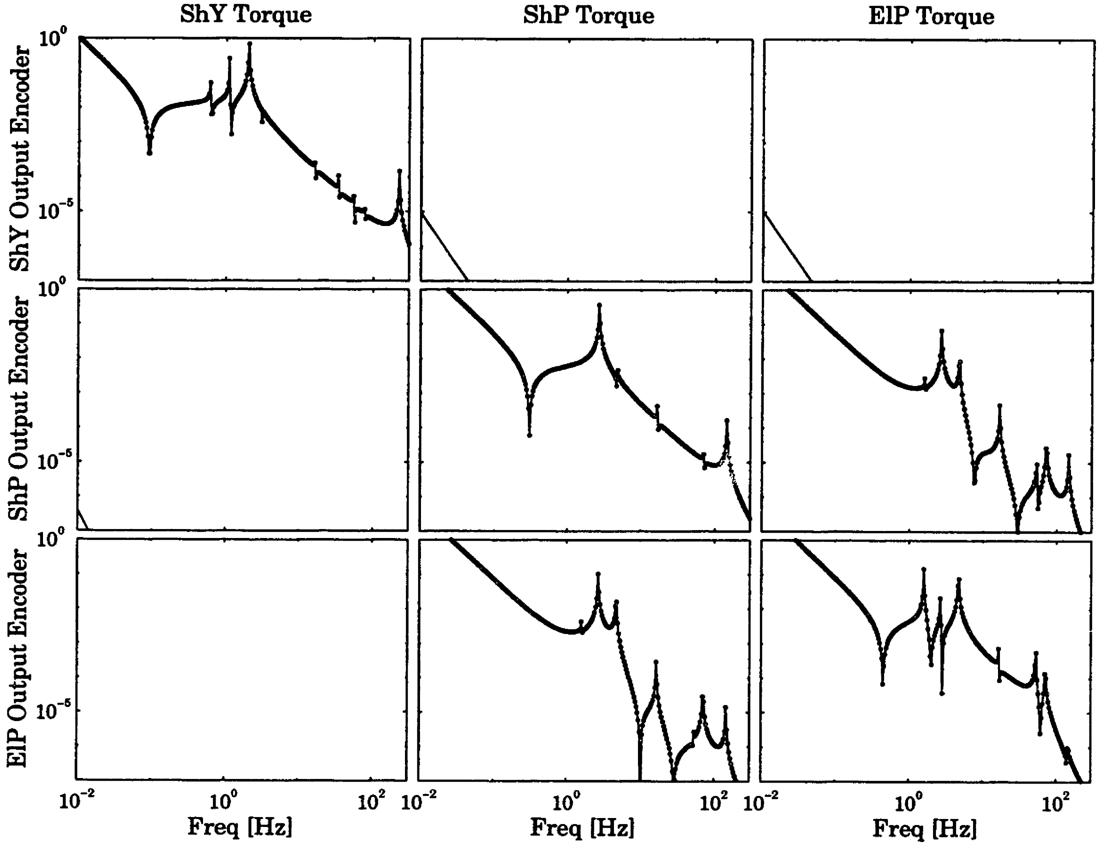

Figure 6-9: Transfer functions from joint torques to output shaft positions for the Lagrangian truth model (solid) and the component model (dotted).

Note that the pitch and yaw axes nearly decouple, as shown by the difference in magnitude from the ShY torque to ShY encoder, and to ShP and ElP encoders. Similarly the ShP and and ElP encoders are excited at a low level by the ShY torque.

Note also that the high magnitude transfer functions for the component model essentially overlay the truth model. This indicates a high degree of correspondence between the mode shapes. The yaw to pitch and pitch to yaw transfer functions, on the other hand, show a large relative error. However, these “off-axis” transfer functions are at an extremely low magnitude, hence the absolute error is actually very small.

The transfer functions from motor inputs to output encoders are given in Figure 6-9. Again, the inputs are ranged column-wise from shoulder yaw outwards, and the outputs are ranged row-wise. Again note that the yaw to yaw, and pitch to

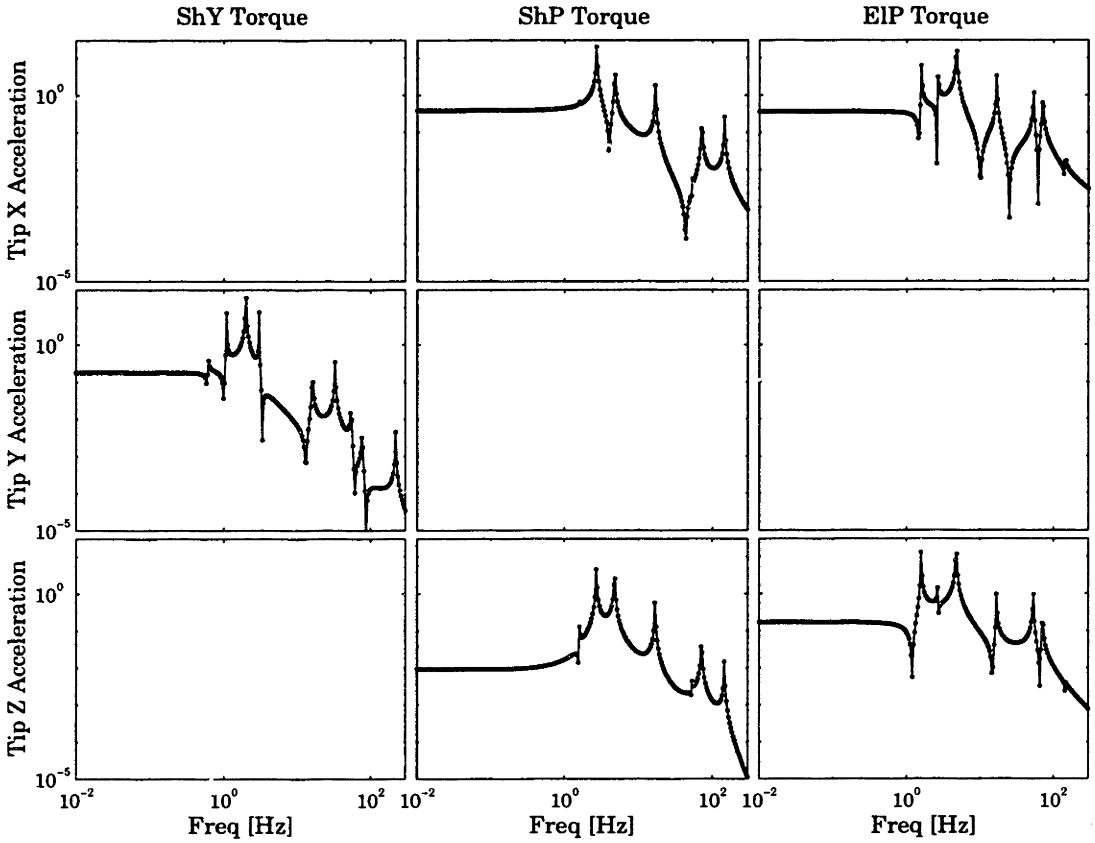

Figure 6-10: Transfer functions from joint torques to end-effector acceleration for the Lagrangian truth model (solid) and the component model (dotted).

pitch transfer functions overlay. The off-axis transfer functions are again at a low magnitude.

The transfer functions from motor inputs to tip accelerations are given in Figure 6-10. The inputs are ranged column-wise, and the X, Y, and Z accelerations (in the end-effector frame) are given row-wise. These outputs are particularly interesting as they will be used as performance and feedback sensors in the control design stage. Once again the component model shows a high degree of fidelity to the Lagrangian mode.

The *rms* transfer function errors are given in Table 6.17. The largest error is on the order of $1e-3$, bearing out the fidelity of the match which was evident from the transfer function plots.

This concludes the demonstration that the linearized component model is a high

Table 6.17: Component model *rms* transfer function error compared to Lagrangian truth model.

	ShY torque	ShP torque	ElP torque
ShY motor	$4.2704e-4$	$2.1919e-3$	$2.5113e-3$
ShP motor	$2.2319e-7$	$1.1900e-6$	$1.3633e-6$
ElP motor	$8.0407e-5$	$8.3898e-4$	$8.9865e-4$
ShY output	$4.3628e-8$	$4.5522e-7$	$4.8760e-7$
ShP output	$8.5501e-6$	$3.9383e-5$	$1.3261e-4$
ElP output	$6.7857e-9$	$3.1209e-8$	$1.0267e-7$
X accel	$3.8107e-10$	$4.5983e-7$	$4.5746e-6$
Y accel	$8.0820e-6$	$4.3529e-10$	$4.3728e-10$
Z accel	$1.6578e-10$	$3.6368e-7$	$8.5835e-6$

fidelity match to the Lagrangian truth model.

6.3.2 Time Domain

The nonlinear component model is compared to the Lagrange model on the basis of time responses from joint motor encoders, output shaft encoders, and end-effector acceleration. These sensors are chosen because they are used as feedback sensors in the control design section. The system is initialized at rest at the hover position. The models are then excited with impulsive torques to the three joint motors. The time histories are calculated using a Runge-Kutta fourth-fifth order integration routine with adaptive stepsize. The response at .01 second intervals for a 10 second run are recorded.

The following plots show the resulting time histories. The Lagrange truth model is plotted as a solid line. The component model is over-plotted using dots. Again, the component model nearly overlays the Lagrange model. Note that the nonlinear models overlay, confirming the accuracy of the component modeling approach.

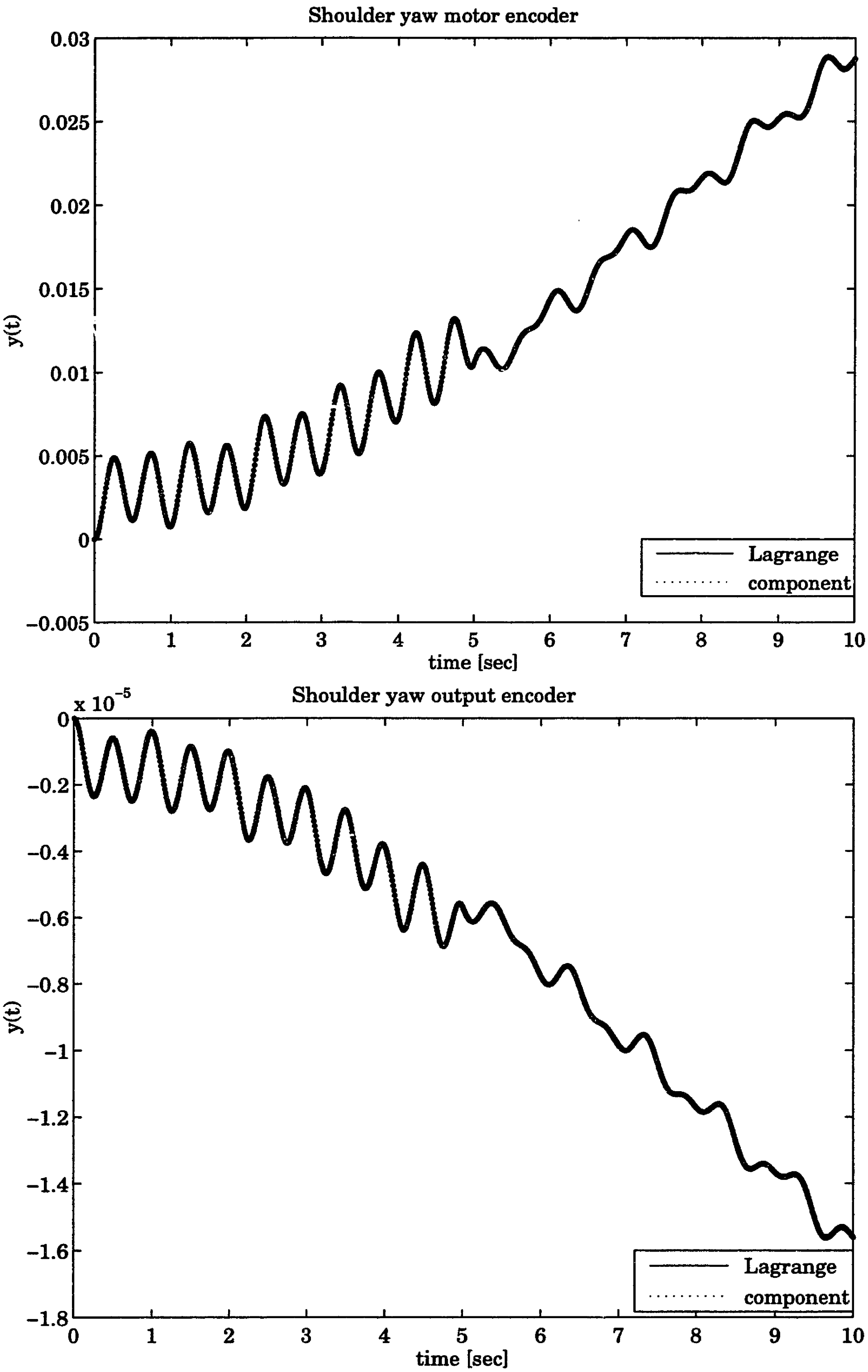

Figure 6-11: Time response of the shoulder yaw joint encoders to simultaneous ShY, ShP, and ElP torques, for the Lagrange and component models: motor (top) and output shaft (bottom).

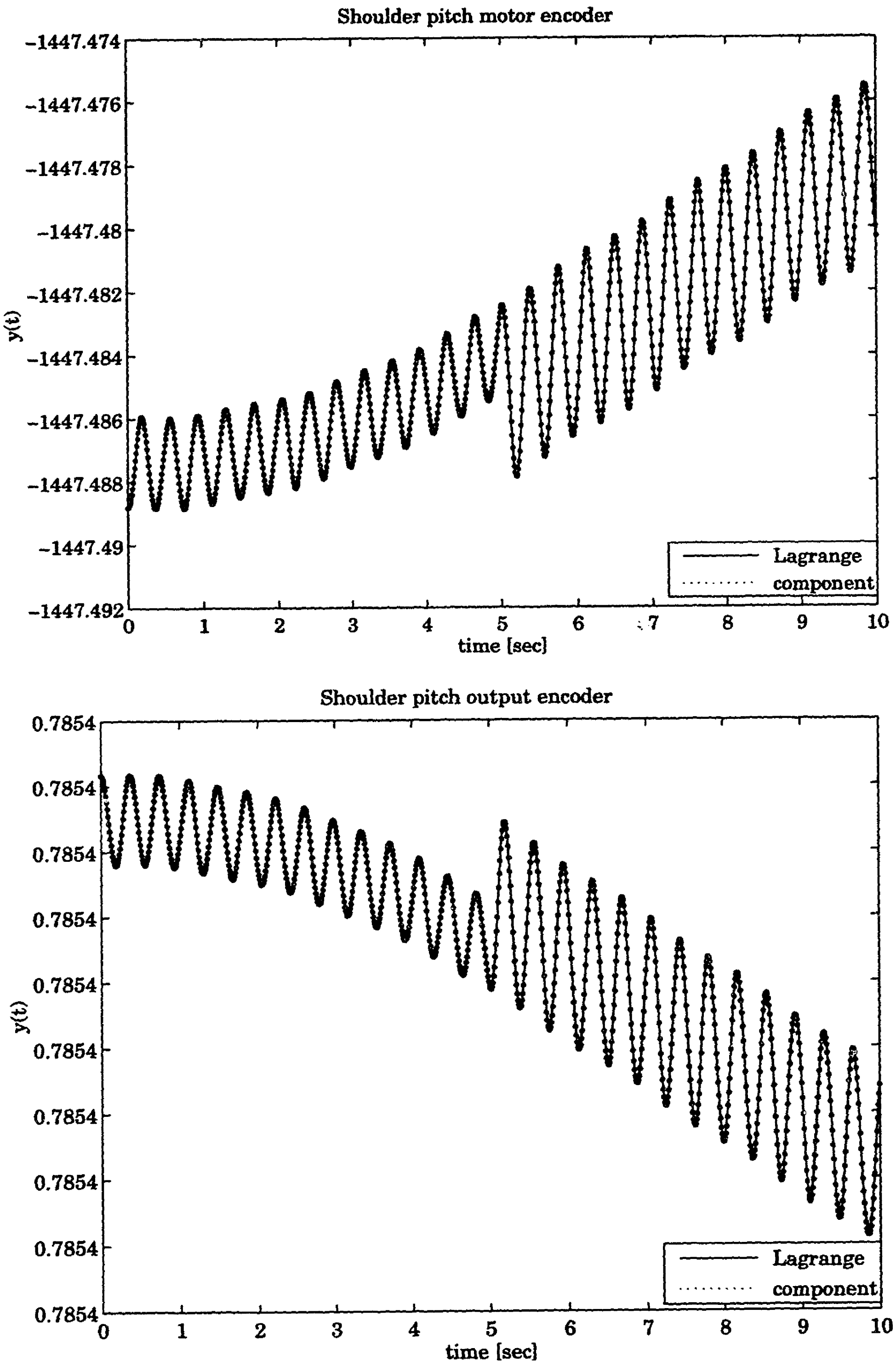

Figure 6-12: Time response of the shoulder pitch joint encoders to simultaneous ShY, ShP, and ElP torques: motor (top) and output shaft (bottom).

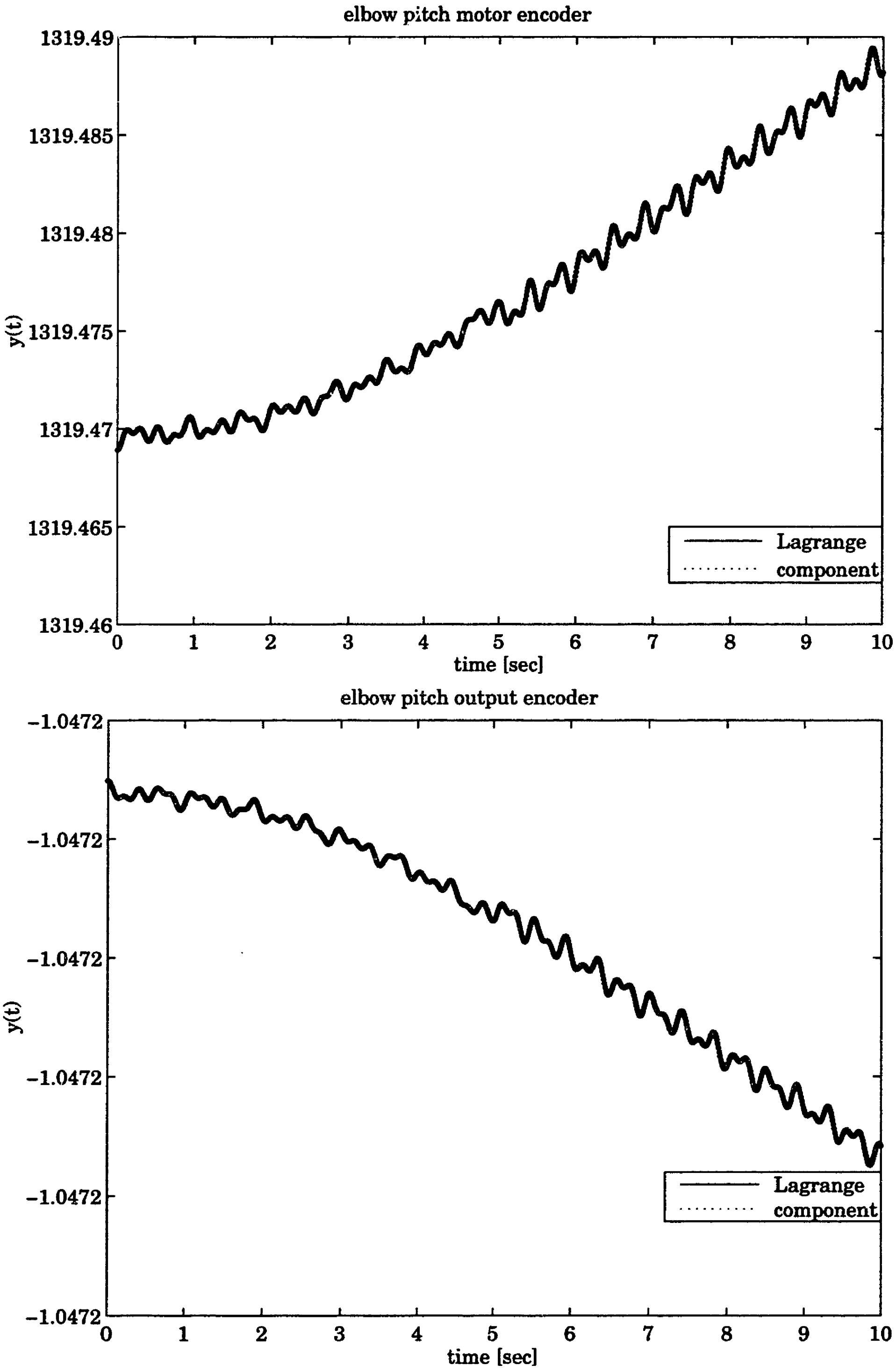

Figure 6-13: Time response of the elbow pitch joint encoders to simultaneous ShY, ShP, and ElP torques: motor (top) and output shaft (bottom) .

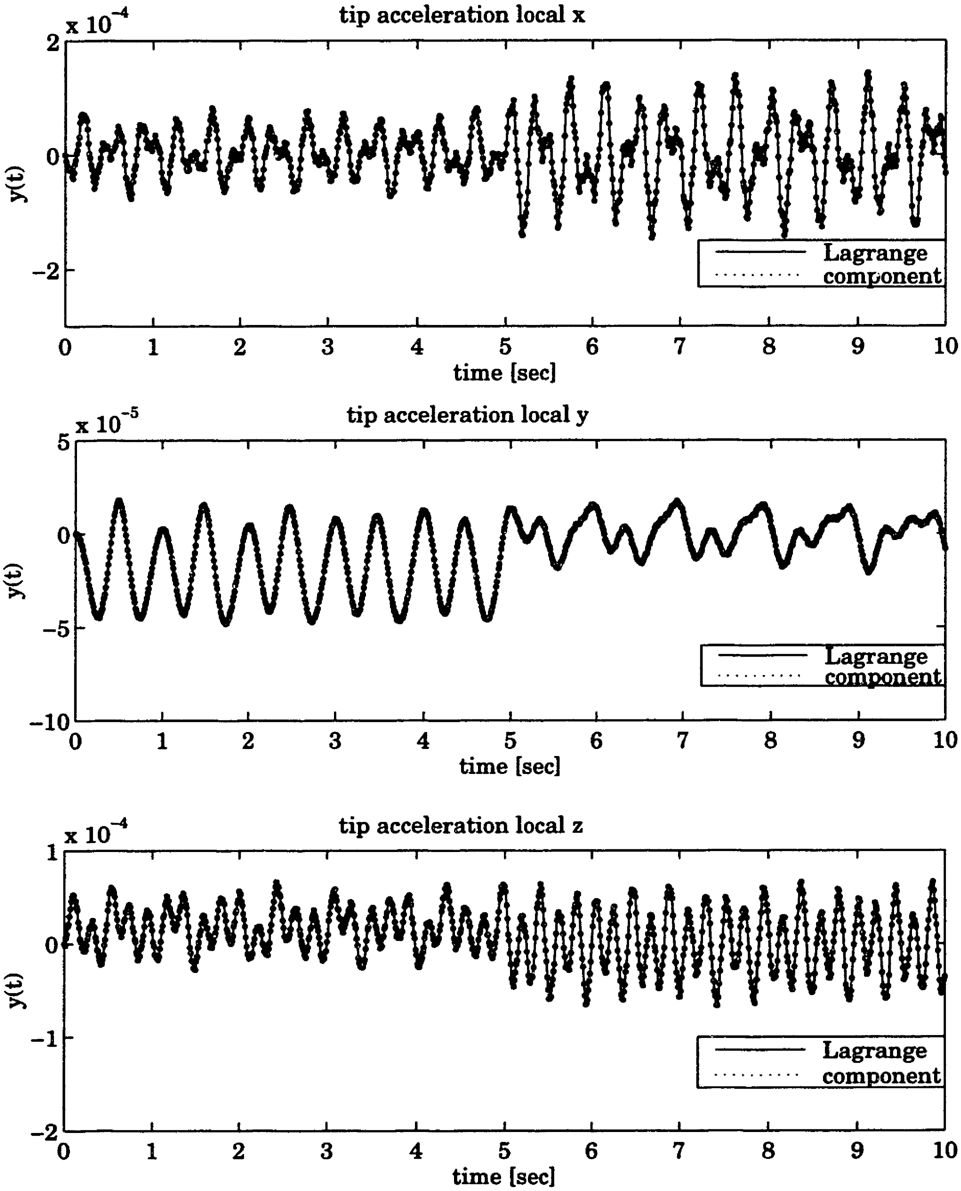

Figure 6-14: Time response of the tip accelerometers to simultaneous ShY, ShP, and ElP torques: X (top), Y (middle), and Z (bottom).

6.4 Control Design

The LMI gain scheduled $\mathcal{H}_\infty$ design equations of Chapter 3 are applied to create a gain-scheduled controller for the Linear Fractional SRMS model.

6.4.1 Design Model

The design model is created from the 2D SRMS model. The design objective is to provide accurate positioning of the end effector over the entire workspace. Positioning will be expressed in terms of end effector rate in end effector coordinates. This corresponds to the SRMS approach in which operator end-effector commands are resolved into joint rates, and joint rate servos achieve the commanded rates. The control requirements for the closed loop system will be stated in the standard Mixed Sensitivity framework[17]. The performance requirements are:

(i) Steady-state error in response to a step command less than 1%.

(ii) Closed loop damping greater than 30%.

In addition, the following robustness requirements are imposed:

(i) Closed loop magnitude less than .01 at $50Hz$ to provide robustness to unstructured uncertainty reflected at the plant input.

(ii) Compensator rolloff after $10Hz$ to avoid control spillover to higher frequency modes.

These goals are to be met over the entire elbow angle workspace of −2° to 160°. The requirements will be expressed in the frequency domain.

The error requirement is enforced with a constraint on the closed loop sensitivity $S(\jmath\omega)$, via a frequency weight $W_s(\jmath\omega)$:

$$\bar{\sigma}\left(S(\jmath\omega)\right) < \left|W_s(\jmath\omega)^{-1}\right| \tag{6.207}$$

The weight has a magnitude of 100 at low frequencies, to ensure that the sensitivity is below .01 whenever the $\mathcal{H}_\infty$ performance is below 1. The weight rolls off to below unity. The frequency of the rolloff is a design variable. Closed loop damping and rolloff are enforced with a weight, W_c, on the complementary sensitivity $C(\jmath\omega)$:

$$\bar{\sigma}\left(C(\jmath\omega)\right) < \left|W_c(\jmath\omega).^{-1}\right| \tag{6.208}$$

The magnitude of the weight is 2 at low frequency, to keep overshoot below 100%, corresponding to 30% damping. Rolloff of the closed loop system is enforced with a second order rolloff on $W_c(s)$ at $5Hz$.

Additional design variables take the form of weights on the control effort $K(s)S(s)$, where $K(s)$ is the compensator frequency response, and sensor noise. The control effort weight rolls up at $10Hz$ to penalize high-frequency control. The overall magnitude of the weight is a design variable. The sensor noise is white. Its magnitude is a design variable.

The control weights are shown in Figure 6-15. To summarize the design variables, the sensitivity rollup frequency, control effort, and sensor noise are adjustable to reach the design goals.

6.4.2 Control Design

The following transformation is made to the uncertainty to allow the workspace to be varied. The range of angles for the manipulator is specified in terms of a maximum angle, $\bar{\theta}$ and a minimum angle $\underline{\theta}$. The Δ blocks at the extremal values of the parameter are determined from the trigonometric functions:

$$\bar{s} = \sin(\bar{\theta}), \ \underline{s} = \sin(\underline{\theta}) \tag{6.209}$$

and

$$\bar{c} = \cos(\bar{\theta}), \ \underline{c} = \cos(\underline{\theta}) \tag{6.210}$$

Then the nominal values of the parameter are determined from the geometric mean:

$$s_0 = \frac{1}{2}(\bar{s} - \underline{s}), \ c_0 = \frac{1}{2}(\bar{(}c) - \underline{c}) \tag{6.211}$$

and the maximum change in each parameter is known:

$$\delta s = \bar{s} - s_0 = s_0 - \underline{s}, \ \delta c = \bar{(}c) - c_0 = c_0 - \underline{c} \tag{6.212}$$

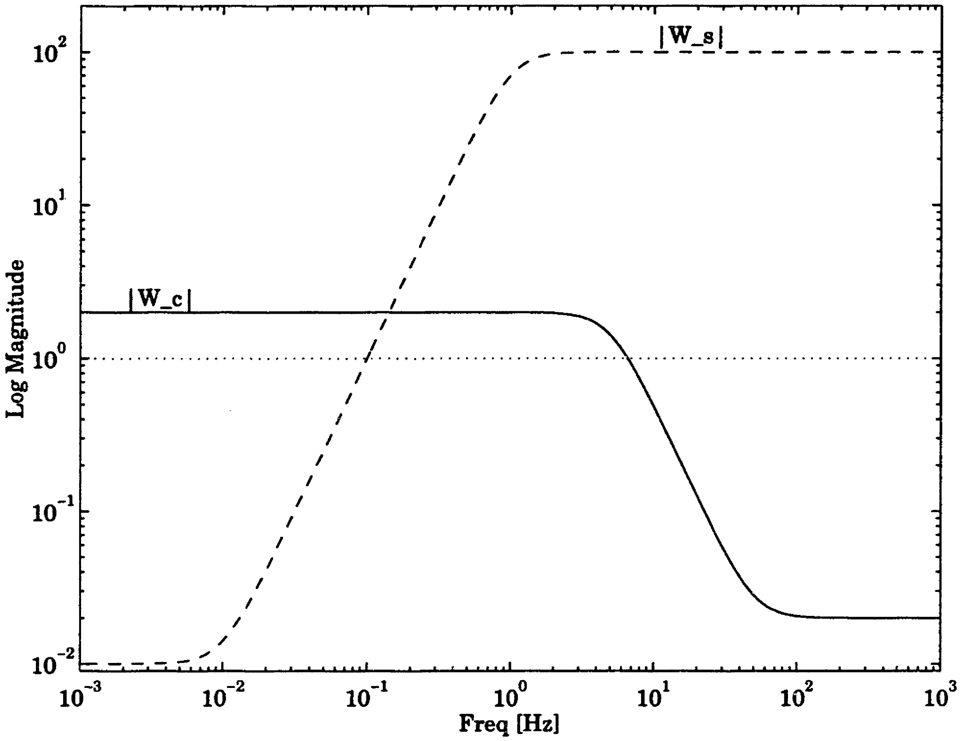

Figure 6-15: Sensitivity and control weights used for the SRMS $\mathcal{H}_\infty$ design sample problem.

At this point the nominal system can be defined as an LFT of the Δ block, evaluated at the nominal value of the parameters, with the system. Then δs and δc are scaled into the uncertainty channels so that the Δ block ranges from -1 to 1.

Next the design weights are appended to the plant inputs and outputs. The feasibility of the gain-scheduled controller can now be evaluated using the LMIs of Equations 3.86-3.89, from Chapter 3. Given a feasible solution, the procedure in Section 3.9 can be followed to find the controller.

This procedure can be followed for the 3D, 19 DOF SRMS model. The LMI conditions have 2830 decision variables. Unfortunately, the memory requirements of the LMI solution algorithm are relatively high. The feasibility solver in LMI-Lab©, 'feasp.m', is unable to determine the feasibility of the resulting system of equations on a Sparc 20 with 192M of memory (64M real, 128M swap space). The projection algorithm used to determine feasibility of the LMI conditions uses a relatively large amount of memory. As a result, the design equations cannot be solved for the full, 3D manipulator.

However, the memory requirements for the 2D, planar arm are tractable (there are 600 decision variables for the particular design weights chosen). The design solution is found for a range of 0.1° around the nominal configuration. However, the elbow angle workspace over which the LMI can be solved is less than 10°. That is, the Small Gain conditions are too conservative to allow a controller with a 10° workspace to be feasible. It is reiterated that conservatism arises from the Small Gain condition which does not use phase information, from the non-physical infinite time rates of change of the scheduling variables allowed by the math, and from the lack of a mathematical description of the functional connection between the scheduling variables and the system states.

Note, also, that the controller computation step requires another LMI solution. The decision variables are the elements of the controller system matrix. This is a full block, without the symmetry of the feasibility matrix solutions. The number of decision variables is $(n_k + r_k + n_{o_2}) * (n_k + r_k + n_{i_2})$. For the 2D arm with $n_k = 18$, $r_k = 18$, $n_{o_2} = 6$, and $n_{i_2} = 2$, there are 1596 decision variables which again is too large to be solved for. Note in addition that the controller computation LMI involves the controller states, not simply the controller inputs and outputs. The implication is that the controller cannot (necessarily) be re-parameterized as a canonical form. A canonical form such as a tridiagonal A_k matrix would reduce the number of decision variables, but may not lie within the feasible solution region.

6.5 Conclusions

The preceding chapter has presented a verification of the component modeling method, as applied to a model of the SRMS. The availability of a standard type of model (derived from the Lagrangian) using identical system parameters allows the accuracy of the component model to be strictly assessed. The accuracy of the component model as compared to the Lagrangian model is found to high, as would be expected.

In the second step of the validation of the modeling algorithm, the gain-scheduled $\mathcal{H}_\infty$ design algorithm of Chapter 3 is applied to the linearized SRMS model. It is noted that this is the first time in the literature that any of the Time-Varying Robust Control design tools is applied to a flexible manipulator. The ability to apply this approach is due solely to the availability of the component modeling approach. The result of the application are that the current LMI solution algorithms are too memory-intensive to provide a solution to realistic flexible manipulator problems. Further, the Small Gain condition for stability, as stated in the current work, is too conservative for application to general manipulator. This statement is based on the fact that the workspace of the manipulator is unduly limited.

Chapter 7

Conclusions

7.1 Summary

This thesis has presented some aspects of the application of modern multivariable control to the high-performance control of flexible space-based manipulators. This chapter summarizes the results which were presented, and indicates directions for future work.

7.2 Contributions and Conclusions

The primary contributions lie in the area of modeling for control.

(i) A modeling approach was developed to represent a geometrically nonlinear structure as a Linear Fractional Transformation of a linear plant through a time-varying structured operator. The primary utility of such a representation is to enable the use of modern robust control methods. In particular the Linear Fractional representation is the form needed for the application of small gain analysis and synthesis methods.

(ii) A secondary benefit of the linear fractional representation is the analytical simplification which results. The components with which the model is constructed are amenable to symbolic evaluation of the dynamics. In particular, for the classes of components considered in this work, a closed form model can be derived. Further, the linear fractional connection approach is a highly structured process. Therefore, since both the component modeling and the connection process are essentially closed form (to within well defined numerical operations), the dynamics of an arbitrary geometrically nonlinear flexible structure are amenable to modeling for simulation and control. In addition, the approach is extendible. Other components can be analyzed and put into the modeling framework. Higher order component models can be readily analyzed. Finally, the approach is well-suited to allow model updating based on physical test results. The symbolic form of the dynamics allow the derivation of analytical sensitivities, which enable updating on a component or system level.

(iii) A simplified dynamic model of a single, three-dimensional flexible link was developed in terms of a set of specific, physically motivated generalized coordinates. The coordinates, which include rigid body translations and rotations expressed in a body fixed link, lead to a low-order dynamic model. In addition, the rigid body coordinates simplify the application of the linear fractional interconnection modeling approach, since the interconnection variables coincide with the generalized coordinates. Finally, the availability of the analytical form for the link model reduces run-time complexity, since time-varying terms which appear more than once in the EOM need only be evaluated once.

(iv) The component modeling approach separates out the dynamics of each component. This leads to the inversion of more, smaller matrices (plus a boundary condition equation). In cases where the number of flexible DOF exceeds the number of rigid body DOF, the result can be dramatically fewer computations, and thus faster run time. Additionally, the component models are independent, raising the possibility of a parallel implementation for additional time savings.

Additionally, contributions to the control of flexible manipulators have been made.

(i) A corrected version of the small gain synthesis equations was generated. The corrected version was found to maintain Lyapunov stability of the closed loop system when the original synthesis equations did not.

(ii) The small gain synthesis equations were extended to produce reduced order controllers, and reduced-size controller uncertainty blocks. The extensions do not allow a controller order or uncertainty block size to be explicitly specified, but performance can be traded off against compensator size. The reduced order/reduced uncertainty order controllers ease real-time computational burden when implementing the controller. In addition, the reduced order LMI solutions help to alleviate numerical ill-conditioning which can complicate the construction the controller system matrices from the LMI solutions. The reduced order solutions are also observed to help limit the gain of the compensator, and prevent the occurrence of pole-zero cancellations which can adversely affect stability robustness in the presence of parametric uncertainty.

(iii) The small gain synthesis approach was applied for the first time to a flexible manipulator. Based on the approach the following deficiencies were noted: the present generation of numerical LMI solution algorithms are too resource intensive to be used on "realistically sized" manipulators. The SRMS model used in the study is low order compared to many flexible structures, since the dynamic range of interest encompasses only the lowest few modes. It is therefore necessary to conclude that for higher-order systems (for example the SRMS with wrist flexibility included, or for a flexible payload) a solution to the LMI conditions is currently out of reach. Second, the conservatism of the small gain approach was found to be too great to allow a single compensator to stabilize the manipulator over the entire workspace. Means for reducing the conservatism must be found if the promise of good performance with guaranteed stability is to be realized.

7.3 Recommendations for Future Work

The modeling approach can be extended in the following directions.

(i) The use of FE modeling to generate the component models should be investigated. FE modeling is a mainstay of commercial engineering, both due to the ability to model general, complex structures as well as to satisfy other requirements such as static and dynamic load testing. Such an extension would involve two areas: inclusion of the element rigid body modes (so as to capture Coriolis forces, nonlinear stiffening, and other effects of finite rigid body rotations) and generation of the "standard form" model, as used in this thesis. That is, the proper input/output map could be automated. In many useful cases this latter step will be trivial. For example, for a boundary between beam elements, the interface motions are the nodal DOF. For more general structures, specifically those with flexibility at the interface, the analysis techniques of Component Mode Synthesis can be brought to bear.

(ii) A general modeling algorithm would include body and surface forces. For example, modeling of gravity effects (gravity stiffening, effects of the gravity acceleration vector on accelerometer sensors, and others) is a necessary step for ground validation of physical models to be used for prediction of on-orbit behavior, to say nothing of modeling of ground based manipulators. A significant portion of this work would entail generating the placeholder variables which track the rotation to the inertial frame (or whichever frame the body forces are expressed in) as a function of the assumed modes DOF. These placeholder variables (for example Euler angles) would then be used to compute body and surface forces.

(iii) Nonlinear friction effects can become dominant in the dynamics of the manipulator for slow or small motions, and for highly geared systems. Important classes of friction include stiction, Coulomb friction, and viscous friction. The primary difficulty presented by friction is the lack of analytical methods for determining the friction coefficients. Secondarily, the friction forces are often highly non-deterministic and the simple models used can be greatly in error. The first difficulty has lead to various means for experimentally estimating friction coefficients. Coulomb friction and viscous friction may fit well into the Linear Fractional system representation proposed in this work. In contrast, stiction is an extremely difficult effect to model. The field for controlling in the presence of stiction remains open.

(iv) A relatively straightforward extension of the configuration modeling algorithm would be to alternative topologies: tree structures and closed structures. The latter would be trivial, involving only matching the "outer" boundary conditions for the components at either end of the chain of bodies. The former extension would be essentially notational, with the boundary output function for each component including a single driving acceleration, and multiple external force outputs. The boundary condition equation would then be written by following the accelerations around a spatially colocated "ring", and would lead to a single coupling condition for all the components at the interface.

The field of modern control offers many possibilities for research.

(i) Increased efficiency of the numerical solution algorithms for LMIs. At present the solution algorithms are too resource-intensive, precluding their use in many problems. Alternatively, it may be fruitful to restate the LMI conditions so as to use other solution approaches such as steepest descent/gradient search, continuation/homotopy methods, or other iterative solution approaches.

(ii) The small gain conditions are excessively conservative because the parameters are allowed to vary infinitely fast. Incorporation of bounds on the parameter rates would reduce conservatism, at the expense of increased computational requirements due to increased system order.

(iii) The small-gain framework may be excessively conservative for manipulator problems. Within the robust/time-varying control framework, alternative approaches such as nonlinear-observer-based controllers, or polytopic controllers, may provide better performance, while maintaining formal guarantees on stability and remaining open to the use of methods for accounting for true parametric or unstructured uncertainty (as contrasted to a known but time-varying quantity).

References

[1] Newsom, J. R., W. E. Layman, H. B. Waites, and R. J. Hayduk, "The NASA Controls-Structures Interaction Technology Program," *Proceedings, 41st Congress of the International Astronautical Federation*, Dresden, Oct. 1990.

[2] Oakley, C. M. and R. H. Cannon, "Anatomy of an Experimental Two-Link Flexible Manipulator under End-Point Control," in *Proceedings of the* 29th *Conference on Decision and Control*, Dec. 1990, pp. 507–513.

[3] Yurkovich, S., A. Tzes, I. Lee, and K. Hillsley, "Control and System Identification of a Two-Link Flexible Manipulator," in *Preceedings of the IEEE International Conference on Robotics and Automation*, May 1990, pp. 1626–1631.

[4] Schmitz, E. and M. Ramey, "Initial Experiments on the End-Point Control of a 2-DOF Long-Reach Elastic Manipulator," *Proceedings, Cooperative Intelligent Robotics in Space*, 1991, pp. 245–256.

[5] Aoustin, Y. and C. Chevallereau, "The Singular Perturbation Control of a Two-Flexible-Link Robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, 1993, pp. 737–742.

[6] Carusone, J. and G. M. T. D'Eleuterio, "Tracking Control for End-Effector Position and Orientation of Structurally Flexible Manipulators," *Journal of Robotic Systems*, Vol. 10, No. 6, 1993, pp. 847–870.

[7] Moudgal, V., K. Passino, and S. Yurkovich, "Rule-Based Control for a Flexible-Link Robot," *IEEE Trans. on Control Systems Technology*, Vol. 2, No. 4, Dec. 1994, pp. 392–405.

[8] Khorrami, F., S. Jain, and A. Tzes, "Experimental Results on Adaptive Nonlinear Control and Input Preshaping for Multi-Link Flexible Manipulators," *Automatica*, Vol. 31, No. 1, 1995, pp. 83–97.

[9] Rey, D. and R. Glaese, "Finite Element Model of the Middeck Active Control Experiment," 1993. Preprint, submitted to the *1993 AIAA Structures, Structural Dynamics, and Materials Conference.*

[10] Glaese, R. M. and D. W. Miller, "On-Orbit Modelling of the Middeck Active Control Experiment from 1-g Analysis and Experimentation," *Proceedings, 12th International Modal Analysis Conference*, Honolulu, Hawaii, Feb. 1994.

[11] Liu, K., R. Jacques, and D. Miller, "Frequency Domain Structural System Identification by Observability Range Space Extraction," in *Proceedings of the Amer-*

ican Control Conference, 1994.

[12] Jacques, R. and D. Miller, "Multivariable Model Identification from Frequency Response Data," in *Proceedings of the 32nd Conference on Decision and Control*, 1993.

[13] Crawley, E. F., M. S. Barlow, M. C. van Schoor, and A. S. Bicos, "Variation in the Modal Parameters of Space Structures," *Proceedings, Proceedings of the 1992 AIAA Structures, Structural Dynamics, and Materials Conference*, (Dallas, TX), Apr. 1992.

[14] Campbell, M., *Uncertainty Modeling for Structural Control Analysis and Synthesis*, Ph.D. thesis, Department of Aeronautics and Astronautics, M.I.T., Jan. 1996.

[15] Douglas, J., *Identification of Parametric Uncertainty for the Control of Flexible Structures*, Ph.D. thesis, Department of Aeronautics and Astronautics, M.I.T., Cambridge, MA, May 1995.

[16] Grocott, S. C. O., *Comparison of Control Techniques for Robust Performance on Uncertain Structural Systems*, Master's thesis, Department of Aeronautics and Astronautics, M.I.T., Cambridge, MA, Jan. 1994.

[17] Doyle, J. C. and G. Stein, "Multivariable System Design: Concepts for a Classical/Modern Synthesis," *IEEE Trans. Automatic Control*, Vol. AC–26, No. 1, Feb. 1981, pp. 4–16.

[18] Apkarian, P., J.-M. Biannic, and P. Gahinet, "Self-Scheduled $\mathcal{H}_\infty$ Control of Missile via Linear Matrix Inequalities," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, May–June 1995, pp. 532–538.

[19] Sesak, J. R., *Sensitivity Constrained Linear Optimal Control Analysis and Synthesis*, Ph.D. thesis, University of Wisconsin, 1974.

[20] Padilla, C. E. and A. H. von Flotow, *Nonlinear Strain-Displacement Relations in the Dynamics of a Two-Link Flexible Manipulator*, Master's thesis, Department of Aeronautics and Astronautics, M.I.T., Cambridge, MA, May 1989.

[21] Book, W. J., "Modeling, Design, and Control of a Flexible Manipulator Arms: a Tutorial Review," in *Proceedings of the* 29th *Conference on Decision and Control*, Dec. 1990, pp. 500–505.

[22] Park, K. C., *Structural Dynamics*, Course Notes, 1992.

[23] Sincarsin and Hughes, "Dynamics of an Elastic Multibody Chain: Part A - Body Motion Equations, and Park B - Global Dynamics," *Dynamics and Stability of Systems*, Vol. 4, No. (3&4), 1989, pp. 209–244.

[24] Demeo, M., S. Gilbert, M. Scott, J. Lepanto, E. Bains, and M. Jensen, "Human-in-the-Loop Evaluation of RMS Active Damping Augmentation," 1993, pp. 1571–1581.

[25] Beck, C., "Minimality for Uncertain Systems and IQCs," in *Proceedings of the 33rd Conference on Decision and Control*, Dec. 1994, pp. 3068–3073.

[26] Cheng, Y. and B. De Moor, "A Multidimensional Realization Algorithm for Parametric Uncertainty Modeling Problems," *Proceedings, Proceedings of the* 32nd *Conference on Decision and Control*, San Antonio, TX, Dec. 1993, pp. 3022–3023.

[27] Feron, E., "Observer-Based Stabilization of Nonlinear Systems," *Proceedings, Proceedings of the* 13th *American Control Conference*, Baltimore, MD, Vol. 1, 1994, pp. 436–439.

[28] Packard, A., "Gain Scheduling via Linear Fractional Transformations," *Systems and Control Letters*, Vol. 22, No. 2, Feb. 1994, pp. 79–92.

[29] Apkarian, P. and P. Gahinet, "A Convex Characterization of Gain-Scheduled $\mathcal{H}_\infty$ Controllers," *IEEE Trans. Automatic Control*, Vol. 40, No. 5, May 1995, pp. 853–864.

[30] Bennani, S., S. Mulder, and O. H. Bosgra, "Robust Control: A Structured Approach to Solve Aircraft Flight Control Problems," *Proceedings, Flight Mechanics Panel Symposium*, Turin, Italy, May 1994.

[31] Book, W. J., "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *International Journal of Robotics Research*, Vol. 3, No. 3, Fall 1984, pp. 87–101.

[32] Blaurock, C., D. W. Miller, E. Schmitz, K. Liu, and W. P. Seering, "Preliminary Performance Versus Cost Comparison of Advanced Control on the Shuttle Remote Manipulator," *Proceedings, Proceedings of the* 37th *AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, Salt Lake City, UT, Apr. 1996.

[33] Sharf, I. and G. M. T. D'Eleuterio, "An Iterative Approach to Multibody Simulation Dynamics Suitable for Parallel Implementation," Tech. Rep. UTIAS-TN-276, Institute for Space Studies, Toronto University, 1994.

[34] Craig, R. R. J., "A Review of Time-Domain and Frequency-Domain Component-Mode Synthesis Methods," *Journal of Modal Analysis*, Apr. 1987, pp. 59–72.

[35] Schmitz, E., "Dynamics and Control of a Planar Manipulator with Elastic Links," in *Proceedings of the* 25th *Conference on Decision and Control*, Dec. 1986, pp. 1135–1139.

[36] Gebler, B., "Feed-Forward Control Strategy For An Industrial Robot with Elastic Links and Joints," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1987, pp. 923–928.

[37] Khorrami, F., "Analysis of Multi-Link Flexible Manipulators Via Asymptotic Expansions," in *Proceedings of the* 28th *Conference on Decision and Control*, Dec. 1989, pp. 2089–2094.

[38] Zaki, A. S. and W. El-Maraghy, "Model Reference Adaptive Control for a Three-Degree of Freedom Manipulator with Flexible Links," in *Proceedings of the American Control Conference*, June 1993, pp. 1400–1405.

[39] Ramey, M. and E. Schmitz, "LQR Design for an Experimental Planar Elastic

Arm with a Large Tip Payload," *Proceedings, Proceedings of the American Control Conference*, San Diego, CA, Vol. 2, 1990, pp. 1728–1732.

[40] Oakley, C. M. and C. H. Barrat, "End-Point Controller Design For an Experimental Two-Link Flexible Manipulator Using Convex Optimization," *Proceedings, American Control Conference*, May 1990, pp. 1752–1759.

[41] Carusone, J. and G. M. T. D'Eleuterio, "Experiments in the Control of Structurally Flexible Manipulators with the RADIUS Facility," *Proceedings, 2nd Joint Japan/US Conference on Adaptive Structures*, Nagoya, Japan, 1991, pp. 588–605.

[42] Menq, C.-H. and J.-S. Chen, "Dynamic Modeling and Payload-Adaptive Control of A Flexible Manipulator," in *American Control Conference*, 1988.

[43] Rovner, D. M. and G. F. Franklin, "Experiments in Load-Adaptive Control of a Very Flexible One-Link Manipulator," *Automatica*, Vol. 24, No. 4, July 1988, pp. 541–548.

[44] Alder, L. and S. Rock, "Adaptive Control of a Flexible-Link Manipulator with Unknown Payload Dynamics," in *Proceedings of the 1993 American Control Conference*, 1993, pp. 2088–2092.

[45] Yurkovich, S., F. Pacheco, and A. Tzes, "On-Line Frequency Domain Information for Control of a Flexible-Link Robot with Varying Payload," *IEEE Trans. Automatic Control*, Vol. AC–33, No. 12, 1989, pp. 1300–1303.

[46] Beck, C., B. Bodenheimer, and P. Bendotti, "LMI-Based Model Reduction for a Vectored-Thrust Ducted Fan," in *Proceedings of the 34th Conference on Decision and Control*, Dec. 1995, pp. 871–872.

[47] Bodenheimer, B., P. Bendotti, and M. Kantner, "Linear Parameter-Varying Control of a Ducted Fan Engine," *International Journal of Robust and Nonlinear Control*, 1995.

[48] Cetinkunt, S. and W. J. Book, "Performance Limitations of Joint Variable-Feedback Controllers Due to Manipulator Structural Flexibility," *IEEE Trans. on Robotics and Automation*, Vol. 6, No. 2, Apr. 1990, pp. 219–231.

[49] Scott, M. A., M. G. Gilbert, and M. E. Demeo, "Active Damping of the Space Shuttle Remote Manipulator System," *Proceedings, Proceedings of the AIAA 1991 Conference on Guidance and Control*, New Orleans, LA, Aug. 1991, pp. 194–204.

[50] Kotnik, P. T., S. Yurkovich, and U. Ozguner, "Acceleration Feedback for Control of a Flexible Manipulator Arm," *Journal of Robotic Systems*, Vol. 5, No. 3, 1988, pp. 181–196.

[51] Siciliano, B. and W. J. Book, "A Singular Perturbation Approach to Control of Lightweight Flexible Manipulators," *International Journal of Robotics Research*, Vol. 7, No. 4, Aug. 1988, pp. 79–90.

[52] Madhavan, S. and S. N. Singh, "Variable Structure Trajectory Control of an

Elastic Robotic Arm," *Journal of Robotic Systems*, Vol. 10, No. 1, 1993, pp. 23–44.

[53] Khorrami, F., S. Jain, and A. Tzes, "Experiments on Rigid Body-Based Controllers with Input Preshaping for a Two-Link Flexible Manipulator," *IEEE Trans. on Robotics and Automation*, Vol. 10, No. 1, Feb. 1994, pp. 55–65.

[54] Yurkovich, S., K. Hillsley, and A. Tzes, "Indentification and Control for a Manipulator with Two Flexible Links," in *Preceedings of the 29^{th} Conference on Decision and Control*, Dec. 1990, pp. 1995–2000.

[55] Askew, C. and M. Sundareshan, "Adaptive Variable Structure Control of Flexible Manipulators by Neural Network Payload Identification," in *Proceedings of the 32^{nd} Conference on Decision and Control*, Dec. 1993, pp. 3249–3250.

[56] Clarke, D. W., C. Mohtadi, and P. S. Tuffs, "Generalized Predictive Control-Part I. The Basic Algorithm Part II. Extensions and Interpretations," *Automatica*, Vol. 23, No. 2, 1987, pp. 137–160.

[57] Singer, N. C. and W. P. Seering, "Preshaping Command Inputs to Reduce System Vibration," *Transactions of the ASME*, Vol. 112, Mar. 1990, pp. 76–82.

[58] Bayo, E., "Computed Torque for the Position Control of Open-Chain Flexible Robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Apr. 1988, pp. 316–321.

[59] Pfeiffer, F. and R. Johanni, "A Concept for Manipulator Trajectory Planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1986, pp. 1397–1405.

[60] Meirovich, L., *Elements of Vibration Analysis*, McGraw-Hill, New York, 1986.

[61] Kane, T. R., *Dynamics: Theory and Application*, McGraw-Hill, NY, 1985.

[62] Athans, M., D. Castanon, K.-P. Dunn, C. Greene, W. H. Lee, J. Sandell, N. R., and A. S. Willsky, "The Stochastic Control of the F-8C Aircraft using a Multiple Model Adaptive Control (MMAC) Method: I. Equilibrium," *IEEE Transactions on Automatic Control*, Vol. AC-22, No. 5, Oct. 1977, pp. 768–780.

[63] Oakley, C. M. and R. H. Cannon., "End-Point Control of a Two-Link Manipulator with a Very Flexible Forearm: Issues and Experiments," in *Proceedings of the American Control Conference*, June 1989, pp. 1381–1388.

[64] Tzes, A. and S. Yurkvich, "Application and Comparison of On-Line Identification Methods for Flexible Manipulator Control," *Proceedings, International Conference on Advanced Robotics*, Columbus, OH, 1989.

[65] Catto, E. S. and F. C. Moon, "Robust Control of a Shape-Changing Flexible Robot Arm," in *1995 SPIE North American Conference on Smart Structures and Materials*, 27 February–3 March 1995, pp. 171–181.

[66] De Luca, A., P. Lucibello, and G. Ulivi, "Inversion Techniques for Trajectory Control of Flexible Robot Arms," *Journal of Robotic Systems*, Vol. 6, No. 4, 1989, pp. 325–344.

[67] Ravindran, R. and K. H. Doetsch, "Design Aspects of the Shuttle Remote Manipulator Control," *Proceedings, Proceedings of the Guidance and Control Conference*, Aug. 1982, pp. 456–465.

[68] Slotine, J.-J. and W. Li, *Applied Nonlinear Control*, Prentice–Hall, Englewood Cliffs, NJ, 1991.

[69] Miller, D. W., *Modelling and Active Modification of Wave Scattering in Structural Networks*, Ph.D. thesis, Department of Aeronautics and Astronautics, M.I.T., 1988. (Space Systems Laboratory Report #12–88).

[70] Hyde, T. T., *Active Vibration Isolation for Precision Space Structures*, Ph.D. thesis, Department of Aeronautics and Astronautics, M.I.T., Cambridge, MA, Feb. 1996.

[71] Blackwood, G., *Active Vibration Isolation for Controlled Flexible Structures*, Ph.D. thesis, Department of Aeronautics and Astronautics, M.I.T., Cambridge, MA, Oct. 1993.

[72] Bernstein, D. S., W. M. Haddad, and D. C. Hyland, "Small Gain Versus Positive Real Modeling of Real Parameter Uncertainty," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 2, Feb. 1992, pp. 538–540.

[73] Kwakernaak and Sivan, *Linear Optimal Control Systems*, Wiley-Interscience, 1972.

[74] Padilla, C. E., *Performance Limits and Robustness Issues in the Control of Flexible Link Manipulators*, Ph.D. thesis, Department of Aeronautics and Astronautics, M.I.T., Cambridge, MA, May 1992.

[75] Doyle, J. C., "Analysis of Feedback Systems with Structured Uncertainties," *IEE Proceedings*, Vol. 129, Part D, No. 6, Nov. 1982, pp. 242–250.

[76] Athans, M., *Optimal Control*, McGraw-Hill, New York, 1966.

[77] Gahinet, P. and P. Apkarian, "A Linear Matrix Inequality Approach to $\mathcal{H}_\infty$ Control," *International Journal of Robust and Nonlinear Control*, Vol. 4, 1994, pp. 421–448.

[78] Gahinet, P., A. Nemirovski, A. Laub, and M. Chilali, *LMI Control Toolbox*, The MathWorks, 1995.

[79] Feliu, V., K. S. Rattan, and J. H. B. Brown, "Control of Flexible Arms with Friction in the Joints," *IEEE Trans. on Robotics and Automation*, Vol. 9, Aug. 1993, pp. 467–475.

[80] Nguyen, P. K., R. Carr, and D. M. Gossain, "Structural Flexibility of the Shuttle Remote Manipulator System Mechanical Arm," *Proceedings, Proceedings of the Guidance and Control Conference*, AIAA, Aug. 1982, pp. 246–256.

[81] "Payload Deployment and Retrieval System Sim Database, Version 1.0," Tech. Rep. JSC-25134, Johnson Space Center, July 1991.

[82] Abelow, A., "Dynamic Equation Set for a Simplified Simulation of the SSRM

System," Tech. Rep. R-1258, Charles Stark Draper Laboratories, Feb. 1980.

[83] Grace, A., *Matlab Optimization Toolbox*, MathWorks, 1992.

Appendix A

Mass-Loaded Mode Shapes

Oakley has determined the mode shapes for a mass-loaded cantilevered beam in closed form[2]. In some cases it may be advantageous to numerically determine the mode shapes. For example, a stiffness boundary condition at either end can be readily introduced.

A.1 Mass-Loaded Transverse Mode Shapes

The spatial shape functions of the Bernoulli-Euler beam are determined as the solutions to the partial differential equation

$$\frac{\partial}{\partial x^2}EI(x)\frac{\partial w(x,t)}{\partial x^2} - \rho A\frac{dw(x,t)}{\mathrm{d}t^2} = 0 \tag{A.1}$$

The transverse deformations $w(x,t)$ are assumed separable in space and time:

$$w(x,t) = \phi(x)e^{(i\omega t)} \tag{A.2}$$

where ω is the temporal frequency. It is clear that in order for this to be a solution to the PDE, the fourth derivative of the mode shape must be proportional to the first. The trial function

$$\phi(x) = a_1\cos(\kappa x) + a_2\cosh(\kappa x) + a_3\sin(\kappa x) + a_4\sinh(\kappa x) \tag{A.3}$$

satisfies this criteria. The PDE holds when $\kappa^4 = \omega^2\frac{\rho A}{EI}$. The spatial frequency κ and the shape function coefficients a_i are determined from the boundary conditions.

The mode shape parameters can be determined numerically. To proceed, the mode shape can be defined in terms of vectors ψ and a:

$$\phi(x) = \psi(x)^T a \tag{A.4}$$

$$\psi(x)^T = \begin{bmatrix} \cos(\kappa x) & \cosh(\kappa x) & \sin(\kappa x) & \sinh(\kappa x) \end{bmatrix} \tag{A.5}$$

$$a^T = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \end{bmatrix} \tag{A.6}$$

For the mass loaded cantilevered beam, the displacement and rotation boundary conditions at the constrained end are set equal to zero. The force and moment conditions at the mass-loaded end are set equal to the reaction force and moment of the end mass. In vector form,

$$\begin{bmatrix} w(0) \\ w'(0) \\ -EIw'''(L) \\ EIw''(L) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ m_t\ddot{w}(L_t) + m_t b_x \ddot{w}'(L_t) \\ m_t\ddot{w}(L_t) + (I_t + m_t b_x^2)\ddot{w}'(L_t) \end{bmatrix} \tag{A.7}$$

where the lower two elements of the vector on the left are the internal force and moment of the beam. The trial solution, in terms of space via $\phi = \psi(x)a$ and time via $e^{\imath\omega t}$, is substituted into the boundary condition equations. The BC must hold independent of time. Taking the time-independent portion, the following equation results:

$$\begin{bmatrix} \psi(0)^T \\ \psi(0)'^T \\ -EI\psi(L)'''^T \\ EI\psi(L)''^T \end{bmatrix} a = \begin{bmatrix} 0 \\ 0 \\ m_t\omega^2\psi(L)^T + m_t b_x \omega^2 \psi(L)'^T \\ m_t\omega^2\psi(L)^T + (I_t + m_t b_x^2)\omega^2\psi(L)'^T \end{bmatrix} a \tag{A.8}$$

Note that the right hand side is a function of the temporal frequency ω. The mode number relation can be used to produce a set of equations in terms of the spatial frequency only. When this is done the right hand side will have EI in the numerator and ρA in the denominator. Since each of the equations which make up the rows of the matrix are equal to zero, the EI terms can be divided out and the ρA terms cleared. The matrix then has the form

$$\begin{bmatrix} \psi(0)^T \\ \psi(0)'^T \\ -\rho A\psi(L)'''^T - m_t\kappa^4\psi(L)^T - m_t b_x\kappa^4\psi(L)'^T \\ \rho A\psi(L)''^T - m_t b_x\kappa^4\psi(L)^T - (I_t + m_t b_x^2)\kappa^4\psi(L)'^T \end{bmatrix} a = 0 \qquad \text{(A.9)}$$

which is a function only of the inertia parameters. The spatial frequencies κ are determined by setting the determinant of the above matrix to zero. The coefficients a are found from the eigenvector associated with the zero direction.

A.2 Mass Loaded Torsional Mode Shapes

The torsional mode shapes are determined similarly to the transverse mode shapes. The torsional PDE is

$$\frac{\partial}{\partial x}GJ\frac{\partial\theta}{\partial x} + \rho I\theta = 0 \qquad \text{(A.10)}$$

which is solved by spatial trial functions of the form

$$\phi = a_1\cos(\kappa x) + a_2\sin(\kappa x) = \psi(x)^T a \qquad \text{(A.11)}$$

with $\kappa = \omega\sqrt{\frac{\rho I}{GJ}}$. Setting the root displacement to zero and the tip internal moment to the inertial reaction moment of the tip mass,

$$\begin{bmatrix} \theta(0,t) \\ GJ\theta'(L,t) \end{bmatrix} = \begin{bmatrix} 0 \\ I_t\ddot{\theta} \end{bmatrix} \qquad \text{(A.12)}$$

Substituting the trial function, the following matrix condition results:

$$\begin{bmatrix} \psi(0)^T \\ GJ\psi(L)'^T - \omega^2 I_t\psi(x)^T \end{bmatrix} a = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \text{(A.13)}$$

Note that

$$\omega^2 I\psi(x) = \frac{GJ}{\rho I}\kappa^2\psi(x) \qquad \text{(A.14)}$$

and further

$$\kappa^2\psi(x) = -\psi(x)'' \qquad \text{(A.15)}$$

so that the above matrix for the evalulation of the mode numbers κ can be written

$$\begin{bmatrix} \psi(0)^T \\ GJ\psi(L)'^T + I_t \frac{GJ}{\rho I}\psi(x)''^T \end{bmatrix} \tag{A.16}$$

Further, since each row is equal to zero at the solution, the stiffness term GJ can be canceled and the distributed inertia term ρI can be cleared, resulting in the matrix expression

$$\begin{bmatrix} \psi(0)^T \\ \rho I\psi(L)'^T + I_t\psi(x)''^T \end{bmatrix} \tag{A.17}$$

which is solely a function of the inertia parameters of the link. Again the spatial frequencies are found by setting the determinant equal to zero. The coefficients are determined from the associated eigenvector.

Appendix B

Link Local Equations of Motion

This section presents the full form of the link EOM as presented in Chapter 4.

The following constant scalars and matrices are defined:

$$M_{xx} = \int_0^L \phi(r)_x \phi(r)_x^T \mathrm{d}r \tag{B.1}$$

$$M_{xy} = \int_0^L \phi(r)_x \phi(r)_y^T \mathrm{d}r \tag{B.2}$$

$$M_{xz} = \int_0^L \phi(r)_x \phi(r)_z^T \mathrm{d}r \tag{B.3}$$

$$M_{yy} = \int_0^L \phi(r)_y \phi(r)_y^T \mathrm{d}r \tag{B.4}$$

$$M_{yz} = \int_0^L \phi(r)_y \phi(r)_z^T \mathrm{d}r \tag{B.5}$$

$$M_{zz} = \int_0^L \phi(r)_z \phi(r)_z^T \mathrm{d}r \tag{B.6}$$

$$N_{xx} = \int_0^L r\phi(r)_x \phi(r)_x^T \mathrm{d}r \tag{B.7}$$

$$N_{xy} = \int_0^L r\phi(r)_x \phi(r)_y^T \mathrm{d}r \tag{B.8}$$

$$N_{xz} = \int_0^L r\phi(r)_x \phi(r)_z^T \mathrm{d}r \tag{B.9}$$

$$N_{yy} = \int_0^L r\phi(r)_y \phi(r)_x^T \mathrm{d}r \tag{B.10}$$

$$N_{yz} = \int_0^L r\phi(r)_y\phi(r)_y^T \mathrm{d}r \tag{B.11}$$

$$N_{zz} = \int_0^L r\phi(r)_z\phi(r)_z^T \mathrm{d}r \tag{B.12}$$

$$M'_{xy} = \int_0^L \phi(r)_x\phi(r)_y'^T \mathrm{d}r \tag{B.13}$$

$$M'_{xz} = \int_0^L \phi(r)_x\phi(r)_z'^T \mathrm{d}r \tag{B.14}$$

$$K_{xx} = \int_0^L \phi(r)_x''\phi(r)_x''^T \mathrm{d}r \tag{B.15}$$

$$K_{yy} = \int_0^L \phi(r)_y''\phi(r)_y''^T \mathrm{d}r \tag{B.16}$$

$$K_{zz} = \int_0^L \phi(r)_z''\phi(r)_z''^T \mathrm{d}r \tag{B.17}$$

$$M_x = \int_0^L \phi(r)_x \mathrm{d}r \tag{B.18}$$

$$M_y = \int_0^L \phi(r)_y \mathrm{d}r \tag{B.19}$$

$$M_z = \int_0^L \phi(r)_z \mathrm{d}r \tag{B.20}$$

$$N_x = \int_0^L \phi(r)_x \mathrm{d}r \tag{B.21}$$

$$N_y = \int_0^L \phi(r)_y \mathrm{d}r \tag{B.22}$$

$$N_z = \int_0^L \phi(r)_z \mathrm{d}r \tag{B.23}$$

$$M''_{yy} = \int_0^L \phi(r)_y'\phi(r)_y'^T \mathrm{d}r \tag{B.24}$$

$$M''_{zz} = \int_0^L \phi(r)_z'\phi(r)_z'^T \mathrm{d}r \tag{B.25}$$

$$M'_x = \int_0^L \phi(r)_x' \mathrm{d}r \tag{B.26}$$

$$M'_y = \int_0^L \phi(r)_y' \mathrm{d}r \tag{B.27}$$

$$M'_z = \int_0^L \phi(r)_z' \mathrm{d}r \tag{B.28}$$

$$N'_x = \int_0^L r\phi(r)'_x \mathrm{d}r \tag{B.29}$$

$$N'_y = \int_0^L r\phi(r)'_y \mathrm{d}r \tag{B.30}$$

$$N'_z = \int_0^L r\phi(r)'_z \mathrm{d}r \tag{B.31}$$

with $(.)' = \frac{\partial .}{\partial r}$.

And also the quantities:

$$\beta = \frac{1}{2}L^2, \;\; \gamma = \frac{1}{3}L^{\frac{1}{3}} \tag{B.32}$$

From these define the time varying signals:

$$m_1 = m_t + L\rho A \tag{B.33}$$

$$m_2 = b_x m_t + L_t m_t + \beta\rho A \tag{B.34}$$

$$m_3 = M_y\rho A + m_t\phi_{y_t} + b_x m_t\phi'_{y_t} \tag{B.35}$$

$$m_4 = M_z\rho A + m_t\phi_{z_t} + b_x m_t\phi'_{z_t} \tag{B.36}$$

$$m_5 = Ix + I_{x_t} \tag{B.37}$$

$$m_6 = M_x pIx + I_{x_t}\phi_{x_t} \tag{B.38}$$

$$m_7 = Iy + I_{y_t} + b_x^2 m_t + 2b_x L_t m_t + L_t^2 m_t + \gamma\rho A \tag{B.39}$$

$$m_8 = -(N_z\rho A) - M'_z\rho I_y - b_x m_t\phi_{z_t} - L_t m_t\phi_{z_t} - I_{y_t}\phi'_{z_t} - b_x^2 m_t\phi'_{z_t} - b_x L_t m_t\phi'_{z_t} \tag{B.40}$$

$$m_9 = Iz + I_{z_t} + b_x^2 m_t + 2b_x L_t m_t + L_t^2 m_t + \gamma\rho A \tag{B.41}$$

$$m_{10} = N_y\rho A + M'_y\rho I_z + b_x m_t\phi_{y_t} + L_t m_t\phi_{y_t} + I_{z_t}\phi'_{y_t} + b_x^2 m_t\phi'_{y_t} + b_x L_t m_t\phi'_{y_t} \tag{B.42}$$

$$m_{11} = M_{xx}pIx + I_{x_t}\phi_{x_t}\phi_{x_t} \tag{B.43}$$

$$m_{12} = M_{yy}\rho A + M''_{yy}\rho I_z + m_t\phi_{y_t}\phi_{y_t} + b_x m_t(\phi_{y_t}\phi'_{y_t} + \phi'_{y_t}\phi_{y_t}) + I_{z_t}\phi'_{y_t}\phi'_{y_t} + b_x^2 m_t\phi'_{y_t}\phi'_{y_t} \tag{B.44}$$

$$m_{13} = M_{zz}\rho A + M''_{zz}\rho I_y + m_t\phi_{z_t}\phi_{z_t} + b_x m_t(\phi_{z_t}\phi'_{z_t} + \phi'_{z_t}\phi_{z_t}) + I_{y_t}\phi'_{z_t}\phi'_{z_t} + b_x^2 m_t\phi'_{z_t}\phi'_{z_t} \tag{B.45}$$

$$b_1 = -b_x - L_t \tag{B.46}$$

$$b_2 = \phi_{x_t} \tag{B.47}$$

$$b_3 = \phi_{y_t} + b_x \phi'_{y_t} \tag{B.48}$$

$$b_4 = \phi'_{y_t} \tag{B.49}$$

$$b_5 = \phi_{z_t} + b_x \phi'_{z_t} \tag{B.50}$$

$$b_6 = -\phi'_{z_t} \tag{B.51}$$

$$v_1 = 2 b_x m_t \phi'_{z_t} \tag{B.52}$$

$$v_2 = N_y \rho A + b_x m_t \phi_{y_t} + L_t m_t \phi_{y_t} + b_x^2 m_t \phi'_{y_t} + b_x L_t m_t \phi'_{y_t} \tag{B.53}$$

$$v_3 = N_z \rho A + b_x m_t \phi_{z_t} + L_t m_t \phi_{z_t} - b_x^2 m_t \phi'_{z_t} - b_x L_t m_t \phi'_{z_t} \tag{B.54}$$

$$m_{x_1} = (-(M_{xz} \rho A) - m_t \phi_{x_t} \phi_{z_t} - b_x m_t \phi_{x_t} \phi'_{z_t})^T q_x \tag{B.55}$$

$$m_{x_2} = (M_{xy} \rho A + m_t \phi_{x_t} \phi_{y_t} + b_x m_t \phi_{x_t} \phi'_{y_t})^T q_x \tag{B.56}$$

$$m_{x_3} = (-(N_{xy} \rho A) - M'_{xy} \rho I_y - b_x m_t \phi_{x_t} \phi_{y_t} - L_t m_t \phi_{x_t} \phi_{y_t} - I_{y_t} \phi_{x_t} \phi'_{y_t} - b_x^2 m_t \phi_{x_t} \phi'_{y_t} - b_x L_t m_t \phi_{x_t} \phi'_{y_t})^T q_x \tag{B.57}$$

$$m_{x_4} = (-(N_{xz} \rho A) - M'_{xz} \rho I_z - b_x m_t \phi_{x_t} \phi_{z_t} - L_t m_t \phi_{x_t} \phi_{z_t} - I_{z_t} \phi_{x_t} \phi'_{z_t} - b_x^2 m_t \phi_{x_t} \phi'_{z_t} - b_x L_t m_t \phi_{x_t} \phi'_{z_t})^T q_x \tag{B.58}$$

$$m_{x_5} = (2 b_x m_t \phi_{x_t} \phi'_{z_t})^T q_x \tag{B.59}$$

$$m_{x_6} = (-2 b_x m_t \phi_{x_t} \phi_{x_t})^T q_x \tag{B.60}$$

$$m_{x_7} = (-(N_{xz} \rho A) - b_x m_t \phi_{x_t} \phi_{z_t} - L_t m_t \phi_{x_t} \phi_{z_t} + b_x^2 m_t \phi_{x_t} \phi'_{z_t} + b_x L_t m_t \phi_{x_t} \phi'_{z_t})^T q_x \tag{B.61}$$

$$m_{x_8} = (N_{xy} \rho A + b_x m_t \phi_{x_t} \phi_{y_t} + L_t m_t \phi_{x_t} \phi_{y_t} + b_x^2 m_t \phi_{x_t} \phi'_{y_t} + b_x L_t m_t \phi_{x_t} \phi'_{y_t})^T q_x \tag{B.62}$$

$$m_{x_9} = (-2 b_x^2 m_t \phi_{x_t} \phi_{x_t} - 2 b_x L_t m_t \phi_{x_t} \phi_{x_t})^T q_x \tag{B.63}$$

$$m_{y_1} = (-(M_y \rho A) - m_t \phi_{y_t} - b_x m_t \phi'_{y_t})^T q_y \tag{B.64}$$

$$m_{y_2} = (-(b_x m_t \phi'_{y_t} \phi'_{y_t}))^T q_y \tag{B.65}$$

$$m_{y_3} = (M_{xy} \rho A + m_t \phi_{x_t} \phi_{y_t})^T q_y \tag{B.66}$$

$$m_{y_4} = (-(N_y \rho A) - b_x m_t \phi_{y_t} - L_t m_t \phi_{y_t} - b_x^2 m_t \phi'_{y_t} - b_x L_t m_t \phi'_{y_t})^T q_y \tag{B.67}$$

$$m_{y_5} = (M_{yz}\rho A + m_t\phi_{y_t}\phi_{z_t} + 2b_x m_t\phi'_{y_t}\phi_{z_t} + b_x^2 m_t\phi'_{y_t}\phi'_{z_t})^T q_y \tag{B.68}$$

$$m_{y_6} = (-(N_{xy}\rho A) - b_x m_t\phi_{x_t}\phi_{y_t} - L_t m_t\phi_{x_t}\phi_{y_t})^T q_y \tag{B.69}$$

$$m_{y_7} = -(b_x m_t\phi_{x_t}\phi'_{y_t}))^T q_y \tag{B.70}$$

$$m_{y_8} = -(M_{yy}\rho A) - m_t\phi_{y_t}\phi_{y_t} - b_x m_t(\phi_{y_t}\phi'_{y_t} + \phi'_{y_t}\phi_{y_t}) - b_x^2 m_t\phi'_{y_t}\phi'_{y_t})^T q_y \tag{B.71}$$

$$m_{y_9} = N_{xy}\rho A + b_x m_t\phi_{x_t}\phi_{y_t} + L_t m_t\phi_{x_t}\phi_{y_t} + b_x^2 m_t\phi_{x_t}\phi'_{y_t} + b_x L_t m_t\phi_{x_t}\phi'_{y_t})^T q_y \tag{B.72}$$

$$m_{y_10} = M_{yz}\rho A + m_t\phi_{y_t}\phi_{z_t} - b_x^2 m_t\phi'_{y_t}\phi'_{z_t})^T q_y \tag{B.73}$$

$$m_{z_1} = (M_z\rho A + m_t\phi_{z_t} - b_x m_t\phi'_{z_t})^T q_z \tag{B.74}$$

$$m_{z_2} = (b_x m_t\phi'_{z_t}\phi'_{z_t})^T q_z \tag{B.75}$$

$$m_{z_3} = (-(M_{xz}\rho A) - m_t\phi_{x_t}\phi_{z_t})^T q_z \tag{B.76}$$

$$m_{z_4} = (-(N_z\rho A) - b_x m_t\phi_{z_t} - L_t m_t\phi_{z_t} + b_x^2 m_t\phi'_{z_t} + b_x L_t m_t\phi'_{z_t})^T q_z \tag{B.77}$$

$$m_{z_5} = (-(M_{yz}\rho A) - m_t\phi_{y_t}\phi_{z_t} + b_x^2 m_t\phi'_{y_t}\phi'_{z_t})^T q_z \tag{B.78}$$

$$m_{z_6} = (-(N_{xz}\rho A) - b_x m_t\phi_{x_t}\phi_{z_t} - L_t m_t\phi_{x_t}\phi_{z_t})^T q_z \tag{B.79}$$

$$m_{z_7} = b_x m_t\phi_{x_t}\phi'_{z_t})^T q_z \tag{B.80}$$

$$m_{z_8} = -(M_{zz}\rho A) - m_t\phi_{z_t}\phi_{z_t} + b_x m_t(\phi_{z_t}\phi'_{z_t} + \phi'_{z_t}\phi_{z_t}) - b_x^2 m_t\phi'_{z_t}\phi'_{z_t})^T q_z \tag{B.81}$$

$$m_{z_9} = -(N_{xz}\rho A) - b_x m_t\phi_{x_t}\phi_{z_t} - L_t m_t\phi_{x_t}\phi_{z_t} + b_x^2 m_t\phi_{x_t}\phi'_{z_t} + b_x L_t m_t\phi_{x_t}\phi'_{z_t})^T q_z \tag{B.82}$$

$$m_{\dot{x}_1} = (-(b_x m_t\phi_{x_t}\phi'_{z_t}))^T \dot{q}_x \tag{B.83}$$

$$m_{\dot{x}_2} = (b_x m_t\phi_{x_t}\phi'_{y_t})^T \dot{q}_x \tag{B.84}$$

$$m_{\dot{x}_3} = (-(M'_{xy}\rho I_y) - I_{y_t}\phi_{x_t}\phi'_{y_t} - b_x^2 m_t\phi_{x_t}\phi'_{y_t} - b_x L_t m_t\phi_{x_t}\phi'_{y_t})^T \dot{q}_x \tag{B.85}$$

$$m_{\dot{x}_4} = (-(M'_{xz}\rho I_z) - I_{z_t}\phi_{x_t}\phi'_{z_t} - b_x^2 m_t\phi_{x_t}\phi'_{z_t} - b_x L_t m_t\phi_{x_t}\phi'_{z_t})^T \dot{q}_x \tag{B.86}$$

$$m_{\dot{y}_1} = (-(b_x m_t\phi_{x_t}\phi'_{y_t}))^T \dot{q}_y \tag{B.87}$$

$$m_{\dot{y}_2} = (2M_y\rho A + 2m_t\phi_{y_t} + 2b_x m_t\phi'_{y_t})^T \dot{q}_y \tag{B.88}$$

$$m_{\dot{y}_3} = (-(N_y\rho A) - b_x m_t\phi_{y_t} - L_t m_t\phi_{y_t} - b_x^2 m_t\phi'_{y_t} - b_x L_t m_t\phi'_{y_t})^T \dot{q}_y \tag{B.89}$$

$$m_{\dot{y}_4} = (2M_{yz}\rho A + 2m_t\phi_{y_t}\phi_{z_t} + 2b_x m_t\phi'_{y_t}\phi_{z_t})^T \dot{q}_y \tag{B.90}$$

$$m_{\dot{y}_5} = (2M'_{xy}\rho I_y + 2I_{y_t}\phi_{x_t}\phi'_{y_t} + b_x^2 m_t\phi_{x_t}\phi'_{y_t} + b_x L_t m_t\phi_{x_t}\phi'_{y_t})^T \dot{q}_y \tag{B.91}$$

$$m_{\dot{z}_1} = (-2b_x m_t \phi'_{z_t})^T \dot{q}_z \tag{B.92}$$

$$m_{\dot{z}_2} = (b_x m_t \phi_{x_t} \phi'_{z_t})^T \dot{q}_z \tag{B.93}$$

$$m_{\dot{z}_3} = (-2M_z \rho A - 2m_t \phi_{z_t})^T \dot{q}_z \tag{B.94}$$

$$m_{\dot{z}_4} = (-(N_z \rho A) - b_x m_t \phi_{z_t} - L_t m_t \phi_{z_t} + b_x^2 m_t \phi'_{z_t} + b_x L_t m_t \phi'_{z_t})^T \dot{q}_z \tag{B.95}$$

$$m_{\dot{z}_5} = (-2M_{yz} \rho A - 2m_t \phi_{y_t} \phi_{z_t} - 2b_x m_t \phi'_{y_t} \phi_{z_t})^T \dot{q}_z \tag{B.96}$$

$$m_{\dot{z}_6} = (2M'_{xz} \rho I_z + 2I_{z_t} \phi_{x_t} \phi'_{z_t} + b_x^2 m_t \phi_{x_t} \phi'_{z_t} + b_x L_t m_t \phi_{x_t} \phi'_{z_t})^T \dot{q}_z \tag{B.97}$$

$$w_{x_t 1} = (-\phi_{x_t})^T q_x \tag{B.98}$$

$$w_{x_t 2} = (-(b_x \phi_{x_t}) - L_t \phi_{x_t})^T q_x \tag{B.99}$$

$$w_{y_t 1} = (-\phi'_{y_t})^T q_y \tag{B.100}$$

$$w_{y_t 2} = (\phi_{y_t} + b_x \phi'_{y_t})^T q_y \tag{B.101}$$

$$w_{y_t 3} = (-\phi_{y_t} + L_t \phi'_{y_t})^T q_y \tag{B.102}$$

$$w_{z_t 1} = (\phi'_{z_t})^T q_z \tag{B.103}$$

$$w_{z_t 2} = (-\phi_{z_t} + b_x \phi'_{z_t})^T q_z \tag{B.104}$$

$$w_{z_t 3} = (\phi_{z_t} + L_t \phi'_{z_t})^T q_z \tag{B.105}$$

$$w_{\dot{x}_t 1} = (b_x \phi_{x_t})^T \dot{q}_x \tag{B.106}$$

$$w_{\dot{y}_t 1} = (2\phi_{y_t} + b_x \phi'_{y_t})^T \dot{q}_y \tag{B.107}$$

$$w_{\dot{y}_t 2} = (-2\phi_{y_t} - 2b_x \phi'_{y_t})^T \dot{q}_y \tag{B.108}$$

$$w_{y_t 4} = (\phi_{y_t} + 2b_x \phi'_{y_t} + L_t \phi'_{y_t})^T q_y \tag{B.109}$$

$$w_{y_t 5} = (b_x \phi'_{y_t} + L_t \phi'_{y_t})^T q_y \tag{B.110}$$

$$w_{\dot{z}_t 1} = (-2\phi_{z_t} - b_x \phi'_{z_t})^T \dot{q}_z \tag{B.111}$$

$$w_{\dot{z}_t 2} = (2\phi_{z_t} + 2b_x \phi'_{z_t})^T \dot{q}_z \tag{B.112}$$

$$w_{z_t 4} = (\phi_{z_t} - 2b_x \phi'_{z_t} - L_t \phi'_{z_t})^T q_z \tag{B.113}$$

$$w_{z_t 5} = (-(b_x \phi'_{z_t}) - L_t \phi'_{z_t})^T q_z \tag{B.114}$$

$$w_{y_t 6} = (\phi_{y_t} \phi'_{y_t})^T q_y \tag{B.115}$$

$$w_{y_t 7} = (\phi_{x_t}\phi_{y_t})^T q_y \tag{B.116}$$

$$w_{y_t 8} = (-(\phi'_{y_t}\phi'_{z_t}))^T q_y \tag{B.117}$$

$$w_{y_t 9} = (-(\phi_{x_t}\phi'_{y_t}))^T q_y \tag{B.118}$$

$$w_{z_t 6} = (-(\phi_{z_t}\phi'_{z_t}))^T q_z \tag{B.119}$$

$$w_{z_t 7} = (-(\phi_{x_t}\phi_{z_t}))^T q_z \tag{B.120}$$

$$w_{z_t 8} = (-(\phi'_{y_t}\phi'_{z_t}))^T q_z \tag{B.121}$$

$$w_{z_t 9} = (\phi_{x_t}\phi'_{z_t})^T q_z \tag{B.122}$$

$$M = \begin{bmatrix} m_7 & m_{y_6} & m_{x_3} & m_8 \\ m_{y_6} & m_{11} & 0 & 0 \\ m_{x_3} & 0 & m_{12} & 0 \\ m_8 & 0 & 0 & m_{13} \end{bmatrix} \tag{B.123}$$

$$V = \begin{bmatrix} -m_{y_1} - m_{\dot{y}_3}\dot{\theta}_x - m_{\dot{z}_1}\dot{x} - m_2\dot{\theta}_x\dot{y} - m_{z_1}\dot{\theta}_z\dot{y} - m_{y_1}\dot{\theta}_z\dot{z} \\ -(m_{\dot{y}_5}\dot{\theta}_y) - m_{z9}\dot{\theta}_x\dot{\theta}_y - m_{x9}\dot{\theta}_y^2 - m_{\dot{z}_6}\dot{\theta}_z - m_{y9}\dot{\theta}_x\dot{\theta}_z - m_{x9}\dot{\theta}_z^2 - \\ m_{y7}\dot{\theta}_y\dot{x} - m_{z7}\dot{\theta}_z\dot{x} - m_{\dot{z}2}\dot{y} + m_{y7}\dot{\theta}_x\dot{y} - m_{x6}\dot{\theta}_z\dot{y} - m_{\dot{y}1}\dot{z} + \\ m_{z7}\dot{\theta}_x\dot{z} + m_{x6}\dot{\theta}_y\dot{z} \\ -(m_{\dot{z}5}\dot{\theta}_x) - m_{y8}\dot{\theta}_x^2 - m_{\dot{x}3}\dot{\theta}_y - v_2\dot{\theta}_x\dot{\theta}_y - m_{x8}\dot{\theta}_x\dot{\theta}_z + m_{z5}\dot{\theta}_y\dot{\theta}_z - \\ m_{y8}\dot{\theta}_z^2 - m_{y2}\dot{\theta}_z\dot{y} - m_{\dot{x}2}\dot{z} + m_{y2}\dot{\theta}_y\dot{z} \\ -(m_{\dot{y}4}\dot{\theta}_x) - m_{z8}\dot{\theta}_x^2 - m_{x7}\dot{\theta}_x\dot{\theta}_y - m_{z8}\dot{\theta}_y^2 - m_{\dot{x}4}\dot{\theta}_z - v_3\dot{\theta}_x\dot{\theta}_z - \\ m_{y10}\dot{\theta}_y\dot{\theta}_z - v_1\dot{\theta}_y\dot{x} - m_{x5}\dot{\theta}_z\dot{x} - m_{\dot{x}1}\dot{y} + v_1\dot{\theta}_x\dot{y} - m_{z2}\dot{\theta}_z\dot{y} + \\ m_{x5}\dot{\theta}_x\dot{z} + m_{z2}\dot{\theta}_y\dot{z} \end{bmatrix} \tag{B.124}$$

$$B_a = \begin{bmatrix} -m_{z1} & 0 & m_2 & -m_{y4} & 0 \\ 0 & -m_{z3} & -m_{y3} & -m_6 & -m_{z6} \\ -m_{y2} & -m_3 & -m_{x2} & -m_{z5} & -m_{10} \\ -m_{z2} & -m_{x1} & -m_4 & -m_{y5} & -m_{x4} \end{bmatrix} \tag{B.125}$$

$$B_F = \begin{bmatrix} w_{z_t 3} & w_{x_t 2} & b_1 & -w_{y_t 1} & w_{x_t 1} & 1 & 1 \\ 0 & 0 & 0 & b_2 & 0 & 0 & 0 \\ 0 & b_3 & 0 & 0 & b_4 & 0 & 0 \\ 0 & 0 & b_5 & 0 & 0 & 0 & b_6 \end{bmatrix} \tag{B.126}$$

$$H_F = \begin{bmatrix} -(m_{\dot{z}3}\dot{\theta}_y) - m_{y1}\dot{\theta}_x\dot{\theta}_y - m_2\dot{\theta}_y^2 - m_{\dot{y}2}\dot{\theta}_z + m_{z1}\dot{\theta}_x\dot{\theta}_z - m_2\dot{\theta}_z^2 \\ m_{\dot{z}3}\dot{\theta}_x + m_{y1}\dot{\theta}_x^2 + m_2\dot{\theta}_x\dot{\theta}_y + m_{z1}\dot{\theta}_y\dot{\theta}_z + m_{y1}\dot{\theta}_z^2 \\ m_{\dot{y}2}\dot{\theta}_x - m_{z1}\dot{\theta}_x^2 - m_{z1}\dot{\theta}_y^2 + m_2\dot{\theta}_x\dot{\theta}_z - m_{y1}\dot{\theta}_y\dot{\theta}_z \\ m_{\dot{y}3}\dot{\theta}_y + m_{\dot{z}4}\dot{\theta}_z - m_{y1}\dot{\theta}_y\dot{x} + m_{z1}\dot{\theta}_z\dot{x} - m_{\dot{z}1}\dot{y} - m_2\dot{\theta}_y\dot{y} - m_2\dot{\theta}_z\dot{z} \\ -m_{z1} + m_{\dot{z}4}\dot{\theta}_x - m_{z1}\dot{\theta}_y\dot{y} + m_2\dot{\theta}_x\dot{z} - m_{y1}\dot{\theta}_y\dot{z} \end{bmatrix} \tag{B.127}$$

$$G_{Fa} = \begin{bmatrix} m_1 & 0 & 0 & 0 & m_{y1} \\ 0 & m_1 & 0 & -m_{z1} & m_2 \\ 0 & 0 & m_1 & -m_{y1} & 0 \\ 0 & -m_{z1} & -m_{y1} & m_5 & m_{z4} \\ m_{y1} & m_2 & 0 & m_{z4} & m_9 \end{bmatrix} \tag{B.128}$$

$$G_{FF} = \begin{bmatrix} -1 & -w_{y_t1} & -w_{z_t1} & 0 & 0 & 0 & 0 \\ w_{y_t1} & -1 & -w_{x_t1} & 0 & 0 & 0 & 0 \\ w_{z_t1} & w_{x_t1} & -1 & 0 & 0 & 0 & 0 \\ 0 & -w_{z_t2} & -w_{y_t2} & -1 & -w_{z_t1} & 0 & -w_{y_t1} \\ -w_{y_t3} & b_1 & -w_{x_t2} & w_{z_t1} & -1 & 0 & w_{x_t1} \end{bmatrix} \tag{B.129}$$

$$G_F = \begin{bmatrix} m_{z1} & 0 & m_{y2} & m_{z2} \\ 0 & m_{z3} & m_3 & m_{x1} \\ -m_2 & m_{y3} & m_{x2} & m_4 \\ m_{y4} & m_6 & m_{z5} & m_{y5} \\ 0 & m_{z6} & m_{10} & m_{x4} \end{bmatrix} \tag{B.130}$$

$$H_a = \begin{bmatrix} w_{\dot{z}_t2}\dot{\theta}_y + w_{y_t4}\dot{\theta}_x\dot{\theta}_y + b_1\dot{\theta}_y^2 + w_{\dot{y}_t2}\dot{\theta}_z + w_{z_t4}\dot{\theta}_x\dot{\theta}_z + b_1\dot{\theta}_z^2 \\ -w_{y_t2} + w_{\dot{z}_t1}\dot{\theta}_x + w_{\dot{x}_t1}\dot{\theta}_y - b_1\dot{\theta}_x\dot{\theta}_y + w_{y_t5}\dot{\theta}_y^2 - w_{x_t2}\dot{\theta}_x\dot{\theta}_z - w_{z_t2}\dot{\theta}_y\dot{\theta}_z + w_{y_t3}\dot{\theta}_z^2 \\ w_{z_t2} + w_{\dot{y}_t1}\dot{\theta}_x + w_{x_t2}\dot{\theta}_x\dot{\theta}_y - w_{z_t3}\dot{\theta}_y^2 + w_{\dot{x}_t1}\dot{\theta}_z - b_1\dot{\theta}_x\dot{\theta}_z + w_{y_t2}\dot{\theta}_y\dot{\theta}_z + w_{z_t5}\dot{\theta}_z^2 \\ 0 \\ 0 \end{bmatrix} \tag{B.131}$$

$$G_{aa} = \begin{bmatrix} 1 & -w_{y_t1} & -w_{z_t1} & 0 & w_{y_t3} \\ w_{y_t1} & 1 & -w_{x_t1} & w_{z_t2} & -b_1 \\ w_{z_t1} & w_{x_t1} & 1 & w_{y_t2} & w_{x_t2} \\ 0 & 0 & 0 & 1 & -w_{z_t1} \\ 0 & 0 & 0 & w_{z_t1} & 1 \end{bmatrix} \tag{B.132}$$

$$G_a = \begin{bmatrix} w_{z_t3} & 0 & w_{y_t6} & w_{z_t6} \\ w_{x_t2} & w_{z_t7} & b_3 & 0 \\ b_1 & w_{y_t7} & 0 & b_5 \\ -w_{y_t1} & b_2 & w_{z_t8} & w_{y_t8} \\ w_{x_t1} & w_{z_t9} & b_4 & 0 \end{bmatrix} \tag{B.133}$$

$$H_r = \begin{bmatrix} w_{z_t3}\dot{\theta}_y + w_{y_t3}\dot{\theta}_z - w_{y_t1}\dot{y} - w_{z_t1}\dot{z} \\ w_{y_t1} + b_3\dot{q}_y + w_{z_t2}\dot{\theta}_x + w_{x_t2}\dot{\theta}_y - b_1\dot{\theta}_z + \dot{y} - w_{x_t1}\dot{z} \\ w_{z_t1} + b_5\dot{q}_z + w_{y_t2}\dot{\theta}_x + b_1\dot{\theta}_y + w_{x_t2}\dot{\theta}_z + w_{x_t1}\dot{y} + \dot{z} \\ b_2\dot{q}_x + \dot{\theta}_x \\ b_6\dot{q}_z + \dot{\theta}_y - w_{x_t1}\dot{\theta}_z \\ b_4\dot{q}_y + w_{x_t1}\dot{\theta}_y + \dot{\theta}_z \end{bmatrix} \tag{B.134}$$

THESIS PROCESSING SLIP

FIXED FIELD: ill. ______ name ______

index ______ biblio ______

► COPIES: [✓ Archives] [✓ Aero] Dewey Eng Hum

Lindgren Music Rotch Science

TITLE VARIES: ►□ ______

NAME VARIES: ►☑ middle: ALLEN

IMPRINT: (COPYRIGHT) ______

►COLLATION: 243p

►ADD. DEGREE: ______ ► DEPT.: ______

SUPERVISORS: ______

NOTES:

cat'r: date:

page: ►F47

►DEPT: Aero

►YEAR: 1998, ►DEGREE: Ph.D.

►NAME: BLAUROCK, Carl