AN ABSTRACT ON THE THESIS OF

Brian P. Brisbine for the degree of Master of Science in Mechanical Engineering presented on August 11, 1998. Title: Computer-Aided Model Generation and Validation for Dynamic Systems.

Abstract approved: _____
Swavik A. Spiewak

The primary goal of any model is to emulate, as closely as possible, the desired behavioral phenomena of the real system but still maintain some tangible qualities between the parameters of the model and the system response. In keeping with this directive, models by their very nature migrate towards increasing complexity and hence quickly become tedious to construct and evaluate. In addition, it is sometimes necessary to employ several different analysis techniques on a particular system, which often requires modification of the model. As a result, the concept of versatile, step-wise automated model generation was realized as a means of transferring some of the laborious tasks of model derivation from the analyst to a suitable program algorithm. The focus of this research is on the construction and verification of an efficient modeling environment that captures the dynamic properties of the system and allows many different analysis techniques to be conveniently implemented. This is accomplished through the implementation of *Mathematica* by Wolfram Research, Inc..

The presented methodology utilizes rigid body, lumped parameter systems and Lagrange's energy formalism. The modeling environment facilitates versatility by allowing straightforward transformations of the model being developed to different forms and domains. The final results are symbolic expressions derived from the equations of motion. However, this approach is predicated upon the absence of significant low frequency flexible vibration modes in the system. This requirement can be well satisfied in the parallel structure machine tools, the main subject of this research.

The modeling environment allows a number of techniques for validation to be readily implemented. This includes intuitive checks at key points during model derivation as well as applications of more traditional experimental validation. In all presented cases the analysis can be performed in the same software package that was used for model development.

Integration of the generation, validation, and troubleshooting methodology delineated in this research facilitates development of accurate models that can be applied in structure design and exploitation. Possible applications of these models include parameter identification, visualization of vibration, automated supervision and monitoring, and design of advanced control strategies for minimization of dynamic tool path errors. The benefits are especially prevalent in parallel structure machine tools, where there is still a lack of experience. Latest developments in measurement techniques and the emergence of new sensors facilitate reliable validation and optimization of the models.

Computer-Aided Model Generation and Validation for Dynamic Systems

by

Brian P. Brisbine

A THESIS
submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented August 11, 1998
Commencement June 1999

Master of Science thesis of Brian P. Brisbine presented on August 11, 1998

APPROVED:

Major Professor, representing Mechanical Engineering

Chair of Department of Mechanical Engineering

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for Privacy

Brian P. Brisbine, Author

ACKNOWLEDGEMENT

I would like to express my appreciation to my advisor Dr. Swavik A. Spiewak. I first met Dr. Spiewak in his controls theory classes, where he introduced me to the power of high-level computer programs. I was particularly impressed with the active role he took in all of his students' research projects. Without his encouragement, advice, and tireless efforts, this thesis could not have been possible. I would also like to thank the students of my research group, Thomas Nickel and Thanat Jitpraphai for their support and assistance with many difficulties.

I would like to thank my colleague Sascha Weikert for sharing his expertise on high performance machining. His research concerning new parallel structure machines was one of the primary motivations for this thesis.

I would like to thank my parents, Glen E. Brisbine and Dr. Donna S. Houston for their continued support. I am especially indebted to my Mother Donna Houston, Grandmother Ruth Brisbine, and Brother David Brisbine, whose encouragement and support in many aspects allowed me to conduct this research.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

NOMENCLATURE

W	Action integral.
L	Lagrangian function.
q_i	i-th generalized coordinate.
$\dot{q}_i$	Time derivative of i-th generalized coordinate.
q_i^e	Extremum position of i-th generalized coordinate.
$q_{i,T}$	Global generalized coordinate for i-th body.
$q_{i,t}$	Local generalized coordinate for i-th body.
$\mathbf{q}_i$	Vector of generalized coordinates for i-th body.
$\mathbf{q}^g$	Global list of generalized coordinates.
δf	Perturbation of f about a particular point.
T	Total kinetic energy.
T_i	Total kinetic energy of i-th body.
$T_{i,rot}$	Rotational kinetic energy of i-th body.
$T_{i,trans}$	Translational kinetic energy of i-th body.
U	Total potential energy.
$U_{i,g}$	Potential energy of i-th body due to gravity.
U_{ij}	Potential energy of all SDEs between i-th and j-th bodies.
D_{ij}	Damping energy of all SDEs between i-th and j-th bodies.
Q_i	External force or torque associated with i-th generalized coordinate.
$\boldsymbol{\lambda}_{i,R}$	Rotational transformation matrix for i-th body.
$\mathbf{R}_\Theta$	Rotational transformation matrix about x-axis.
$\mathbf{T}_{XYZ}$	Translational transformation matrix.
$\boldsymbol{\lambda}_{i,TR}$	Homogeneous transformation matrix.
$\mathbf{I}_{i,r}$	Inertia tensor for i-th body.

NOMENCLATURE (Continued)

m_i	Mass of i-th body.
g	Acceleration due to gravity.
$l_{i,k}$	Elongation of SDE from i-th body movement.
$\mathbf{L}_i$	Vector of elongations of SDEs.
$\mathbf{K}_{SDE}$	Matrix of SDE stiffnesses.
$\mathbf{B}_{SDE}$	Matrix of SDE damping.
$\mathbf{m}$	Global mass matrix.
$\mathbf{c}$	Global damping matrix.
$\mathbf{k}$	Global stiffness matrix.
Δq_j	Small deformations of j-th generalized coordinate.
A_{jk}	Coefficient of 3rd term in Taylor Series expansion.
$\mathbf{x}(t)$	Vector of state variables.
$\mathbf{y}(t)$	Vector of outputs.
$\mathbf{u}(t)$	Vector of inputs.
$\mathbf{A}$	Evolution matrix.
$\mathbf{B}$	Controllability matrix.
$\mathbf{C}$	Observability matrix.
G_{ij}	Transfer function of i-th input, j-th output.
$\mathbf{G}$	Matrix of transfer functions.
$\mathbf{I}$	Identity matrix.
λ_i	i-th eigenvalue.

COMPUTER-AIDED MODEL GENERATION AND VALIDATION FOR DYNAMIC SYSTEMS

1. INTRODUCTION

The derivation of models is essential for the understanding of any dynamical system. This is especially prevalent in the realm of High Speed Machining (HSM), where the minimization of tool path errors necessitates optimal machine tool design and the implementation of advanced control strategies. Models, when carefully constructed, can provide a plethora of information pertaining to a particular system in many different aspects including, but not limited to, aid in system design, system analysis, parameter identification, implementation of control strategies, and dynamic visualization.

The primary goal of any model is to emulate, as closely as possible, the desired behavioral phenomena of the real system but still maintain some tangible qualities between the parameters of the model and the system response. In keeping with this directive, models by their very nature migrate towards increasing complexity and hence quickly become tedious to construct and evaluate. In addition, it is sometimes necessary to employ several different analysis techniques on a particular system, which often requires modification of the model. As a result, the concept of versatile, step-wise automated model generation was realized as a means of transferring some of the laborious tasks of model derivation and validation from the analyst to a suitable program algorithm. The focus of this research is on the construction and verification of an efficient modeling environment that captures the dynamic properties of systems and allows many different analysis techniques to be conveniently implemented.

1.1 Model Generation and Validation

Computer-Aided Model Generation and Validation (CAMGV) of dynamic systems is accomplished through the implementation of *Mathematica* by Wolfram Research, Inc., a software package capable of symbolic manipulation utilizing a unique 'problem solving' interface [Wolfram, 1991]. Although symbolic manipulation increases the

computational load, automated derivation (as compared to manual) is generally very proficient. This program was chosen for four main benefits: 1) the bulk of symbolic computations resides within the program, 2) expressions can be evaluated with infinite precision, 3) the model can be reconfigured easily without having to re-derive the base structure, and 4) the program architecture provides immediate adaptation to many different modeling domains and analysis techniques. The second and third benefits offer a distinct advantage over common Finite Element methods [Carne, et. al., 1988; Cheung, Leung, 1991; Fagan, 1992; Friswell, Mottershead, 1995; Weaver, Johnston, 1987; Weck, 1984], where machine precision must be taken into account, and the model has to be reconstructed and evaluated for each different set of parameters.

The mechanical representation of the general model consists of rigid bodies connected by spring damper elements, where the dynamic relationships between the forces and displacements are represented by lumped parameters. Lagrangian energy formalism is implemented to obtain the necessary equations that describe the model. The form of the model can then be used in either the (usually) nonlinear form, or it can be linearized for use in many analysis techniques. Use of the rigid body model and Lagrangian energy formalism is desirable since they naturally facilitate the casting of these tasks into an algorithm suitable for computer-aided model generation.

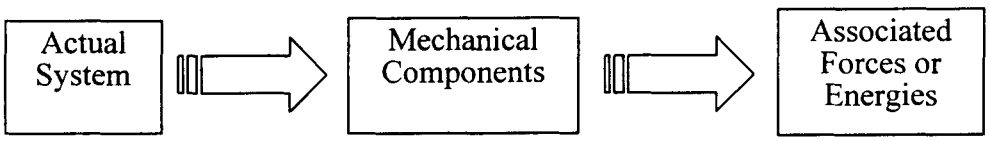


Fig. 1.1: Decomposition of system into definable quantities.

The approach just outlined reinforces the notion of automated modeling in that the implementation requires minimal user input. Most often, all that is required as input is definition of the coordinate systems and geometric information concerning orientation of the rigid bodies as well as their connection points. The program takes this information and generates the dynamic model, which can then be tailored for the desired analysis.

Throughout the course of this research one of the problems encountered was validation of the model in terms of accuracy and intuitive interpretation. It was found that the difficulties of even moderately complex models increased so rapidly that detection of algorithm or concept errors became a considerable challenge. As a result of this, a number of validation techniques were developed and implemented in an attempt to enhance the confidence level of the model structure.

1.2 Scope of Work

The research discussed here addresses two main issues concerning CAMGV. A concise, generalized form of model development is presented including a detailed description of the justifying principles used to generate the computer code. It is important to note that the model considers only rigid body motion at this time. Flexible mode vibrations are beyond the scope of this work. It is possible to emulate (to an extent) several dominant flexible modes by using a rigid body approximation, but that is not covered here. As a result of this, care must be taken when selecting which structures to model; reasonable results can only be expected when flexible mode vibrations are not a dominant factor in the dynamic behavior of the system under consideration.

The second important issue addressed in this research is validation of the model. Several methods have been integrated into the program to act as a pre-check of the model's accuracy. For example, certain quantities have a known value due to the nature of the system (e.g. potential energy at equilibrium), and this is checked in several places. In addition, several tests from controls theory can be applied[1] for comparison with physical intuition of a known configuration or system. Experimental verification is also an important aspect, and has been dealt with in this research. Tests using accelerometers were performed on a Stewart platform [Fichter, 1986], and the frequency domain results were compared with those obtained from the model.

The examples used for program demonstration and validation were done exclusively with parallel structure machine tools [Heisel, 1996; Codourey, Burdet, 1997]. These structures lend themselves for accurate description by lumped parameter systems, and the flexible mode vibrations are of a high enough frequency to be of little relevance.

---

[1] E.g. natural frequencies, damping ratios, mode shapes, and stability.

1.3 Chapter Overview

Most frequently used automated modeling approaches are introduced in chapter 2. These discussions include the most common methodologies, particularly those concerning distributed and lumped parameter systems and three different formalism techniques. A discussion of numeric vs. symbolic modeling is presented, with some concluding remarks concerning current model linearization techniques.

Underlying concepts used in CAMGV and analysis are delineated in Chapter 3. This includes a description of the general model structure, Lagrangian energy formalism, associated system energies, and applied linearization procedures. A discussion of the transformation of the model to State Variable form [DeCarlo, 1989; Franklin, et. al., 1994] is presented as an example of an analysis technique. The chapter concludes with intuitive techniques for verifying the accuracy of the generated model. Theoretical checks integrated into the program at various stages are explained, including the art of physical intuition as a means of validation. Comparisons with another model developed by the Institute for Machine Tools and Manufacturing at the Swiss Federal Institute of Technology [Weikert, et. al., 1998] are discussed. Several techniques for experimental validation are described in Chapter 4. Tests conducted on a Stewart platform using accelerometers are presented and compared to the generated models.

Several applications of these models are outlined in Chapter 5, including minimization of dynamic tool path errors, applications to parallel structure machine tools, parameter identification, visualization of vibrations, and automated supervision[2]. Chapter 6 concludes with a discussion of the accomplishments of this work and directions for future research.

---

[2] Control, monitoring, and diagnosis.

2. MODELING APPROACHES

The concept of CAMGV, both numeric and symbolic, can be found in literature dating back a quarter of a century. Programs for simulating motion of multi-body systems have been advantageous in the fields of robotics, machine design, and spacecraft dynamics for many years [Schaechter and Levinson, 1988]. Progress has occurred virtually independently of one another in these fields, and as a result, many different approaches have been successfully implemented. The following chapter outlines the structural configuration and dynamical theories used in several automated modeling methods.

2.1 Basic Building Blocks

The types of models dealt with here are abstract mathematical models, where the system is described by partial or ordinary differential equations. It is important to note that no system can be modeled exactly; inclusion of all the parameters affecting a particular system would be impossible to construct and analyze [Karnopp and Rosenberg, 1975]. Indeed, one of the most difficult aspects of modeling is knowing what to ignore [Amsterdam, 1991]. Conversely, if the model were too simplified, important information about the system would be lost. It is also necessary to focus on only the pertinent parameters affecting the desired response. Hence, a balance must be reached in that the model must retain simplicity so as not to clutter the results and cloud the analysis, but must also be of sufficient complexity to accurately exhibit the system's behavior.

The focus of this work is on modeling of the dynamic behavior of structures without the influence of external processes (e.g., cutting forces in machine tools). The dynamic motions of interest involve movements of one component in the structure relative to another. For example, in machining processes, the primary factor of concern is the interface between the tool and workpiece. To that regard, the first step required is disassembly of the actual system into describable components in terms of dynamics. This is accomplished either by the use of a distributed parameter system, a lumped parameter system, or a combination of both.

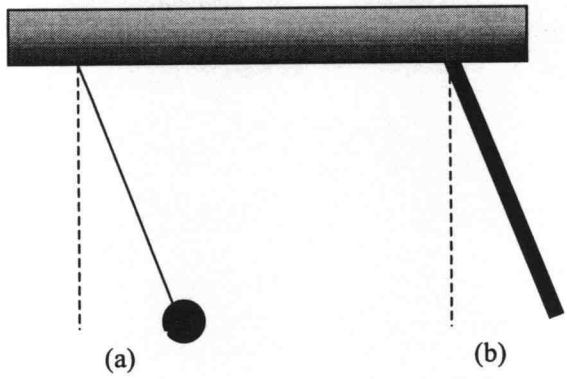


Fig. 2.1: Pendulums with (a) lumped and (b) distributed masses.

The choice of which to use depends on the system and what type of information is sought. A system dominated by flexible components would require a distributed parameter approach; one dominated by rigid components would require a lumped parameter approach; and of course a combination would require the use of both methods.

2.1.1 Distributed Parameter System

Distributed, or continuous parameter methods are typically used wherever the deformation of a body plays a significant role in the behavior being analyzed. These models are characterized by partial differential equations, can be highly nonlinear, and in most cases are computationally intensive. The primary advantages of this method is that it is the most accurate representation of the actual system, and produces exact closed form solutions in simple cases such as simply supported or cantilever beams.

Since these structures are truly "continuous", they possess an infinite number of degrees of freedom [Thomson, 1981]. For example, exciting a simply supported flexible machine tool spindle with continuous mass and elasticity distributions can result in any of an infinite number of mode shapes. Comparin, in his M.S. Thesis, used modal synthesis techniques to analyze such a spindle provided by General Motors [Comparin, 1983]. The spindle was part of a machine tool responsible for grinding valve seats in cylinder heads, and was subject to chatter problems due to the flexible modes of the spindle.

Fig. 2.2: Simply supported flexible spindle with excited second mode.

Although the use of partial differential equations provides an excellent description of the system, it does not always produce closed form solutions for complex shapes and/or multiple bodies in the system. However, since in most cases the dominant modes are the lowest few (Comparin found only the first three to be relevant), these mode shapes can be approximated by a polynomial fit for the spindle deflection [Ewins, 1984]. The result is a set of ordinary differential equations in place of a set of partial differential equations [Shabana, 1991].

The next question is the choice of polynomial to use. Indeed, as structural shapes increase in complexity, the choice of polynomial fit becomes obscure. This problem can be remedied by the use of Finite Element methods. With these methods the structure is divided into simpler elements, and the deformations within each element are described by interpolating polynomials [Shabana, 1991]. These methods are often successfully implemented with good accuracy where body deformations are of a concern, but are by their very nature computation intensive and exclusively numerical [Carne, et. al., 1988; Cheung and Leung, 1991; Fagan, 1992; Friswell and Mottershead, 1995; Weaver and Johnston, 1987; Gysin; Zatarain, 1998; for spindles, Reddy and Sharan, 1987; Comparin, 1983; for machine tools, Bianchi and Paolucci, 1996; Weck, 1984].

2.1.2 Lumped Parameter System

The use of continuous distribution of parameters is essential in situations where the flexible modes of a particular body are significant. But there also exist a large number of systems where deformations of bodies within the system have little impact on the overall dynamic behavior of the structure. In such cases a much simpler method can be applied that still retains sufficient accuracy. For these systems, the dominant source of dynamic

behavior is the elastic coupling between individual components, rather than deformations of the components themselves, and hence the components are emulated as rigid bodies. This leads to a simplified or lumped parameter description of components in the structure. Typically masses and associated inertia are all that is required to completely specify a rigid body element. Likewise, connections between the rigid bodies consist of linear, elastic elements that can be described by scalar quantities.

One example of such a structure is the spindle housing system shown in Fig. 2.3 [Aini, et. al., 1990; Matsubara, 1988; Shin, et. al., 1990; Spiewak, 1995; Weikert, et. al., 1997]. In machining processes, low to medium frequency dynamics of the structure play a critical role in tool path errors. This can be successfully modeled by the use of lumped parameters, since the housing and spindle structures are of sufficient rigidity[3] such that their flexible modes (usually high frequency) have little influence on the dynamic frequency range of interest [Comparin, 1983; Weck, 1984; Weikert, et. al., 1998].

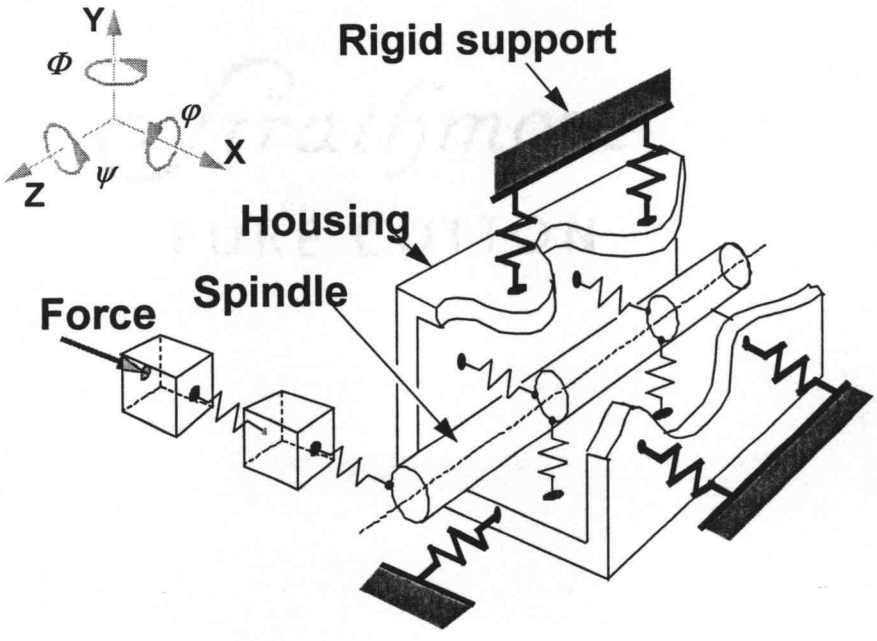


Fig. 2.3: Rigid body model of spindle housing structure.

[3] Short, stocky spindles are especially suited for this method.

The spindle and housing are coupled by a pair of bearings, which have a stiffness and damping associated with them. In addition, the housing is fastened to the inertial (fixed) base by means of elastic fasteners. Since the masses of the bearings and fasteners are considerably less than those of the spindle and housing, then omission of these masses still provides a good approximation of the actual structure. In addition, most of the low frequency deformations in the structure occur at the bearings and fasteners.

Some obvious advantages of the lumped parameter method include a reduced number of generalized coordinates, use of ordinary differential equations (as opposed to partial), simplified computations, and most importantly, the existence of closed form solutions[4]. This last point is essential for the application of controls analysis since system properties (parameters) are a part of the closed form solutions.

These characteristics make it appealing to attempt inclusion of flexible modes in this method, and it can be done to an extent. If it is known that an element's lowest flexible modes are a significant factor in the dynamics of the structure, then in some cases it is possible to model this by splitting the element into several coupled rigid bodies capable of assuming the required mode shape(s) [Ewins, 1984]. For the mode shape of Fig. 2.2, the spindle must be decomposed into at least four separate rigid bodies; more divisions can be added if necessary.

Fig. 2.4: Rigid body approximation of spindle mode shape in Fig. 2.2.

Although this allows the inclusion of "flexibility" in the rigid body model, it has limited accuracy and for reasons of practicality can only be used for simple mode shapes.

The concept of a rigid body is an artificial one, since all elements deform to some degree when subjected to a force [Ginsberg, 1995]. But the use of rigid bodies, where applicable, can be used to simplify equations and computations considerably. In cases

---

[4] 'Closed form' in this context pertains to obtainable results from controls theory.

where there exists both dominant flexible modes and rigid body motion, it is useful to combine both methods [Weck, 1984]. In addition, formulation of the equations of motion for deformable bodies often finds it convenient to separate out the rigid body and deformational contributions from the overall motion [Ginsberg, 1995; Marion and Thornton, 1988; Weck, 1984].

2.2 Modeling Methodologies

Over the past two decades the trends underlying the derivation of dynamic models have changed dramatically. The early programs were general purpose and numerical in nature, which could accommodate a common class of structures and a wide variety of users, not necessarily just dynamicists [Schaechter and Levinson, 1988]. It soon became apparent that these programs suffered from two deficiencies; namely, that they were too slow, and they ran into difficulties when dealing with closed looped bodies or those of 'abnormal' configurations. This realization shifted the focus to special purpose, or highly customized programs that worked for only a very restricted type of geometric topology and required a thorough knowledge of the dynamics involved. The drawback of this was that the specialized programs had to be derived by hand for each system to be analyzed, which resulted in a multitude of inevitable errors. In addition to this, increasing complexities of systems made the programs impractical to formulate by hand.

One of the latest approaches is to use a combination of both general and special purpose methodologies, accomplished by the implementation of symbolic manipulation software. Most of these programs still require the dynamicist to set up the model, but much of the tedious tasks of derivation are carried out on the computer. This eliminates a majority of the human error due to carelessness, and is more time efficient.

Along with the overall methodology, there are also several diverse approaches of generating the usual second order equations of motion. Newton [Doeblin, 1980; Marion and Thornton, 1988], Lagrange [Ginsberg, 1995; Marion and Thornton, 1988; Scheck, 1994], and Kane [Kane and Levinson, 1985; Kane, et. al., 1983] developed the three most widely used formalisms. One is not superior over another; rather, each has a particular area where it works with optimal performance.

The first and most accepted method explicitly derives the equations of motion using Newton's Laws of Mechanics. The ordinary differential equations are found by Newton's

Second Law which equates the forces and torques acting on the bodies with the respective time derivatives of momentum quantities, producing the Newtonian equations of motion [Marion and Thornton, 1988]. This is the most straightforward and intuitive approach, and is often used whenever it is feasible. Unfortunately, most useful structures in industry are of sufficient complexity to warrant this as not a viable method to use; nevertheless, it is still important for model verification purposes at simpler configurations.

When the structure increases in complexity, then Lagrange's method becomes the preferred choice of formalism. In contrast to Newton's method, which is concerned with forces and torques, the Lagrangian method considers the energies (kinetic, potential, and dissipative) of the system. Although a little more abstract, the generated equations[5] are identical to Newton's approach only in a slightly different (or unsimplified) form [Rosenthal and Sherman, 1986]. The main advantage of this method lies in the fact that it is generally much easier to define the energies of a particular system than to define the individual forces acting on each body. Definition of the individual forces can become very taxing in some cases, particularly in the structures considered in this research.

One alternative to Lagrange's energy formalism is Kane's method. Rather than considering all the forces and torques acting on a body, Kane's method deals with *generalized* active and inertia forces [Kane, et. al., 1983]. The primary advantage of this (and the motivation for using generalized forces) is a simplification of the equations needed to describe the system, since some of the forces acting on the bodies contribute nothing to the generalized forces[6]. This is especially prevalent when dealing with rigid body dynamics. Equating the sum of respective generalized active and inertia forces to zero produces Kane's dynamical equations of motion. This method, in general, will produce the equations in the most compact form, implying that this is the easiest approach. However, one must keep in mind that there is also a set of associated kinematical equations that must be satisfied when using this method [Ginsberg, 1995].

Many programs implementing these methods are extensively documented in literature, but due to the versatility and relative ease of using the latter two (Lagrange and Kane), most of the most recent effort has focused on these [for Lagrange, Hong and Curtiss, 1993; Chen, 1996; Weikert, et. al., 1998; Nickel, 1998; for Kane, Rosenthal and Sherman, 1986; Faessler, 1986; Schaechter and Levinson, 1988]. Of particular interest

---

[5] Based on Hamilton's Principle.

[6] Due to force cancellation.

with respect to Lagrange's method is the use of bond graphs, another broad field that has been the focus of much research and documentation [Karnopp and Rosenberg, 1975; Amsterdam, 1992]. Bond graphs essentially consist of a simple abstract 'language' that describes the flow of energy within a system. This is especially useful for formulating the semantics and syntax rules for the description of physical systems, and since the description is in terms of energy, it is a natural segue to Lagrange's method.

The Lagrangian energy formalism was chosen in this work over Kane's method for several reasons. For programming purposes, system energies (both electrical and mechanical) are straightforward to define and manipulate using the unique symbolic problem-solving environment provided by *Mathematica*. It was also felt that the use of system energies would allow a broader class of generalized structures to be analyzed with minimal (human) effort. Although this is not as efficient and does not produce the most compact equations, the systems under consideration have a very manageable number of degrees of freedom and hence the use of unsimplified equations and slightly extended computation time is not overly hindering. In addition to this, a general lack of experience in the use of Kane's method prevented an efficient implementation of this formalism, although it is a very viable option.

2.3 Numeric vs. Symbolic Models

Early automated modeling programs were exclusively numeric and required extensive algebraic manipulation of the equations during the setup process. It soon became apparent that computer-aided algebraic computations were the viable alternative to manual algebra since most of the tedious tasks were done automatically and much less prone to errors [Gaonkar et. al., 1990]. Numerical computations also suffer from a number of drawbacks, including:

1. repeated setup of the dynamic equations at each computation step or integration, resulting in excessive operations and extensive computation time,
2. difficulty in implementing control strategies in numerical equations, obstructing real time operations as required by some multi-body systems,
3. unclear physical insight into the system as a result of numerical expressions, and
4. equations of motion existing as only mathematical operations in the computer program [Lieh and Haque, 1991; Hale and Meirovitch, 1978].

Additionally, it has been found in this work and several others that the limits of machine precision have forced the program to repeatedly compute many terms that, when analyzed symbolically, were zero [Rosenthal and Sherman, 1986; Nickel, 1998]. It must be noted that numeric algorithms are well established and can be highly accurate; but for the purposes of this work pertaining to the reasons above, numerical computation algorithms are not sufficient.

The first attempt at non-numeric computation was done by Levinson in 1977 using the FORMAC language to develop equations of motion for a specific multi-body system [Levinson, 1977]. Since then a number of symbolic manipulation software packages and programs have been utilized, with varying degrees of success. Several of these programs include SYMBOD [Macala, 1983], DYMIR [Cesareo and Nicolo, 1984], SD/EXACT [Rosenthal and Sherman, 1986], NEWEUL [Kreuzer and Schiehlen], and AUTOLEV [Schaechter and Levinson, 1988]. Some of the appealing advantages of using symbolic manipulation include:

1. infinite precision, since calculated values are not subjected to accumulated errors caused by limited machine precision,
2. one time model derivation, since iterative calculations only involve parameter value substitutions,
3. clear intuitive insight into the physical system,
4. straightforward control strategy implementation as a result of (3),
5. greater accuracy of estimating unknown parameters, and
6. ability to potentially produce closed form solutions, as opposed numeric computations which give iterative solutions.

Symbolic manipulation also alleviates (to an extent) the difficulties[7] that arise from simplifying assumptions made by manual derivations in an attempt to reduce the amount of tedious work required [Miller and White, 1987].

Due to storage limitations and to maintain optimal program efficiency, symbolic formalisms must be based on proficient dynamic principles [Kreuzer and Schiehlen]. Kane's method and Lagrangian energy formalism are well suited for this task.

---

[7] Simplifying assumptions can result in a poor representation of the system.

2.4 Model Linearization

Most systems in nature are highly nonlinear. Unfortunately, nonlinear analysis is very difficult not only because the system is complex and exhibits exponential error propagation (chaos), but also because there are very few control subroutines available for nonlinear analysis [Lieh and Haque, 1991]. As a result, an important aspect of any modeling program is effective and accurate linearization of the system where appropriate. Since complicated multi-body structures can be sufficiently represented by low order (linear) approximations to problems with a high number of degrees of freedom, several approaches can be taken to produce linear equations from nonlinear systems [Hale and Meirovitch, 1978]. One method involves an orderly kinematical procedure and discretization/ truncation scheme that allows the potential and kinetic energy of each substructure to be written in a compact, linear matrix form [Hale and Meirovitch, 1978]. Another method uses the standard first order expansion of trigonometric functions (arising from coordinate transformations), and then searches for and eliminates all higher order nonlinearities [Lieh and Haque, 1991]. Several attempts have been done to make linearization more efficient. Miller and White used an innovative approach by writing all transformation matrices[8] as exponentials making differentiation, and hence linearization, easier [Miller and White, 1987].

For most systems, the movement of interest usually involves small displacements or rotations about a nominal or equilibrium position. This nominal position is not necessarily fixed, but can change with varying configurations of the system. For such a system, the most widely used method of linearization is a multi-variable Taylor Series expansion about the nominal position [Ginsberg, 1995; Marion and Thornton, 1988]. Some analysts choose to perform the expansion on the complete nonlinear equations of motion, while others perform the expansion at an earlier stage of equation development. For the Lagrangian energy formalism, the simplest form of linearization is accomplished by expanding the energies, which is also done in this work. Another advantage that arises

[8] Transformation matrices are necessary for coordinate transformations between reference frames. A detailed description of these can be found in Appendix A.

from the expansion of potential energy is a pre-check concerning verification of the model, which is explained in the next chapter.

3. MODEL GENERATION AND VALIDATION

Thorough understanding of the dynamic behavior of a particular structure often requires the use of many different analysis techniques. Some of these might include simulated response to an actual input, modal properties (natural frequencies and mode shapes) of the system, location of poles and zeros, tests for controllability and observability, inspection of complete nonlinear or linear equations of motion, as well as many others. In order to accommodate this, the generated dynamic model must take on different forms, some of which include standard linear and nonlinear equations of motion, input-output (transfer function) form, or state-variable form. The model must also be able to work in several domains, whether it be frequency, continuous time, Laplace, or discrete time. Consequently, the generated model and environment it is developed in must be versatile enough to facilitate mapping between different model forms and domains.

The chapter is divided into two subjects, namely model development and intuitive validation. Interactive sessions with the model in the *Mathematica* environment are illustrated throughout both subjects. The first six sections lay the foundation for implementation of the proposed modeling methodology. Description of the basic principles begins with a discussion of the generic structure of the system. Lagrangian energy formalism is then presented, with a discussion of the associated energies and how they are found from the generalized system. The chapter then describes the process used for linearization and outlines the steps for transformation into state-variable form, since this is the most common practice for analysis purposes.

The second subject of the chapter delineates several methods for intuitive validation. Many of these techniques take advantage of key properties inherent to the basic principles of the modeling methodology. In addition, validation is enhanced by the ability to transition between different model forms and domains, illustrated again by several examples from interactive *Mathematica* sessions.

3.1 Generalized Rigid Body System

The class of structures considered here are of the type where lumped parameter models can represent the dynamic relationships between forces and displacements. The most general case is an arbitrary collection of rigid bodies, connected by 'spring-damper-elements' (SDE). These SDEs have no constraint on the number or location of connections to the rigid bodies. The SDEs can represent, for example, bearing couplings such as shown in Fig. 2.3, component/fastener interfaces, or strut couplings in parallel structure machine tools. Each rigid body has 6 degrees of freedom, namely 3 translations and 3 rotations with respect to a chosen global reference frame[9]. For n rigid bodies the structure as a whole will have $6n$ degrees of freedom.

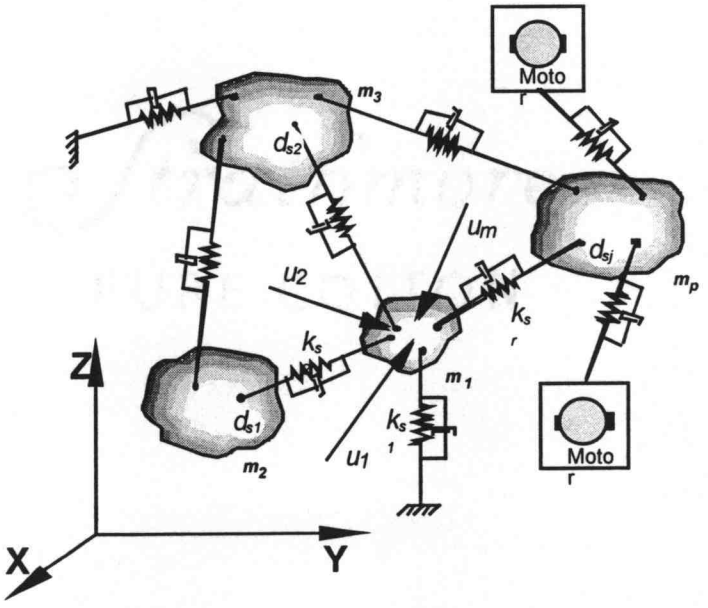


Fig. 3.1: A generalized multi-degree-of-freedom rigid body system.

[9] For this work, the 'global' reference frame is assumed to be inertial (fixed). However, the method can easily accommodate non-inertial reference frames.

Each rigid body contains a local reference frame with the origin located at the center of mass and the axes pointing along the principal axes [Marion and Thornton, 1988; Ginsberg, 1995] of the body. Any number of forces and torques can act upon the rigid bodies, such as those generated by actuators or the cutting process of machining, which can be either internal or external to the system. Main advantages of using the above model are as follows:

1. Efficient mapping between parameters of the actual system and their representations in the model,
2. Scalability and versatility of the model,
3. Good computational efficiency due to the model's parsimony [Box and Jenkins, 1976].

3.2 Lagrangian Energy Formalism

The next step after the generalized model structure has been defined is to derive the equations of motion. There are a number of approaches that can be implemented for this. Since the rigid body structure is completely arbitrary, a method capable of handling this in a reasonable fashion should be used. For arbitrary structures, defining the energies is generally more straightforward than defining the forces. It may not be the most efficient from a computational standpoint, but it assures maximum reliability and provides a means of intuitive inspection and interpretation. With that in mind, it is helpful to first take a more detailed look at the formulation of Lagrange's method.

The idea of a particle following a physical trajectory from point (1) to point (2) deals with the notion of stability. This was first expressed by Hamilton's principle of least action, where a perturbation on the particle's path results in a quantity that is stable for the real trajectory. This quantity generally manifests itself in the form of Lagrange's equations [Scheck, 1994; Marion and Thornton, 1988].

Suppose a conservative[10] mechanical system can be described by a Lagrange function, L, which is a function of all generalized coordinates $q_i(t)$, velocities $\dot{q}_i(t)$, and

[10] No external forces (or disturbances) act on the system [Marion, Thornton, 1988].

time. For all trajectories of this system that travel from point (1) to point (2) there exists an action[11] integral [Jansen, 1997; Scheck, 1994]

$$W = \int_{t_1}^{t_2} L(\{q_i(t)\},\{\dot{q}_i(t)\},t)dt \tag{3.1}$$

The physical (or actual) path that the system takes is that which results in an extremum for W. This requires finding a minimum for the multi-variable function L, and is accomplished by noting that all perturbations (in position) about the extremum position, $\{x_i^e\}$, must be equal to zero (in first order)

$$\delta f = f(\{x_i^e + \delta x_i\}) - f(\{x_i^e\}) = 0 \tag{3.2}$$

The perturbation about the physical trajectory of the Lagrangian system gives

$$\delta W = \int_{t_1}^{t_2} L(\{q_i^e(t) + \delta q_i(t)\},\{\dot{q}_i^e(t) + \delta \dot{q}_i(t)\},t)dt - \int_{t_1}^{t_2} L(\{q^e{}_i(t)\},\{\dot{q}^e{}_i(t)\},t)dt \tag{3.3}$$

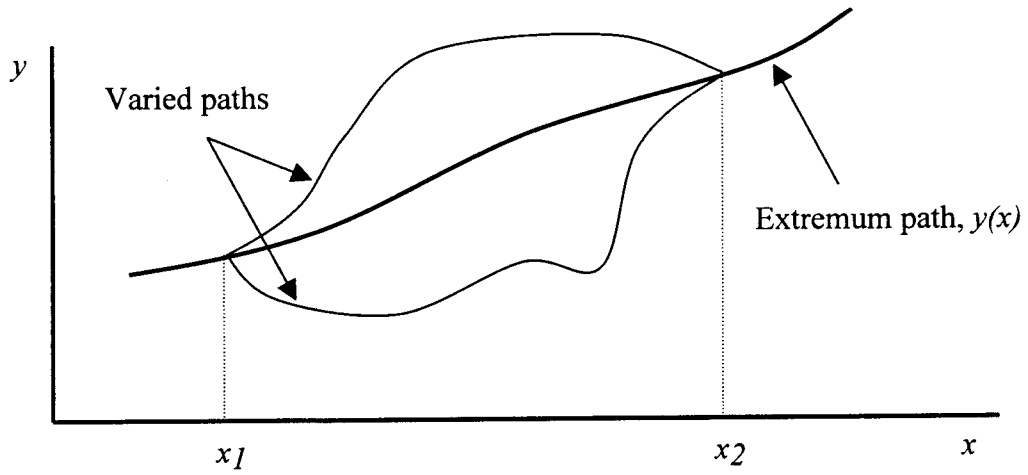


Fig. 3.2: Perturbations on the physical, or extremum, path [Marion and Thornton, 1988].

[11] The name *action* arises because L has the dimension of energy and the product (energy and time) is called action; hence, this is an action integral [Scheck, 1994].

Using the expansion

$$f(x_i + \delta x_i) = f(x_i) + \delta x_i \frac{\partial f}{\partial x_i} \tag{3.4}$$

The perturbed action integral becomes

$$\delta W = \int_{t_1}^{t_2} \left[\sum_j \frac{\partial L}{\partial q_j} \left(\{q_i^e(t)\}, \{\dot{q}_i^e(t)\}, t \right) \delta q_j(t) + \sum_j \frac{\partial L}{\partial \dot{q}_j} \left(\{q_i^e(t)\}, \{\dot{q}_i^e(t)\}, t \right) \delta \dot{q}_j(t) \right] dt \tag{3.5}$$

Noting that

$$\dot{\delta q}_i(t) = \frac{\partial}{\partial t} \delta q_i(t) \tag{3.6}$$

The second term can be integrated by parts to produce

$$\delta W = \int_{t_1}^{t_2} \left[\sum_j \frac{\partial L}{\partial q_j} \left(\{q_i^e(t)\}, \{\dot{q}_i^e(t)\}, t \right) - \sum_j \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} \left(\{q_i^e(t)\}, \{\dot{q}_i^e(t)\}, t \right) \right] \delta q_j(t) dt \tag{3.7}$$

Where the endpoints are evaluated to be zero (in the integration by parts) since by definition there is no perturbation of the endpoints. The perturbed action integral must be zero for all arbitrary perturbations $\delta q_j(t)$, and this only occurs when

$$\frac{\partial L}{\partial q_j} \left(\{q_i^e(t)\}, \{\dot{q}_i^e(t)\}, t \right) - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} \left(\{q_i^e(t)\}, \{\dot{q}_i^e(t)\}, t \right) = 0 \tag{3.8}$$

which must hold for all j. Hence, the physical trajectory of a conservative system is described by a set of Lagrange's equations of the first kind [Pandit, 1991]

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = 0 \tag{3.9}$$

The choice for L is not unique, but the natural choice (and the convention followed here) is to set

$$L = T - U \tag{3.10}$$

where T represents the kinetic energy and U represents the potential energy.

External forces acting on the system are taken into account by Lagrange's equations of the second kind [Pandit, 1991]

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = Q_i \tag{3.11}$$

Where Q_i represents the external forces or torques associated with the i-th generalized coordinate.

3.3 Associated System Energies

With the use of Lagrange's method, the focus is shifted from directly defining the equations of motion for a structure to finding the energies associated with the system. For conservative systems, only two quantities are of concern: kinetic and potential energy. However, since conservative systems[12] are primarily found in theory and rarely in reality, the inclusion of a dissipative, or damping, energy is necessary. For the model described in section 3.1, energy is lost or dissipated due to damping associated with the couplings of the rigid bodies, represented by dashpots in Fig. 3.1.

3.3.1 Kinetic Energy

Since the SDEs are assumed massless, then the only contributors to the kinetic energy are the movements of the rigid bodies themselves. Assuming the energy is defined about the center of mass, then the kinetic energy for the *i*-th rigid body in Fig. 3.1 can be conveniently separated into translational and rotational terms

$$T_i = T_{i,trans} + T_{i,rot} \tag{3.12}$$

The translational kinetic energy is simply written as

$$T_{i,trans}(\dot{\mathbf{q}}_{i,T}) = \frac{1}{2}\dot{\mathbf{q}}_{i,T}^T \mathbf{m}_i \dot{\mathbf{q}}_{i,T} \tag{3.13}$$

Where $\mathbf{m}_i$ is the diagonal mass matrix and $\dot{\mathbf{q}}_{i,T} = \{X_i, Y_i, Z_i\}$ is defined as the vector of generalized translational velocities[13] for the *i*-th rigid body in the global reference frame[14].

The rotational kinetic energy has a similar form, only it uses the *local* body inertia tensor and vector of generalized *local* rotational velocities, $\dot{\mathbf{q}}_{i,r} = \{\theta_i, \phi_i, \varphi_i\}$

$$T_{i,rot}(\dot{\mathbf{q}}_{i,r}) = \frac{1}{2}\dot{\mathbf{q}}_{i,r}^T \mathbf{I}_{i,r} \dot{\mathbf{q}}_{i,r} \tag{3.14}$$

---

[12] Referring to mechanical structures, which is the focus of this research.

[13] Time derivatives of the generalized translational coordinates.

[14] Global reference frames are denoted with upper case; local reference frames are denoted with lower case.

The inertia tensor, $\mathbf{I}_{i,r}$, will always be diagonal as long as the local coordinate axes correspond with the principal axes of inertia [Marion and Thornton, 1988; Ginsberg, 1995] for the body. This is not always the case, but it is generally less tedious to define the principal inertias (diagonal elements of the tensor) and transform to another configuration rather than fill in all the elements of the tensor for each change in orientation.

It is often useful to look at the rotations about the global axes instead of the local axes. In order to do this, a transformation[15] relating rotations in the local frame to rotations in the global frame is used

$$\mathbf{q}_{i,R} = \boldsymbol{\lambda}_{i,R}\mathbf{q}_{i,r} \tag{3.15}$$

where $\mathbf{q}_{i,R} = \{\Theta_i, \Phi_i, \Psi_i\}$. The simplest situation involves small displacements and/or rotations about an arbitrary configuration of the rigid body. For simplification purposes, the transformation matrix, $\boldsymbol{\lambda}_i$, can be assumed time independent, so

$$\dot{\mathbf{q}}_{i,R} = \boldsymbol{\lambda}_{i,R}\dot{\mathbf{q}}_{i,r} \tag{3.16}$$

[15] This is further discussed in Appendix A.

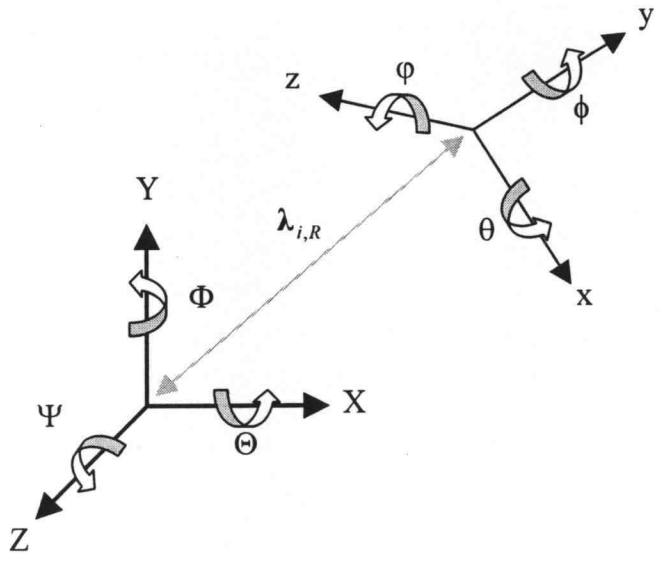


Fig. 3.3: Transformation, $\boldsymbol{\lambda}_{i,R}$, relating rotations between local and global coordinates.

Substitution of Eq. (3.16) into Eq. (3.14) gives the rotational kinetic energy for the *i*-th rigid body about the global axes

$$T_{i,rot}(\dot{\mathbf{q}}_{i,R}) = \frac{1}{2}\dot{\mathbf{q}}_{i,R}^{T}(\boldsymbol{\lambda}_{i,R}^{-1})^{T}\mathbf{I}_{i,r}\boldsymbol{\lambda}_{i,R}^{-1}\dot{\mathbf{q}}_{i,R} \tag{3.17}$$

This equation can be simplified by noting that for special cases the inverse is equivalent to the transpose for orthogonal transformation matrices [Marion and Thornton, 1988]

$$T_{i,rot}(\dot{\mathbf{q}}_{i,R}) = \frac{1}{2}\dot{\mathbf{q}}_{i,R}^{T}\boldsymbol{\lambda}_{i,R}\mathbf{I}_{i,r}\boldsymbol{\lambda}_{i,R}^{T}\dot{\mathbf{q}}_{i,R} \tag{3.18}$$

Eq. (3.18) expresses the kinetic energy in terms of the global rotational coordinates and utilizes the principal inertia values of the *i*-th rigid body (i.e., evaluated in local coordinates).

Substitution of Eq. (3.18) and Eq. (3.13) into Eq. (3.12) gives the total kinetic energy for the *i*-th rigid body with respect to generalized global coordinates (or rather, their time derivatives)

$$T_i(\dot{\mathbf{q}}_{i,T}, \dot{\mathbf{q}}_{i,R}) = \frac{1}{2}\dot{\mathbf{q}}_{i,T}^T \mathbf{m}_i \dot{\mathbf{q}}_{i,T} + \frac{1}{2}\dot{\mathbf{q}}_{i,R}^T \boldsymbol{\lambda}_{i,R} \mathbf{I}_{i,r} \boldsymbol{\lambda}_{i,R}^T \dot{\mathbf{q}}_{i,R} \tag{3.19}$$

The total kinetic energy for a system of *n* rigid bodies is simply the summation of the kinetic energies of each individual body

$$T = \sum_{i=1}^{n} T_i(\dot{\mathbf{q}}_{i,T}, \dot{\mathbf{q}}_{i,R}) \tag{3.20}$$

Provided the generalized coordinate, mass, inertia and transformation matrices are defined (either symbolically or numerically), calculation of the kinetic energy in *Mathematica* is accomplished with two lines:

```
Ti = 1/2 Transpose[qi'[t]] . mi . qi'[t] +
   1/2 Transpose[qi'[t]] . λi . Ii . Transpose[λi] . qi'[t];
T = Σ_{i=1}^{n} Ti;
```

Comparisons of these commands with equations (3.19) and (3.20) show that they are virtually identical. This is an important fact; the symbolic capabilities and problem-solving interface of Mathematica provides an immediate (and obvious) mapping between theoretical postulates and the actual program code. This makes the development, validation, and troubleshooting of models generated in this environment more straightforward.

3.3.2 Potential and Damping Energies

The model described in section 3.1 can store potential energy in two forms. The first form is gravitational potential energy, which is merely the vertical distance a rigid body is elevated from a chosen 'zero' point; this case being the origin of the global reference frame

$$U_{i,g} = m_i g Z_i \tag{3.21}$$

The second form of energy storage is in compression or tension of the connection springs. The elongation of any elastic element between bodies i and j that is due to motion of the i-th body can be written as

$$l_{i,k} = l_{i,k}(\mathbf{q}_i) \qquad k = 1,2,...,m \tag{3.22}$$

Where m is the number of SDEs connecting the two bodies. The elongations are defined as functions of generalized global coordinates, $\mathbf{q}_i = \{X_i, Y_i, Z_i, \Theta_i, \Phi_i, \Omega_i\}$, which are translations and rotations of the body i.

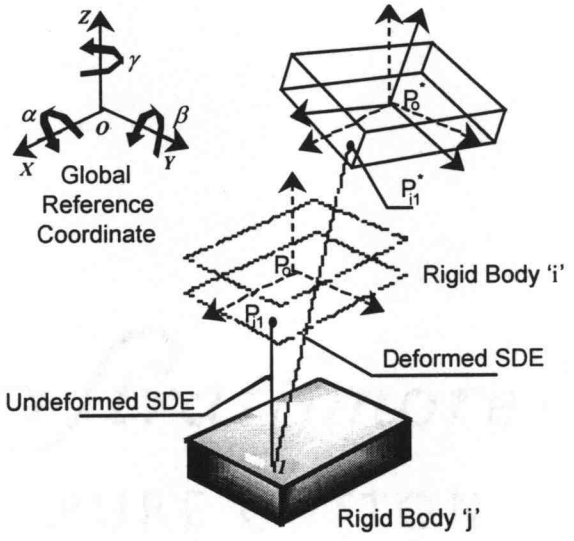


Fig. 3.4: Elongation of a SDE between two moving rigid bodies.

The elongations of all SDE couplings between the i-th and j-th bodies due to motion of the i-th body can be written in vector form

$$\mathbf{L}_i(\mathbf{q}_i) = [l_{i,1}, l_{i,2}, ..., l_{i;m}]^T \tag{3.23}$$

If a displacement of the j-th body is introduced in the same way, the potential energy due to deformations of all SDEs between the two bodies can be calculated as

$$U_{ij}(\mathbf{q}_i, \mathbf{q}_j) = \frac{1}{2}[\mathbf{L}_i - \mathbf{L}_j]^T \mathbf{K}_{SDE}[\mathbf{L}_i - \mathbf{L}_j] \tag{3.24}$$

Where $\mathbf{K}_{SDE}$ represents a diagonal matrix of stiffnesses of SDEs between the i-th and j-th bodies. Summing over all n rigid bodies in Equations (3.21) and (3.24) gives the total potential energy

$$U = \sum_{i=1}^{n}\left(U_{i,g} + \frac{1}{2}\sum_{j=1}^{n} U_{ij}(\mathbf{q}_i, \mathbf{q}_j) \right) \tag{3.25}$$

Similar to the kinetic energy, definition of potential energy in *Mathematica* requires three lines:

```
Ui,g = mi g Zi ;
Uij = 1/2 Transpose[Li - Lj] . KSDE . (Li - Lj) ;
U = Sum[ Ui,g + 1/2 Sum[Uij, {j,1,n}], {i,1,n} ] ;
```

Evaluation of the dissipation energy due to damping is exactly analogous to that of the potential energy, except the former is proportional to the deflection *velocity* of the SDE attachment points

$$D_{ij}(\dot{\mathbf{q}}_i, \dot{\mathbf{q}}_j) = \frac{1}{2}[\dot{\mathbf{L}}_i - \dot{\mathbf{L}}_j]^T \mathbf{B}_{SDE}[\dot{\mathbf{L}}_i - \dot{\mathbf{L}}_j] \tag{3.26}$$

Where $\mathbf{B}_{SDE}$ represents a diagonal matrix of damping of SDEs between the i-th and j-th bodies. Again, summation over all n bodies gives the total damping energy

$$D = \sum_{i=1}^{n}\sum_{j=1}^{n} D_{ij}(\dot{\mathbf{q}}_i, \dot{\mathbf{q}}_j) \tag{3.27}$$

3.4 The Equations of Motion

Substituting the Lagrangian of Eq. (3.10) into Eq. (3.11) gives the simplified result [Pandit, 1991]

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_i^g}\right) - \frac{\partial T}{\partial \mathbf{q}_i^g} + \frac{\partial U}{\partial \mathbf{q}_i^g} = Q_i \qquad i = 1,2,...,6n \tag{3.28}$$

where Q_i represents the external forces associated with the i-th generalized coordinate from the *global* list of generalized coordinates $\mathbf{q}^g$ representing all n bodies. Modifying this to include the damping energy is accomplished by the addition of another term [Pandit, 1991]

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\mathbf{q}}_i^g}\right) - \frac{\partial T}{\partial \mathbf{q}_i^g} + \frac{\partial U}{\partial \mathbf{q}_i^g} + \frac{\partial D}{\partial \dot{\mathbf{q}}_i^g} = Q_i \qquad i = 1,2,...,6n \quad (3.29)$$

Application of Eq. (3.29) produces the equations of motion for each generalized coordinate for the multi-degree-of-freedom system under consideration. Eq. (3.29) can also be immediately seen in the *Mathematica* code:

```
∂t (∂q'i T) - ∂qi T + ∂qi U + ∂qi D = Qi
```

The final result can be either a complete set of linear or nonlinear equations. For linear systems, the equations can be reduced to a general vector-matrix equation encapsulating motion of all rigid bodies

$$\mathbf{m}\ddot{\mathbf{q}}^g + \mathbf{c}\dot{\mathbf{q}}^g + \mathbf{k}\mathbf{q}^g = \mathbf{Q} \quad (3.30)$$

Where **m**, **c**, and **k** represent *global* mass[16], damping, and stiffness matrices, respectively.

3.5 Linearization Procedures

Linear systems can be directly cast into the convenient form of Eq. (3.30). But most systems are nonlinear, creating difficulties in analysis. Nonlinear methods of system analysis exist, but many times the dynamics of the system behave in an 'almost linear' fashion, so that linearization of the equations is a viable approximation [DeCarlo, 1989; Doeblin, 1980]. Since motions of interest concern small increments about a nominal (or equilibrium) position, then the most common method is to use a multi-variable Taylor Series expansion about the nominal point. As previously mentioned in section 2.4, expansion of the complete equations is possible, but it is usually more efficient to expand the nonlinear energy quantities [Marion and Thornton, 1988; Ginsberg, 1995]. This

[16] **m** also includes the moments of inertia.

method is illustrated with the linearization of the potential energy, but it can be applied to any or all of the energy quantities, if necessary.

For a $6n$ degree of freedom system at the equilibrium position, Lagrange's equations are satisfied by

$$q_k = q_{k0},\ \dot{q}_k = 0,\ \ddot{q}_k = 0 \qquad k = 1,2,...6n \tag{3.31}$$

Where q_{k0} are the nominal values of the generalized coordinates at equilibrium. Since all nonzero terms of the first part in (3.11) contain $\dot{q}_k$ explicitly, Lagrange's equations at equilibrium reduce to

$$\left.\frac{\partial L}{\partial q_k}\right|_0 = \left.\frac{\partial T}{\partial q_k}\right|_0 - \left.\frac{\partial U}{\partial q_k}\right|_0 = 0 \tag{3.32}$$

Where the subscript 0 denotes evaluation at the equilibrium position. Since the coordinate transformations do not explicitly contain the time, the kinetic energy is then a quadratic function of the generalized velocities [Marion and Thornton, 1988], and

$$\left.\frac{\partial T}{\partial q_k}\right|_0 = 0 \qquad k = 1,2,...6n \tag{3.33}$$

hence, from (3.32)

$$\left.\frac{\partial U}{\partial q_k}\right|_0 = 0 \qquad k = 1,2,...6n \tag{3.34}$$

Expanding the potential energy in a Taylor's series about the equilibrium position

$$U(\mathbf{q}^g) = U_0 + \sum_k \left.\frac{\partial U}{\partial q_k}\right|_0 \Delta q_k + \frac{1}{2}\sum_{j,k} \left.\frac{\partial^2 U}{\partial q_j \partial q_k}\right|_0 \Delta q_j \Delta q_k + ... \tag{3.35}$$

The first term is zero, since the potential is chosen to be zero at equilibrium, and the second term also vanishes in light of (3.34). Assuming all displacements are small, then terms higher than second order are ignored, and the potential energy becomes

$$U(\mathbf{q}^g) = \frac{1}{2}\sum_{j,k} A_{jk} \Delta q_j \Delta q_k \tag{3.36}$$

where

$$A_{jk} = \left.\frac{\partial^2 U}{\partial q_j \partial q_k}\right|_0 \tag{3.37}$$

This expansion produces second order terms in the potential energy, which result in linear contributions to the equations of motion after application of Eq. (3.29). This method can be used, if necessary, to expand all energies of the system by implementing Eq. (3.35) and retaining all nonzero terms up to second order.

Expansion of a particular energy quantity in Mathematica is accomplished with one line (potential energy is shown here), and appears identical to Eq. (3.35):

$$U_{lin} = U /. \{q_k \to q_o\} + \sum_{k=1}^{n} \partial_{q_k} U \Delta q_k /. \{q_k \to q_o\} + \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \partial_{q_j, q_k} U \Delta q_j \Delta q_k /. \{q_k \to q_o\};$$

3.6 Synthesis of CAMGV Components

A flowchart of the steps required for the entire model generation process is shown in Fig. 3.5.

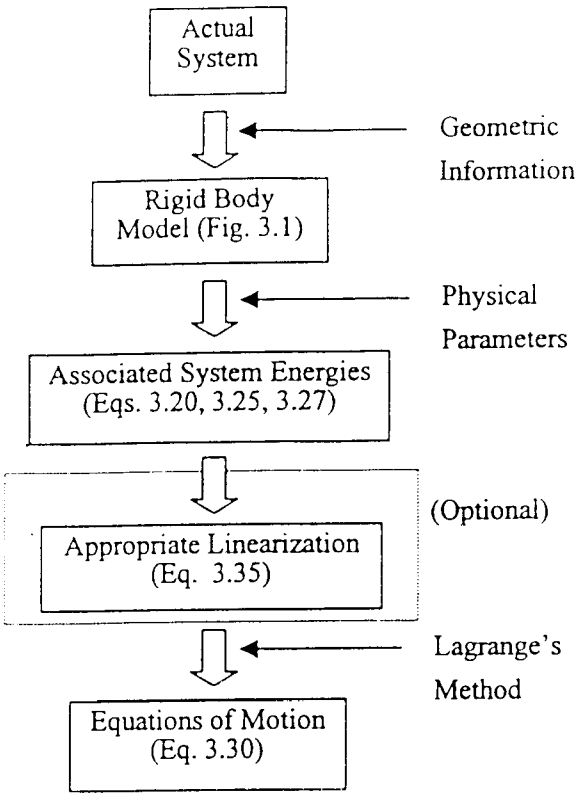


Fig. 3.5: Steps taken in the Computer-Aided Model Generation and Validation process.

Beginning with the actual system, the rigid bodies and couplings are defined, as well as the known and unknown physical parameters. A formal definition of the system is then constructed using known information about the system (e.g., geometries and locations of the rigid bodies and points where the couplings are attached). This is accomplished by utilizing the first principles to generate the energy equations as functions of the relevant state variables and parameters of the system. Depending on what analysis is required, the associated system energies are linearized accordingly using a Taylor Series expansion about an equilibrium position; however, this is not an absolute requirement. Application of Lagrange's method on the energy equations produces the general equations of motion. As with linearization of the energies, generation of the equations of motion is not a requirement, but is frequently done due to the high versatility of this particular model form.

3.7 State Space Formulation

It is often advantageous to bring the equations of motion into state-variable form which allows easy implementation of a wide variety of methods developed in controls theory. These methods include model tuning and identification, testing invertibility, observability, controllability, and design of controllers and observers.

The state-space form is obtained by transforming the (second order) equations of motion into a set of simultaneous first-order nonlinear differential equations

$$\text{State equation:} \qquad \dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \tag{3.38a}$$

$$\text{Output equation:} \qquad \mathbf{y}(t) = \mathbf{h}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \tag{3.38b}$$

Where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are the vectors of state variables and inputs, respectively. The common representation of a time varying lumped *linear* system is a set of four matrices which defines a first order, n degree vector differential equation [DeCarlo, 1989]

$$\text{State equation:} \qquad \dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{3.39a}$$

$$\text{Output equation:} \qquad \mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t) \tag{3.39b}$$

where

- $\mathbf{x}(t) \in \Re^n$ – state vector of n state variables,
- $\mathbf{u}(t) \in \Re^m$ – input vector of m inputs,
- $\mathbf{y}(t) \in \Re^r$ – output vector of r outputs,
- $\mathbf{A} \in \Re^{n\times n}$, $\mathbf{B} \in \Re^{n\times m}$, $\mathbf{C} \in \Re^{r\times n}$, $\mathbf{D} \in \Re^{r\times m}$, with all elements piecewise continuous[17].

The evolution and control matrices are functions of the **m**, **c**, and **k** matrices found in Eq. (3.30)

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{k}\mathbf{m}^{-1} & -\mathbf{c}\mathbf{m}^{-1} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{m}^{-1} \end{bmatrix} \tag{3.40}$$

The **C** matrix is a property of the system that relates the specified states to the outputs, and the **D** matrix is usually zero. The state space form has several important features worth noting:

---

[17]It is also common, particularly on the West Coast, to use the notation **F**, **G**, **H** and **J** in place of **A**, **B**, **C** and **D**.

- Essential properties of the analyzed structure are encapsulated in the evolution matrix **A**. The influence of system parameters on the dynamic behavior (such as modal frequencies) can be readily seen and evaluated.
- The impact of input signals, both control and disturbances, are encapsulated in the control matrix **B**. This allows the study of the system's controllability as well as its sensitivity to disturbances.
- Testing and correcting important dynamic properties[18] can be readily accomplished through the use of commercially available Computer Aided Control Systems Design (CACSD) packages. Utilizing the state-space form with these packages enables easy handling of equations which, if dealt with by traditional means, would require many pages of calculations.
- Generation of the input-output (transfer function and zero-pole) form commonly used in experimental analysis of dynamic systems is straightforward.

An illustration of the last point will be presented in the following section, with a brief discussion of the advantages gained when using these two forms of system representation.

3.8 Applications of the State-Variable Form

State-variable representations of multi-input, multi-output (MIMO) systems often involve descriptions of internal energy distribution. This provides a relationship between the system states, or internal variables, and the system inputs and outputs. Information provided by a state-variable form pertaining to the internal behavior of a system can be important, depending on the situation, but there are also many cases where a direct input-output relationship is more convenient. The transfer function (TF) model, also referred to as an input-output model, is well known and often used in controls theory. Cause-effect relationships in linear, time-invariant MIMO systems are conveniently described using this model [DeCarlo, 1989; Franklin, et. al., 1994]. The transfer function $G_{ij}(s)$ is a

[18]This includes observability, controllability, and invertibility.

rational polynomial in s and relates one particular output $Y_i(s)$ to one particular input $U_j(s)$

$$G_{ij}(s) = \frac{Y_i(s)}{U_j(s)} = \frac{b_p s^p + b_{p-1} s^{p-1} + \ldots + b_2 s^2 + b_1 s + b_0}{s^q + a_{q-1} s^{q-1} + \ldots + a_2 s^2 + a_1 s + a_0} \tag{3.41}$$

The physical parameters $\{\mathbf{m}, \mathbf{c}, \mathbf{k}\}$ are hidden in the model coefficients $\{a_0, a_1, \ldots, a_{q-1}, b_0, b_1, \ldots, b_p\}$ and are typically difficult to extract unless a deterministic relationship between the TF model and a constitutive model based on first principles exists. Nevertheless, this input-output model is still a useful tool for investigating the system's dynamic behavior due to changes in the physical parameters. Visual inspection of Bode plots (magnitude and phase) can aid in determining resonance frequencies and damping ratios of the system.

The zero-pole (ZP) model is a slight variation on the TF model in that the polynomials are represented in factored form

$$G_{ij}(s) = g \frac{(s - z_1)(s - z_2) \cdots (s - z_p)}{(s - p_1)(s - p_2) \cdots (s - p_q)} \tag{3.42}$$

where

g	--	gain of the system,
$z_1, \ldots, z_p$	--	real or complex valued zeros,
$p_1, \ldots, p_q$	--	real or complex valued poles.

The zeros and poles can be plotted on the s plane, and their locations provide immediate visual information concerning stability, resonance frequencies, and damping ratios [Franklin et. al., 1994].

Derivation of either the TF or ZP models is a straightforward matter once the state variable form is available. The transformation involves manipulation of the state space matrices

$$\mathbf{G}(s) = \frac{\mathbf{Y}(s)}{\mathbf{U}(s)} = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{3.43}$$

The result is an n x m matrix of transfer functions that correspond to n outputs and m inputs. Symbolic transfer functions for MIMO systems are readily computed from the state space matrices using a one line command in the *Control System Professional* package from *Mathematica:*

```
TF = TransferFunction[s, {A, B, C, D}];
```

3.9 Intuitive Validation and Troubleshooting

A challenging aspect of any modeling effort is verifying how accurate the model represents the actual system. Granted, the very definition of a model is an *approximation* to a real system, and there are many factors that contribute to the accuracy of representation. Several of the more important ones include:

- system *perception* (the perspective taken on viewing the system),
- 'granularity' of component description (detailed vs. broad),
- choice of lumped or distributed parameters, and which parameters to include,
- type of formalism used to generate equations,
- applicable linearization procedures, and
- capabilities/limitations of the derived model form(s).

The choice of which behavior to emulate (e.g., input-output response, mode shapes, resonance frequencies, etc...) also has a significant impact on the effectiveness of the model. A method of efficiently evaluating the combined effect of all these factors is essential.

The most established and convincing test is comparison between simulated and experimental results. However, there are also methods of testing that require no actual experimental results. Presented here are simple, efficient, and intuitive means of validating the model (on a preliminary basis) during and after the derivation process. This potentially allows some errors and/or poor representations to be detected before extensive analysis of experimental data is performed, minimizing the effort of model 'tuning.'

When confronted with a complex structure to analyze and a proposed modeling strategy with unproven accuracy or 'robustness', it is necessary to begin at a level that is easily interpretable and well established. One of the best ways of accomplishing this is to consider the simplest configuration in the class of structures under consideration. By applying the analysis strategy to the simplified system and comparing it to results of reliable (e.g. 'manual') methods and known quantities[19], it is possible to detect errors and increase the confidence level of the model. Once the simplified system is well established, the next step is to increase the complexity and repeat the analysis. However, this can only be iterated until the limits of human patience are reached. At that point it is

[19] Due to physical or kinematical constraints, and properties of the system.

advantageous to employ analysis techniques provided in experimental modal analysis [Ewins, 1984] and controls theory. Rather than observing pieces of the structure, controls tests analyze the system as a whole, providing easily verifiable information such as stability of the structure[20]. The following three subsections illustrate these intuitive techniques by way of example using closed loop mechanisms that represent parallel structure machine tools.

3.9.1 Incremental Complexity Approach

An inherent feature of parallel structure machine tools is that they are geometric closed loop mechanisms. This means there are multiple paths to take through the components to travel from one point to another on the structure, which makes dynamic analysis challenging. In addition to this, each rigid body has full 6 degree of freedom movement, and is controlled simultaneously by all its strut, or SDE, couplings. Based on this basic description, the simplest structure that retains these characteristics is a single degree of freedom rigid body, controlled by a single strut. Figure 3.6 is an example of such a structure. The rigid 'platform' has one rotational degree of freedom (DOF) about the hinged point, and there are two geometric paths that connect the end of the platform (the primary point of interest, point **P**) to the inertial support. The orientation of the platform is controlled by moving the SDE connection at the inertial support (point **B**) along the *x*-axis.

---

[20] Most mechanical structures are inherently stable, making this an excellent and immediate indicator of a model inconsistency.

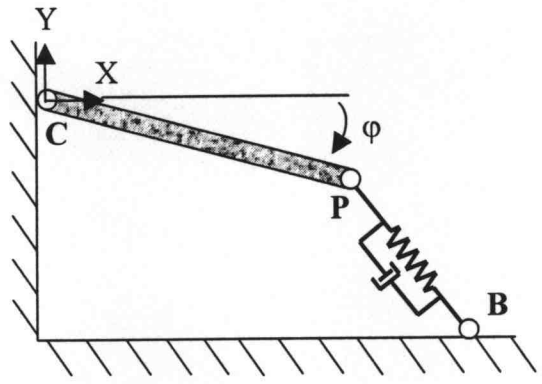


Fig. 3.6: Single DOF, single strut parallel structure.

Straightforward analysis of the system allows several methods to be readily implemented and compared. Although this exercise appears trivial, it is necessary for establishing the beginnings of a proper analysis technique suitable for parallel structures.

The next step is to increment the number of struts by one and re-evaluate the situation. Replacing the hinged connection (point **C**) of Fig. 4.1 with another strut gives the platform two additional degrees of freedom (two translational and one rotation), but this also creates a problem. Initial inspection reveals[21] that the system is insufficiently constrained, so a third strut must be added. Similar to the single DOF case, the orientation of the platform is accomplished by moving the connections of the SDEs to the rigid support along the *x*-axis.

Traditional analysis of this new system is considerably more tedious than the first, but it provides important information about certain aspects of the modeling method that were not addressed in the single DOF system. Namely, that it allows testing of a simple coordinate transformation for only one rigid body. One coordinate transformation on a single rigid body (as opposed to multiple transformations on multiple bodies) is an advantageous situation since it delineates some of the more involved processes of the method in a simplistic, uncluttered manner.

---

[21] Testing for stability will also immediately reveal this fact.

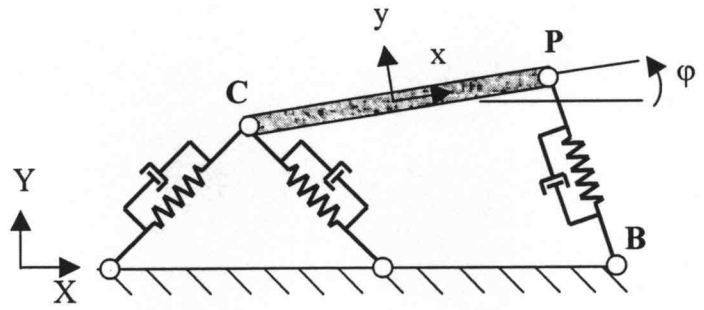


Fig. 3.7: 3 DOF, 3 strut parallel structure.

The obvious next step would be to extend this case to three dimensions, using a full 6 DOF rigid body. However, the usefulness of this intuitive validation approach has been virtually exhausted after the second case, since any more additions to the structure would require extensive analysis that is prone to multiple human error and bountiful frustration. For incremented cases that can not be readily analyzed using the traditional approach, it is worthwhile to switch to other techniques of validation.

3.9.2 Comparison to Known Quantities or Conditions

Other intuitive techniques exist whose effectiveness does not significantly diminish for increasing complexity. Many of these involve special orientations and/or inherent properties of the model components.

Case 1: The first is testing the model's output for a simplified *configuration*, not necessarily a simplified *system*[22]. This requires orienting the structure in a manner that can be easily described by direct methods, such as Newton's second law. Consider the first two cases of the previous section. One of the simplest configurations that both of these can attain are depicted in Fig. 3.8. The equations of motion for small displacements about this position are trivial for the first case, and are straightforward for the second

[22] A simplified *system* is one of limited DOF or components; a simplified *configuration* can be a complicated structure, but oriented in such a manner that only a select few DOF are active.

case. Using these equations as a benchmark provides immediate information about the model's ability (or inability) to predict these configurations correctly.

The simplified configurations can also be used to compare other quantities, such as potential energy of the SDEs, resonance frequencies, and static stiffness. For small displacements the expressions for potential energy are compact and clear, and can even be verified by visual inspection.

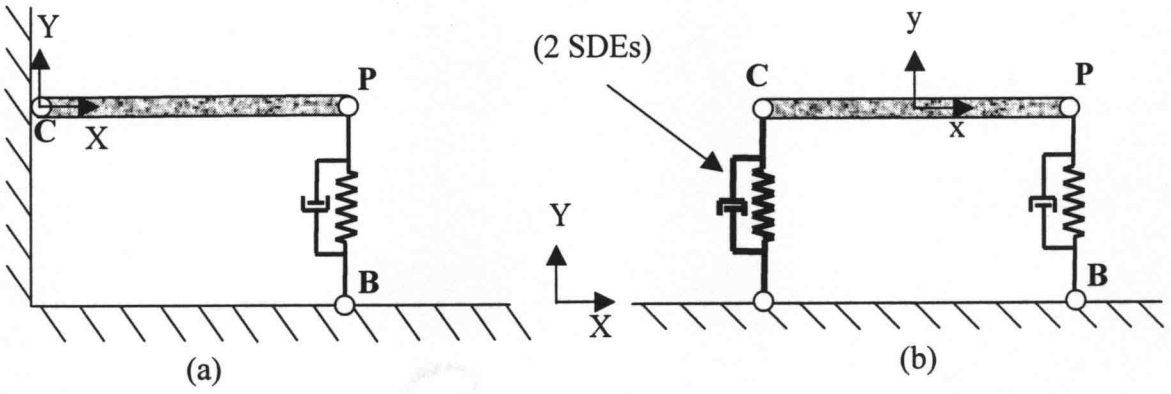


Fig. 3.8: Simplified orientations of (a) 1 DOF and (b) 3 DOF systems.

Resonance frequencies are found from Bode plots of the appropriate TF model, and can also be roughly estimated[23] by visual inspection of the configurations in Fig. 3.8. The 1 DOF case shown in Fig. 3.8(a) is trivial, since only one resonance frequency is possible. Fig. 3.8(b) is more illustrative since it has 3 DOF, and hence will have three resonance frequencies. Exciting the system (applying an input) in the *y* direction will produce a dominant translational resonant frequency in that direction. Since it is also an asymmetrical configuration (two springs on the left and one on the right), there will be a slight rotational resonant frequency. An examination of the Bode plot for this structure should verify this fact, and has actually been observed in this research.

---

[23] 'Roughly estimated' refers to the approximate number of resonance frequencies that *should* appear and their relative magnitudes.

The last quantity that can be analyzed with respect to simplified orientations is static stiffness. Calculation of torsional or linear stiffness of the structures in Fig. 3.8 are straightforward, and are easily found from the equations of motion by setting all the time derivatives of generalized coordinates equal to zero. Comparison of the two provides yet another indicator for validation purposes.

Case 2: One other important check concerning the potential energy of the SDEs can be performed, and is entirely independent of configuration or complexity. It deals with a property inherent of the SDE itself, and is invariant for different configurations or structures. Since the energy stored in the 'spring' is always modeled as a quadratic function centered about equilibrium, then certain terms will always be zero in the Taylor Series expansion about the equilibrium point. This is illustrated by considering a quadratic potential centered about zero

$$U(x) = \frac{1}{2}kx^2 \tag{3.44}$$

Writing the Taylor Series expansion for Eq. (3.44) about zero

$$U(x) = \frac{1}{2}kx^2\Big|_0 + \frac{\partial}{\partial x}\left(\frac{1}{2}kx^2\right)\Big|_0 \Delta x + \frac{1}{2!}\frac{\partial^2}{\partial x^2}\left(\frac{1}{2}kx^2\right)\Big|_0 \Delta x^2 + \ldots \tag{3.45}$$

Evaluating the derivatives results in

$$\begin{aligned} U &= \frac{1}{2}kx^2\Big|_0 + kx\big|_0 \Delta x + \frac{1}{2}k\big|_0 \Delta x^2 + \ldots \\ U &= \frac{1}{2}k(0)^2 + k(0)\Delta x + \frac{1}{2}k\Delta x^2 + \ldots \end{aligned} \tag{3.46}$$

It is clear that the first and second terms are zero, and the only nonzero contribution (up to second order) is the third term. This holds for all cases; x in Eq. (3.44) can be a function of many variables, but when the potential function is expanded about the equilibrium point, the first two terms in the expansion will *always* be zero, because x is *defined* to be zero at that position. This inherent consequence is a powerful tool for validating the correct definition of potential energy. It is also a viable means of checking programming or conceptual errors in the preliminary stages of model development.

Case 3: Comparison of a different modeling method provides an additional tool for validation and troubleshooting. The model considered in this research was developed by the IWF at ETHZ [Weikert, et. al., 1998]. Both methods are constructed in virtually the same manner, but have distinct characteristics that make each unique. The IWF

(referred to as 'direct') approach provides excellent insight into the physical phenomena in machines, but is for reasons of practicality limited to Cartesian machines or those of limited complexity. The direct approach is capable of modeling more complex structures with sufficient accuracy, but as the complexity increases, the primary advantage of this method diminishes; namely, that user input becomes tedious and the results obtained are not easily interpreted. The approach presented in this research (referred to as 'generalized') is more comprehensive, and can accommodate complex configurations with minimum user input. On the other hand it is not suitable for quick, intuitive interpretation of results except for simple configurations.

Both methods use Lagrange's energy formalism as a basis for model derivation and under correct assumptions will produce identical model structures. The major difference between the two methods lies in the definition of the **m**, **c**, and **k** matrices in Eq. (3.30). The generalized approach uses suitable homogeneous and rotational transformation matrices to calculate the matrix coefficients, whereas the direct approach assumes a general form of the **m**, **c**, and **k** matrices and explicitly defines each element accordingly. The advantage of the direct approach is immediately apparent: a direct and intuitive mapping exists between the parameters of the model and those of the actual system. Although the generalized approach easily handles configurations that would be cumbersome with the direct approach, the ease of interpretability by the direct approach on simple (in particular Cartesian) machines is critical for comparison purposes when verifying the validity of the generalized approach. Inconsistencies between the two approaches are easily identifiable, and can be traced back in a straightforward manner due to the intuitive nature of the direct approach.

3.9.3 Controls Tests

Another set of 'preliminary' checks that can be performed is found in controls theory. Several controls related aspects of the system can be analyzed using state space and transfer function forms of the model.

Testing for static compliance using the Final Value Theorem can be done using the transfer function form [Franklin and Powell, 1994; DeCarlo, 1989]. This essentially checks the compliance, or 'inverse' stiffness, of the structure in a particular direction, usually along a generalized coordinate. The result itself, whether it be symbolic or

numeric, is very important for design optimization as well as comparison with other orientations of the structure.

For example, the stiffness in the y direction of the orientation shown in Fig. 4.4(a) will be significantly less than the stiffness in the same direction for the orientation shown in Fig. 4.4(b). Comparing the static compliance of the model between both cases provides another indicator for validation purposes. Additionally, testing stiffness in the x direction checks the model's ability to predict configuration singularities. The compliance value for Fig. 3.9(a) is finite, but is infinite (zero stiffness) for 3.9(b).

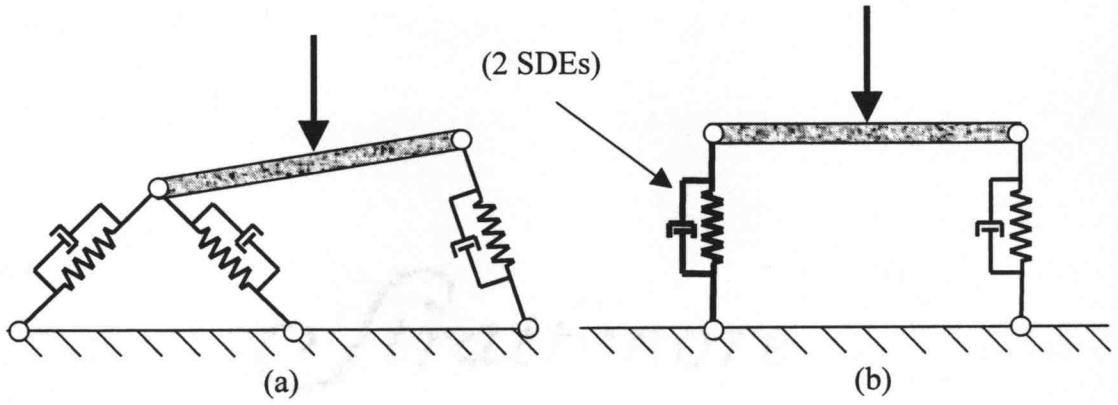


Fig. 3.9: Testing static compliance for different orientations of the structure.

Testing for system stability is usually accomplished through use of the state space form. Calculation of the eigenvalues, λ, for the evolution matrix is a straightforward process

$$(\mathbf{A} - \lambda \mathbf{I}) = 0 \tag{3.47}$$

The eigenvalues are generally complex, so for a system to be stable all associated λ must satisfy the condition [DeCarlo, 1989]

$$\mathrm{Re}\{\lambda_i\} \leq 0 \tag{3.48}$$

Otherwise, the system is considered unstable. The use of computer-aided control system design (CACSD) software packages such as *Control System Professional* in *Mathematica* facilitate controls analysis. Checking the eigenvalues of the evolution matrix is accomplished with a single line command in *Mathematica.*

The other method of defining stability is done with the transfer function form. Plotting the poles (roots of the denominator, or characteristic equation) on the *s* plane provides an immediate visual confirmation of the system's stability. For a completely stable system, all poles must lie in the left half plane (LHP) [Franklin and Powell; 1994]. If any pole is to the right of the imaginary axis (RHP), then the system is unstable. Since it is generally known if a mechanical structure is stable, this is yet another definitive check of the model, and can be performed using state space or transfer function forms.

3.10 Closure

The presented method of model derivation contains several characteristics that enhance versatility. For systems where lumped parameter rigid body approximations are appropriate, the generic multi-degree-of-freedom system offers a lot of freedom in terms of body orientation and couplings. The use of Lagrange's method provides a clear path for transformation into the mathematical domain that can take on several forms, examples of which include state space and input-output transfer functions. The use of *Mathematica* with the *Control System Professional* package provides a unique modeling environment that can accommodate symbolic and numeric computations, and allows numerous analysis techniques to be efficiently implemented in several different domains (e.g., frequency, continuous time, discrete time) and model forms.

Several techniques for validation of generated models were presented. Although no single method provides an absolute indicator of the accuracy of the model, with careful and stepwise application the combined effort can improve the level of confidence with the process. The intuitive procedures can be implemented during the initial development stages, offering an additional systematic approach of pre-checks.

4. EXPERIMENTAL VALIDATION

The final, and often most convincing, way of validating a model is comparison with experimental results. Since the model can take on different forms and work in several domains, there are many options for experimentation. Probably the quickest and most common approaches are carried out in both the frequency and time domains. The first section outlines the most common procedures used with respect to the two domains. The second section presents an example of frequency domain analysis that was performed on a Stewart platform [Fichter, 1986].

4.1 System Dynamics

Models of machine tools obtained according to the delineated algorithm are complicated and not suitable for direct application by machine tool designers and users. Instead, they facilitate the generation of certain features that have appealing and intuitive interpretations. Four examples of these features are: 1) time domain responses, 2) frequency domain responses, 3) modal frequencies, and 4) the corresponding mode shapes.

Time domain responses are useful for investigating the system's behavior under excitations representative of those in working conditions. The standard procedure is to measure responses resulting from an impulse, step, and ramp input. Information obtained from these tests, such as overshoot, settling time and steady state error, are advantageous for determining key dynamic characteristics of the system and in many cases help identify shortcomings of controllers. Plotting both actual and simulated responses together also provides a quick visual means of comparison. A large portion of this information can be obtained with previously mentioned validation techniques, but a correctly simulated response is the benchmark of an accurate model since it combines all validation parameters into one picture.

Frequency domain responses are excellent for identifying the resonance frequencies and dynamic stiffnesses of the system. Investigation into this domain is even simpler than in the time domain since the Frequency Response Function (FRF) is obtained directly

from the transfer function (TF) form of the model. All the necessary information encapsulated in the FRF is typically displayed in a Bode diagram, where the resonance frequencies and their relative magnitudes are readily seen. Of particular interest in machine tools is the Tool Center Point (TCP) dynamic behavior[24], since the relative motion between tool and workpiece determines the accuracy of manufactured parts. Fig. 4.1 shows a typical Bode diagram demonstrating the dynamic stiffness calculated at TCP [Weikert, et. al., 1998].

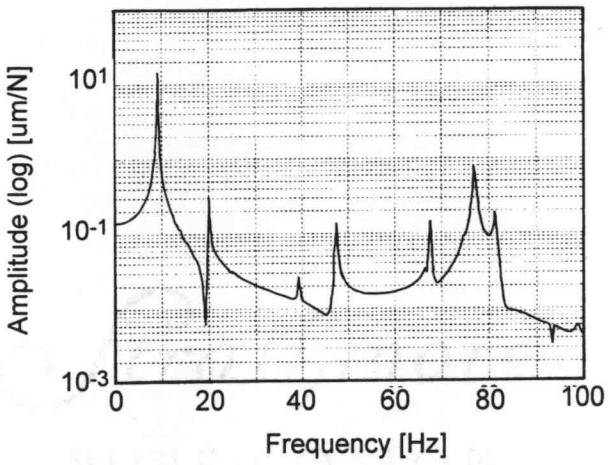


Fig. 4.1: Dynamic stiffness calculated at TCP [Weikert, et. al., 1998].

Modal frequencies are readily obtained from eigenvalues of the evolution (**A**) matrix (see Eq. (3.40)), provided that numerical values of its elements are appropriately identified. The number of these frequencies in the proposed model is equal to the number of degrees of freedom (typically 6 per each rigid body). Modal frequencies are helpful indicators in the conceptual phase of machine tool design, since their high values indicate, as a rule, desirable dynamic properties of the machine. At present it is not possible to obtain closed-form analytical expressions relating the modal frequencies to the design variables. Such expressions would be very helpful to optimize the designs.

---

[24] Dynamics observed at the Tool Center Point (TCP).

Mode shapes visualize distinctive patterns of displacement between the components of the structure represented as rigid bodies in the model. In experimental Modal Analysis these shapes are often animated at modal (natural) frequencies. An example mode shape is shown in Fig. 4.2.

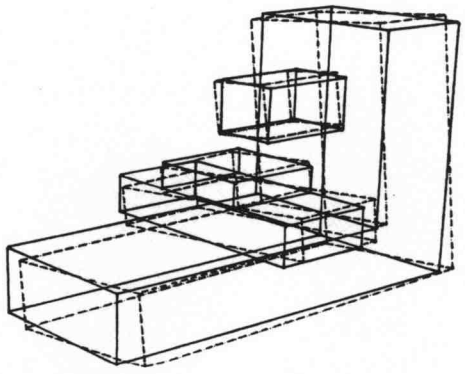


Fig. 4.2: Animated mode shape [Weikert, et. al., 1998].

In the proposed method, animated vibrations of modeled machine tools are readily obtained for arbitrary excitations. By converting the state space model (Eqs. (3.39a), (3.39b)) into the equivalent matrix transfer function using Eq. (3.43) and setting $\mathbf{C} = \mathbf{I}$ one obtains the Laplace transform of the system's *global* generalized coordinates $\mathbf{Q}^g(s)$ (see Eqs. (3.28), (3.29)) in response to the input signals **U**(s). Together with homogeneous matrix transformations they facilitate rapid generation of animated pictures as shown in Fig. 4.2. It should be noted that the computations needed for updating mode shape visualization after a design modification are quick, so the designer can readily have feedback as to the effects of his/her decisions.

4.2 Stewart Platform Analysis

As mentioned previously, parallel structure machine tools are particularly suited for description by lumped parameter methods due to the inherent properties of their design,

and hence prove advantageous for experimental analysis of the generated model. One such structure available at Oregon State University is a Stewart platform, the result of past research on parallel manipulators [Fichter, 1986].

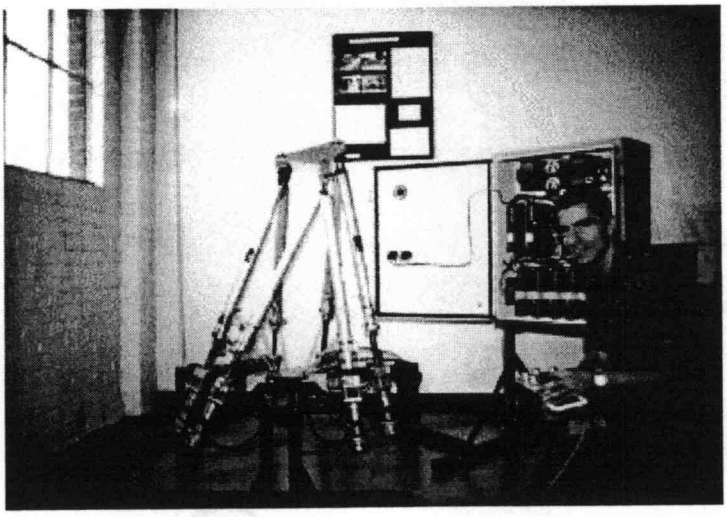


Fig. 4.3: Stewart platform at Oregon State University.

The structure consists of a platform supported by six legs, or struts, that are fixed to a rigid base. Full 6 DOF movement of the platform is accomplished through lengthening or shortening of the legs. One of the simplest decompositions of the Stewart platform is into 1 rigid body (the platform) and 6 SDEs (the legs). In general, each SDE has a stiffness, k_i, and a damping coefficient, b_i, both of which are intended to represent the combined stiffness and damping of the joints (at either end) as well as the leg itself. It is well known that the stiffness of joints is not a constant, but varies with orientation; however, it has been shown that the fluctuations are sufficiently small to be negligible when compared to the overall stiffness of the leg/joint coupling [Weikert, et. al., 1998]. The base is assumed to be completely rigid, and the legs are assumed massless. This last assumption is obviously not correct, but is an acceptable approximation since the leg masses are only a fraction of the platform and servo motor masses.

Experimental analysis of modal frequencies was chosen since the necessary system response can be acquired through use of readily available accelerometers such as

those developed by Kistler [Kistler, 1995]. Several ways of inducing a system response through controlled and recorded inputs include tap tests, sinusoidal excitation, and 'white noise' excitation [Ewins, 1984; Van Brussel, et. al., 1975]. Although the latter two are more comprehensive and complete, the tap test was chosen since it is quick, easy, and provides sufficiently detailed information.

4.2.1 Data Acquisition System

A standard data acquisition (DAQ) system is comprised of the following basic components: 1) a controller, 2) a signal conditioner, 3) a multiplexer and amplifier, 4) an analog-to-digital converter (ADC), 5) a storage or memory unit, and 6) a readout device [Dally, et. al., 1993].

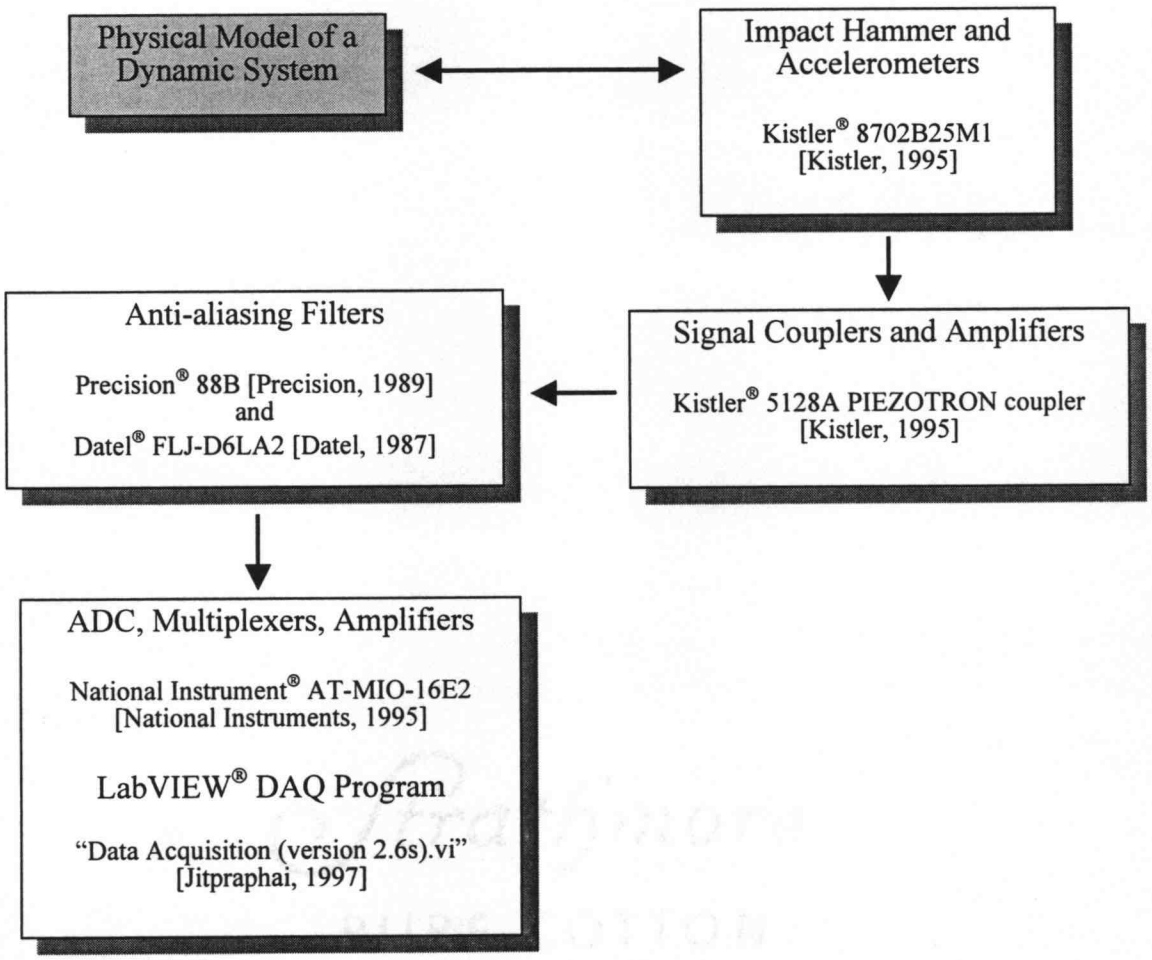


Fig. 4.4: The basic data acquisition system used in this research [Jitpraphai, 1997].

The DAQ system employed in this research is a computer-based instrument where the control is implemented in a LabVIEW® program and the memory and readout are integrated into a desktop computer. The ADC, multiplexer, and amplifier are provided by a plug-in DAQ printed circuit board (type AT-MIO16E2) from National Instruments [National Instruments, 1995]. Low pass filters serve as signal conditioners to prevent signal aliasing. A schematic diagram of the employed DAQ system is shown in Fig. 4.4.

Data acquisition and signal conditioning are both broad fields that have been extensively documented in literature. A recent review of these fields in relation to visualization is provided by Jitpraphai, as well as a detailed description of the software and hardware used in the experimental portion of this research [Jitpraphai, 1997].

The sensor used was a three-axis accelerometer by Kistler and was mounted near the center of mass of the platform using beeswax [Doebelin, 1990]. Such a configuration is only able to record translational accelerations. However, previous research on a similar parallel structure demonstrated that the lowest resonance was translational [Weikert, et. al., 1998], and hence indicates that this method is sufficient for identification of the most dominant resonant frequencies. Signals from the accelerometer were conditioned by an *Accelerometer Coupler* (Kistler® model 5128A) and passed through low-pass *Anti-aliasing Filters* (Precision® model 88-B and Datel® model FLJ-D6LA2 programmable filters). The cutoff frequency was set to 1 kHz and the filters provided signal gains of 10. Filtered and amplified signals were passed through an *Interface Panel* to the *DAQ Card* (National Instrument® AT-MIO 16E2) installed inside a desktop computer. The DAQ card used a 12-bit Analog-to-Digital Converter (ADC), a multiplexer, and additional amplifiers. A program written in LabVIEW® set the gains of these amplifiers [Jitpraphai, 1997].

4.2.2 Experimental vs. Analytical Results

The Stewart platform was stimulated by striking it in an arbitrary direction to excite all possible vibration modes. Data collected from the DAQ system consisted of 2048 points at a sampling rate of 300 Hz. The process was repeated three times using different impact directions for each trial. The nine data sets (three for each axis) were then analyzed using an Auto Regressive Modal Analysis (ARMA) software package to extract the resonant frequencies [Ewins, 1984; Leuridan, 1981; Pandit, 1991].

In order to generate resonant frequencies from the model, critical parameters of the system must first be estimated. These include the mass and inertia values of the platform, and the stiffness of the legs. Estimative calculations of the platform parameters provided the following results:

Parameter	**Symbol**	**Value**	**Units**
Mass of the platform	m	10.84	Kg
Moment of Inertia about the x axis	J_{xx}	0.024	$Kg\ m^2$
Moment of Inertia about the y axis	J_{yy}	0.035	$Kg\ m^2$
Moment of Inertia about the z axis	J_{zz}	0.043	$Kg\ m^2$

Table 4.1: Estimated physical parameters of the Stewart Platform.

The upper joints were lumped in with the platform (assuming one single rigid body) for the above calculations. Calculation of the leg stiffness proved significantly more challenging, but an experimental technique was employed to provide an estimate to within an order of magnitude of the actual value. This was accomplished by testing a complete leg/joint coupling isolated from the structure. A load cell and accelerometer were placed at the end of the joint, and the stiffness was deduced from the relationship between the load placed on the leg and the deflection provided by double integration of the accelerometer signal [Jitpraphai, 1997]. The final result gave an average of 0.5 N/μm, with a standard deviation of 0.2 N/μm.

With estimated physical parameters it is a simple matter to compute the resonant frequencies from the evolution matrix of the model. Comparisons of the two results are shown in Table 4.2.

Experimental Frequencies (Hz)	**Analytical Frequencies (Hz)**
161	162
136	123
109	
75	79
63	
27	44
	17
8	7

Table 4.2: Experimental and analytical resonance frequencies.

Gaps in the table are indicative of inherent noise in the data acquisition system and the fact that some modes can not be detected by the three-axis accelerometer (e.g., rotational resonance frequencies). Experimental and analytical Bode plots were also generated for comparison of the relative magnitudes.

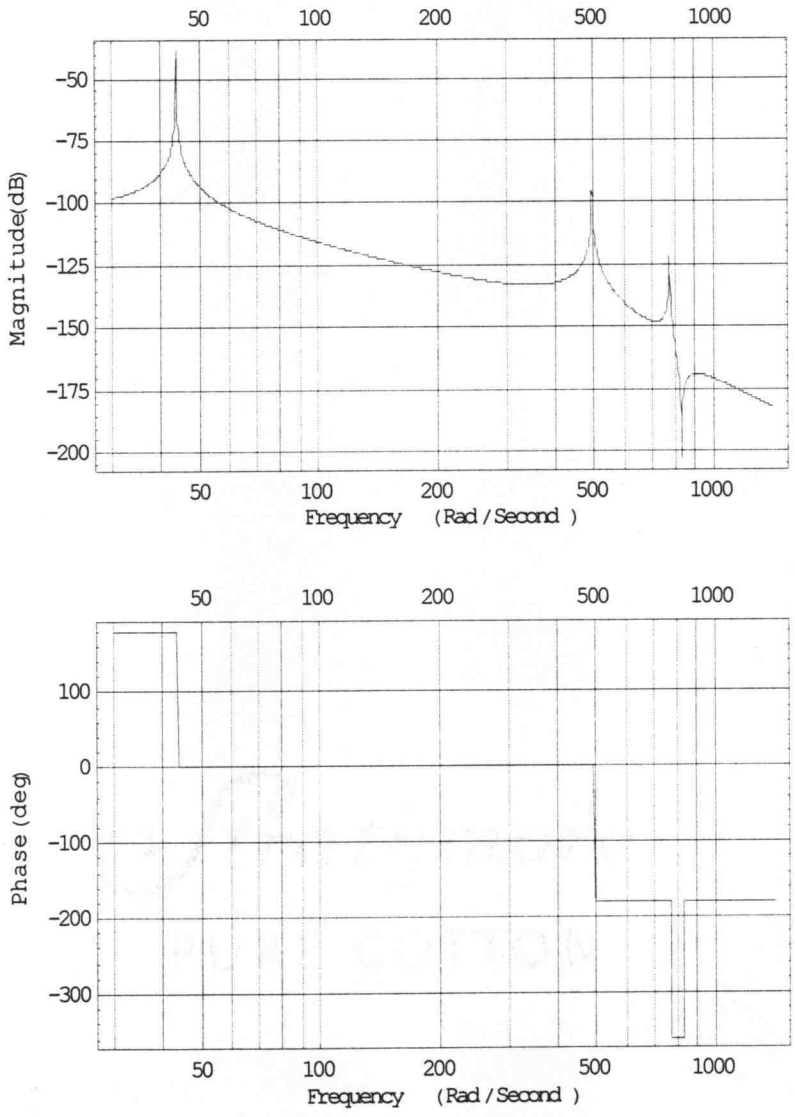


Fig. 4.5: Analytical Bode plot of Stewart platform from *Mathematica*.

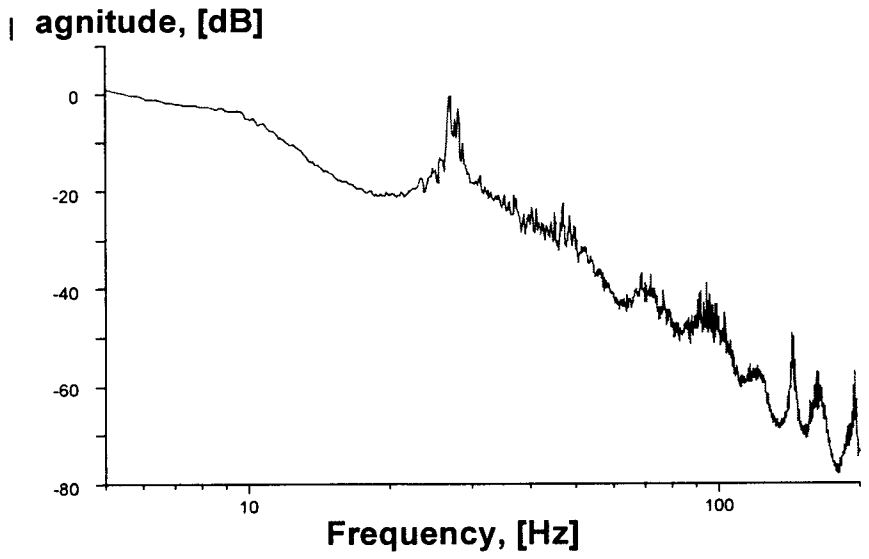


Fig. 4.6: Experimental Magnitude plot of Stewart platform.

Comparison of the magnitudes shows that the resonant frequency at 7-8 Hz is by far the most predominant for the analytical case, but is only slightly distinguishable in the experimental data. This is because the acquisition system and ARMA software used were best suited for higher frequencies and gave poor resolution at the low frequencies required for this structure. If the test were performed again with equipment that is more sensitive to lower frequencies, then the resonance frequency at 8 Hz will be revealed. This assumption is justified by the fact that the resonance was present in all tests performed; it was just not sufficiently detailed to draw definitive conclusions.

The fact that experimental and analytical resonant frequencies do not appear to have good correlation stems from several reasons, the first of which being inaccurate estimates of the physical parameters. However, the primary reason for the discrepancy lies in an invalid initial assumption made about the structure. In order to implement the rigid body model, the base supporting the legs was assumed perfectly rigid, when, in fact, it had significant flexibility. It was known that this would limit the accuracy of the model, but it was assumed that one or two of the most dominant modes would still be reasonably predicted. The obtained results reinforced the assertion that rigid body modeling is not appropriate for this particular Stewart platform. Despite this, the model was still able to identify the most dominant frequency at 7-8 Hz with fairly good accuracy. The model

was also instrumental in aiding experimental setup by providing preliminary, simulated responses.

4.3 Closure

Four commonly used options, available in the presented modeling environment, and concerning standard experimental verification in time and frequency domains were outlined. These options are an indicator of the model's ability to extensively tested using traditional experimental tests including simulated responses and analysis in several domains. The model is also instrumental in choosing these tests. This was further illustrated by a specific example concerning comparisons between experimentally and analytically obtained resonance frequencies of a Stewart platform. This is also a good indicator of how the generated model can accommodate final tuning and troubleshooting of machine tool structures.

The following chapter outlines some of the applications that benefit from the different forms available. It also illustrates some of the advantages afforded the analyst by being able to transition back and forth between model forms and domains, all in the same programming environment.

5. MODELING APPLICATIONS

A dynamic model that accurately represents a physical system serves several purposes. The following chapter highlights the main areas of research conducted at Oregon State University that significantly benefit from the presented modeling environment. It is by no means a complete listing of all possible applications. The following applications can be considered the primary motivation and justification for the development of computer aided modeling in these research fields. Since the generated model is used in a different manner for each application, this chapter is also a good illustration of the versatility available in the presented modeling environment.

5.1 Minimization of Dynamic Tool Path Errors

Minimization of tool path errors in high speed machining necessitates optimal design of machine tools and application of advanced control strategies. Analytical models that accurately represent interactions between the machine tool, workpiece and cutter prove necessary to accomplish these tasks. The impact of machine tool dynamics on product quality and process efficiency has been well recognized. The effects of these dynamics at higher frequencies (chatter and flexible mode vibrations) are well documented. On the other hand, there has been relatively little done with regard to the phenomena occurring at middle to low frequencies. The importance of this phenomena has recently increased, mainly due to extensive research of high speed machining (HSM) and the large inertial forces associated with it [Tu, 1991; Heisel, 1996; Weikert, et. al., 1997; Weikert, et. al., 1998].

An example can be the deviations of the *Tool Center Point* (TCP) from a straight line between points 'Start' and 'Stop' in Fig. 5.1. It shows three tool paths recorded with the Heidenhain Cross Grid [Heidenhain Corp.] on a large machining center during an execution of straight movement along the x-axis.

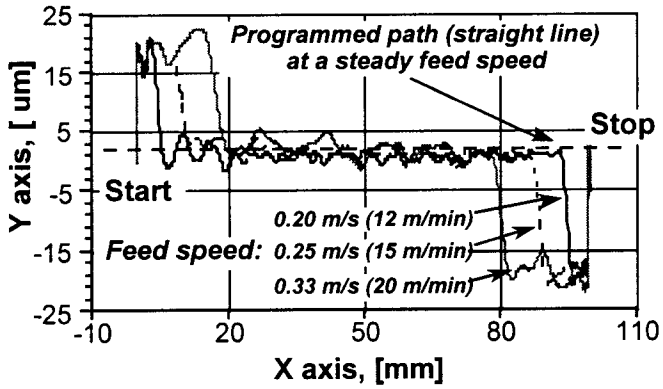


Fig. 5.1: Lateral TCP errors caused by acceleration and deceleration in straight motion [Weikert, et. al., 1998].

Displacements in the y direction ($\pm$ 20 μm in magnitude) proportional to the acceleration in the feed direction can be clearly seen at the beginning and end of the tool motion. They are attributed to a large bending moment due to the distance between the feed force direction and the center of gravity of a heavy machine's column being moved. Deviations like these are referred to as Dynamic Cross Talk (DCT), and are predominantly found in traditional, or serial axis, machine tools.

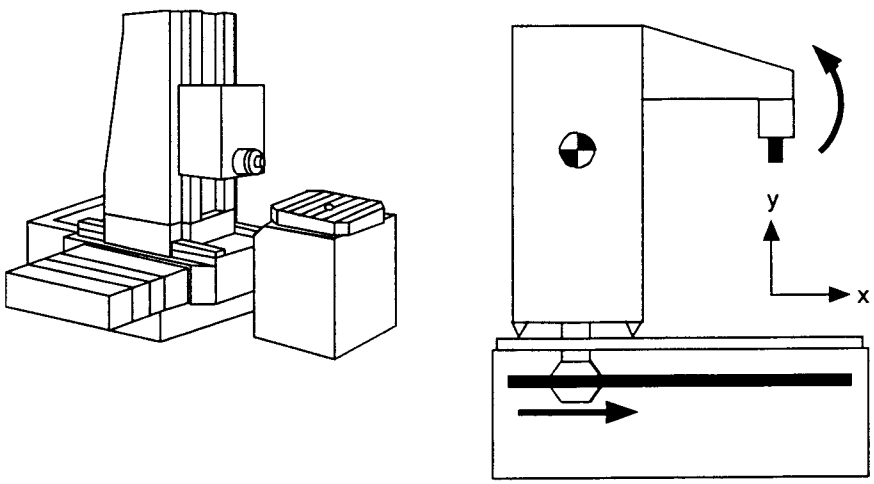


Fig. 5.2: Source of TCP deviations [Weikert, et. al., 1998].

The requirements of HSM necessitate the appropriate modeling of dynamic deviations due to inertial loads such as discussed above, which can be significantly larger than the cutting forces. The availability of accurate machine tool models is essential to minimize the effects of dynamic phenomena on the TCP path errors. These models are used in three different ways:

- First, they provide preliminary dynamic performance during the conceptual design phase of a machine tool structure, identifying 'weak' areas of the design before the before the structure is actually built.
- Second, they facilitate pre-compensation of NC programs to counteract anticipated deviations of the TCP from the nominal trajectory.
- Finally, the models provide a basis for implementing advanced position controllers for fine corrections of the actual tool path trajectory, which is continually affected by unmeasurable disturbances.

5.2 Application to Parallel Structure Machine Tools

Traditional (serial structure) machine tools are inherently difficult to model. This is due to the significant impact of bending moments[25] present in their structural components and resulting strong 'flexible mode' vibrations. Since an accurate description of these dynamic phenomena requires *distributed parameter* treatment and the use of *partial differential equations*, the tasks of modeling and control are very difficult [Bianchi et. al., 1996; Gysin, H.; Mottershead and Friswell, 1993].

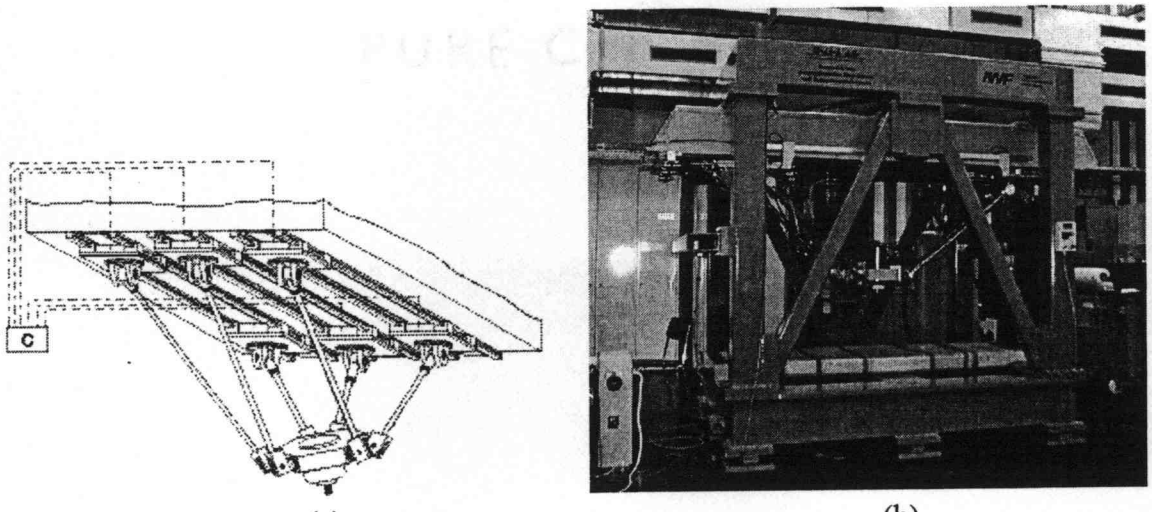
The situation is different in parallel structure machines. Their design virtually eliminates bending moments, so *lumped parameter* modeling techniques are applicable and potential benefits from applying advanced control, monitoring, and diagnosis are realistic. The emergence of this situation poses new challenges to the designers of machine tool subassemblies, in particular spindles, in order to seize the opportunities provided by radically changed main structures. It also poses exciting challenges to the controls community to fully realize the potential of new machines by implementing advanced control algorithms available for the lumped parameter dynamic systems.

---

[25] Including those due to inertial forces.

Fig. 5.3 shows Hexaglide, a representation of parallel structure machine tools that is analyzed in this research. It was designed and built by the Institute for Machine Tools and Manufacturing (IWF) at the Swiss Federal Institute of Technology (ETHZ) in Zurich, Switzerland [Hebsacker]. The spatial 6-axis positioning of the platform and TCP relative to the base (workpiece) is accomplished by moving six linear motors along linear guideways mounted on the base. The main advantage of this configuration in comparison to serial structure designs is effective elimination of bending moments. Since the struts operate mainly under axial tensile and compressive loads, a high stiffness to weight ratio is achieved. In addition, *six* motors push *one mass* as opposed to *one* motor pushing *one or more masses*. This leads to the reduction of weight and superior dynamic performance.

In addition, a high modularity and similarity of components simplifies the design as well as the modeling and control. On the other hand, an obvious disadvantage is the kinematic cross talk (KCT), or coupling between the axes. For example, moving the TCP along a straight line requires the coordinated action of all six linear motors. Similarly, the impact of cutting forces (process disturbances) is complex. Even if a force acts exactly along one axis, in most machine configurations it causes displacements of TCP in all Cartesian directions.



(a) (b)

Fig. 5.3: Hexaglide - schematic diagram (a), and prototype built at IWF (b) [Weikert, et. al., 1998]

Due to the complex kinematics of these configurations, means have to be developed that will allow 'mapping' the design alternatives into the final machine properties in a straightforward and intuitive manner. To meet the demands of HSM, particular attention will have to be paid to the attenuation of dynamic tool path errors by applying suitable control architectures and design for controllability. The modeling approaches presented here have the potential to satisfy these requirements as they facilitate quick, intuitive and accurate insights into the dynamic properties of machine tools.

5.3 Parameter Identification

It is well known that not all physical parameters concerning the dynamic behavior of a system can be found (e.g., physical tables or calculations) with satisfactory accuracy. For instance, damping coefficients are typically very difficult to estimate experimentally. Symbolic modeling facilitates an alternative for in-process identification of these unknown parameters that would otherwise be poorly estimated.

If critical physical parameters of machine tools are monitored on-line, then fluctuations in their values could be used to indicate process changes, system faults, or possibly component failure [Spiewak and Di Corpo, 1991; Spiewak, 1995; Novak and Wiklund, 1996]. A systematic approach is sought on how to identify and estimate physical parameters of a system. Generic models built entirely from signals[26] are very popular in control engineering since they are usually sufficient for the design of control systems, however they do not reveal any information on individual system parameters. The knowledge of the physical parameters is required for problems such as determination of non-measurable constants in natural sciences, performance assessment for technical systems, supervision during on-line operation of technical processes, and quality control in manufacturing.

To complete the generated model, its coefficients have to be provided. These coefficients are functions of physical parameters of the modeled machine tool. While the known parameters, such as masses and moments of inertia can be usually found and entered into the model, the unknown parameters have to be estimated on-line. To accomplish this, a suitable algorithm is necessary that allows tuning the model parameters

[26] Signal based approach to identification.

such that its behavior is close to the behavior of the actual machine tool described by the measured signals. A block diagram detailing the entire model definition process is described in Fig. 5.4. A procedure that derives the model structure is shown on the left-hand side, while the estimation of the unknown parameters is shown on the right. This latter procedure can be performed either off-line or in real-time (required by advanced control algorithms). The combination of model derivation and parameter identification provides the *actual system model*, which consists of the general model structure and the estimated unknown parameters.

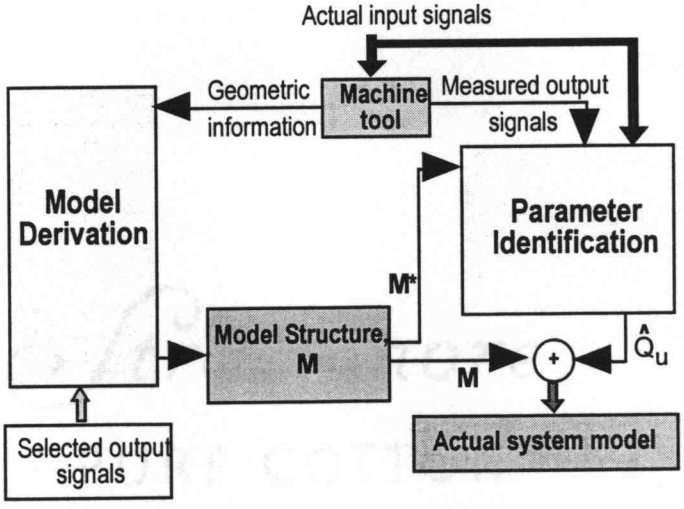


Fig. 5.4 : Integration of model derivation and parameter identification.

The three identification methods described below are based on: 1) transfer functions in the *s*-domain, 2) signals in the time domain, and 3) state estimation using the Kalman filter in the State Space domain [Nickel, 1998].

The transfer function method is based on the comparison of the transfer functions from the analytical model to those from experimental data. From the constitutive equations of motion analytical transfer functions are calculated which contain information about some or all of the physical parameters of the system. Comparisons of coefficients in the analytical and empirical transfer functions yield several equations involving the physical parameters. The task then is to solve this set of equations for the unknown

parameters. Unfortunately, this turns out to be a difficult task and an analytical closed form solution usually does not exist. Still, with the system's behavior known, and with exact measurements of input and output signals the determination of the physical parameters should be possible. Even if the system does not have an analytical closed form solution, iterative numerical methods can be used to obtain estimates of the physical parameters. The determination of the parameters becomes a statistical estimation problem, and it can be solved using global minimization methods.

The signal based method utilizes the theoretical transfer function model as well. However, experimental and theoretical transfer function coefficients are not compared. Rather, an output signal is simulated from the theoretical model, which is then compared with the respective experimental data. In order to estimate the physical parameters, errors between the simulated and experimental data have to be minimized with respect to the unknown values using, for example, the method of least squares. Therefore, the simulated output data has to be a function of the physical parameters. Since the experimental data is in discrete form, the theoretical transfer function of the continuous domain needs to be transformed into the discrete domain, using one of several methods[27].

The Kalman filter is used in the field of controls for giving optimal estimates of the internal states of a system [Dasgupta, et. al. 1988; Hong and Curtiss, 1993; Isermann, 1992; Maciejowski, 1989; Sorensen, 1980]. It is based on the state space representation of a system, and therefore can easily handle multi-input, multi-output (MIMO) systems.

The Kalman filter gains are calculated to give the best estimates of the state vector with respect to minimizing disturbances and measurement noise. It is employed for physical parameter estimation in the following manner. The state space formulation of the system is rewritten so that the unknown physical parameters are represented as internal states of the system. The Kalman filter can then be used to estimate those states, and hence find the unknown system constants. However, the identification process can only be accomplished if a constitutive model of the system exists.

---

[27] E.g., Tustin's method [DeCarlo, 1989].

5.4 Visualization of Vibrations

Visualization of vibrations serves two primary purposes. First, it offers a broader understanding of the system performance. Second, it provides for improved modeling since errors between simulated and actual responses can be immediately identified. The implications of modeling on vibration visualization is a broad field and well documented in literature; a good review of these papers is presented in the work of Jitpraphai [Jitpraphai, 1997; Wiekert, et. al., 1998]. What is presented here merely highlights some of the main points.

The visualization program developed at Oregon State University uses real signals from accelerometers attached at key locations on the vibrating rigid bodies to generate animated motion. The acquired signals are used in conjunction with the generated model of the dynamic system to detect and suppress errors in the signals.

The entire visualization scheme is implemented in two stages. The first stage, referred to as 'signal based vibration visualization' uses a suitable data acquisition and filtering system to collect information from acceleration sensors strategically located on the rigid bodies. The data from this stage are referred to as 'signal based responses' (SBR). The second stage takes each specific SBR and analyzes it to detect possible errors using a generated model of the system under consideration.

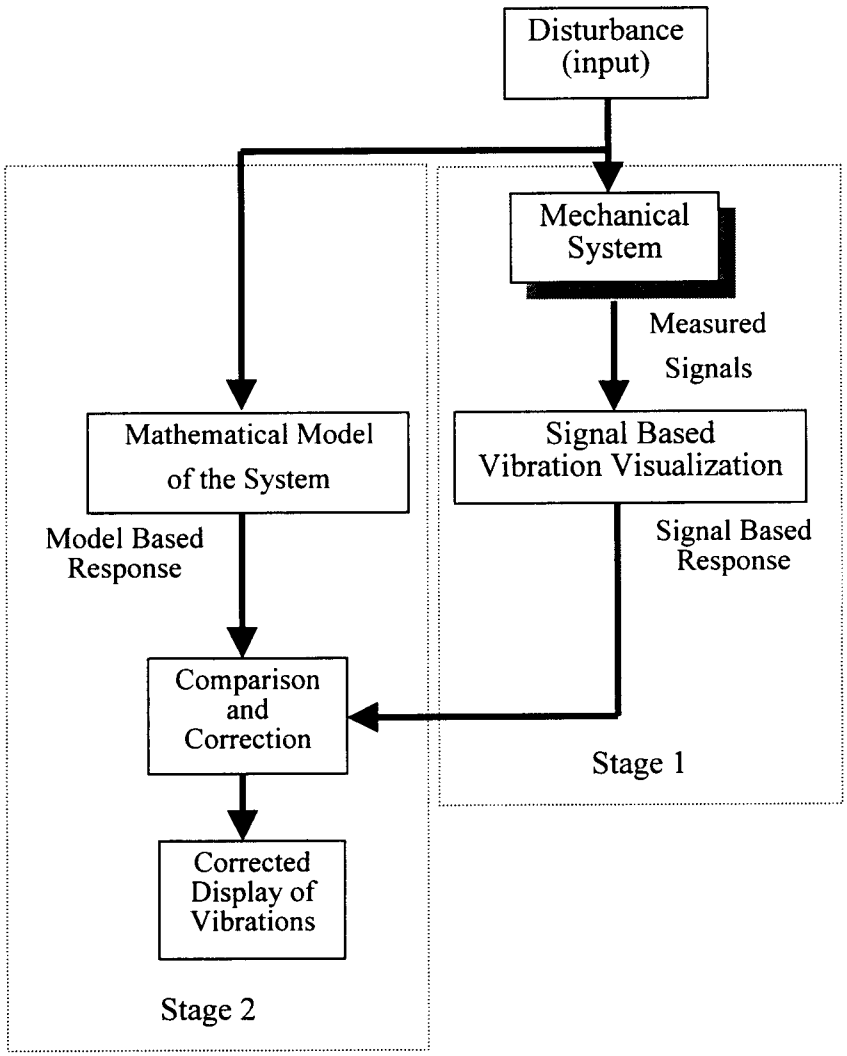


Fig. 5.5: Flowchart of model based visualization of vibrations [Jitpraphai, 1997].

This is accomplished by comparing the SBR with a theoretical response generated by stimulating the model with the actual input signal that acted on the physical system. The generated theoretical outputs are referred to as 'model based responses' (MBR). These outputs are generally a function of system parameters, some of which are unknown. For such cases the parameters have to be estimated using, for example, the procedures in the previous section. The final form of the *actual system model* is essential for identifying and facilitating the suppression of errors such as signal drift and faulty or uncalibrated sensors, which can go undetected otherwise.

5.5 Automated Supervision and Monitoring Systems

In the realm of machining processes it is desirable to monitor the machine tool's performance, both in speed and part tolerances, as well as the integrity of components. In terms of automated supervision and monitoring, dynamic models are used in several areas including adaptive compensation [Chung, 1993], predictive maintenance [Szafarczyk, 1990; Novak and Wiklund, 1996], and failure prediction [Tu, 1991; Willsky, 1976]. Again, this is a broad field of research that is well documented in literature.

Monitoring of machine tools and optimization of machining process require accurate on-line measurements concerning forces, torques, and displacements of the components in the structure [Spiewak and Di Corpo, 1991]. Commercially available sensors are highly accurate, but cannot accommodate the rapidly changing environment they are subjected to. Adjusting the sensors for a specific expected situation does little good since they fall out of calibration quickly due to the continually changing environment. In addition to this, introduction of a sensor (another coupled component) to the system changes the system's dynamic characteristics. Spiewak and Di Corpo presented a system that corrects the dynamic characteristics of sensors in a continuous fashion [Spiewak and Di Corpo, 1991]. The algorithm involved identification and estimation of the model parameters which was used in a model based adaptive filter that compensated the dynamic properties of the actual sensors. Based on the model of the system they were able to attain good results concerning the adaptive compensation of sensor signals in a machining process.

There are several benefits resulting from modeling for predictive maintenance. Down time on a machining process due to an unexpected failure can be costly. To remedy this, each critical component of the structure is inspected or replaced after a given time of service, but this is not necessarily cost effective either. There have been many instances where the part passed inspection or did not even need to be replaced, wasting man hours and part inventory. Since the ultimate goal is to predict a failure and identify which component will fail, an accurate model of the system is necessary that must be capable of monitoring the process and adapting to system property changes on-line. Constitutive modeling methodologies have this capability, and can also provide optimal locations for sensor placement by testing for observability and manipulating the observability matrix, **C** [DeCarlo, 1989]. Model based automated supervision has the potential to extend the

life of components, maximize run time, minimize down time, and avoid catastrophic failures by their early prediction.

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

The presented Computer-Aided Model Generation and Validation methodology provides the comprehensive means necessary to analyze the dynamic behavior of multi-body structures. The modeling environment facilitates versatility by allowing straightforward transformations to different forms and domains. The final results are symbolic expressions derived from the equations of motion. However, this approach is predicated upon the absence of significant low frequency flexible vibration modes in the system. This requirement can be well satisfied in the parallel structure designs.

An immediate advantage of the modeling approach is the ability to provide quick, easily interpretable checks concerning the system's future dynamic behavior. This information is important in the design phase, since potential difficulties can be detected and addressed immediately instead of having to face completely unexpected problems which require a large amount of time to eliminate when the machine has already been built. Another advantage of the approach is the ability to accommodate increasingly complex systems. Addition of rigid bodies and additional (or redundant) couplings is straightforward. This is particularly useful in situations where certain behavioral traits cannot be foreseen or eliminated in the design process (a predominant factor in parallel structures), and advanced control schemes must be employed to attain the required performance.

The modeling environment allows a number of techniques for validation to be readily implemented. This includes intuitive checks at key points during model derivation as well as applications of more traditional experimental validation. In all presented cases the analysis can be performed in the same software package that was used for model development.

Main advantages of using the above modeling methodology are as follows:

- Minimal user input requirements in the initial setup stages,

- Efficient mapping between parameters of the actual system and their representations in the model,
- Scalability and versatility of the model,
- Ability to transition between different model forms and domains in the same programming environment,
- Good computational efficiency due to the model's parsimony.

Integration of the generation, validation, and troubleshooting methodology delineated in this research facilitates development of accurate models that can be applied in structure design and exploitation. Possible applications of these models include parameter identification, visualization of vibration, automated supervision and monitoring, and design of advanced control strategies for minimization of dynamic tool path errors. The benefits are especially prevalent in parallel structure machine tools, where there is still a lack of experience. Latest developments in measurement techniques and the emergence of new sensors[28] facilitate reliable validation and optimization of the models.

6.2 Recommendations for Further Research

As with any modeling approach, the possible avenues for improvement of accuracy, efficiency, interpretability, and ease of use are bountiful. However, there are three immediate areas that should be explored. The first recommendation is implementation of the same methodology using a different formalism, such as Kane's method. It has been suggested that Kane's method produces equations in their most compact form; whether it is more efficient than Lagrange's method is still to be determined. In any case, an investigation on the suitability of various modeling approaches would prove beneficial for improving model accuracy and consistency.

The second recommendation is inclusion of more detail in the description of model components. These can be itemized in the following manner:

1) 'Composite' rigid bodies should be separated into their constitutive parts. For example, the assumed 'rigid' platform on the Hexaglide is actually composed of

[28] E.g., Heidenhain Cross Grid [Heidenhain Corp.].

a spindle, housing, and base, all elastically coupled through bearings and fasteners. This would allow for gyroscopic effects from spinning components, such as a rotating spindle.

2) The approximation of a 'rigid' body may not be sufficiently accurate, so inclusion of some flexible modes might be necessary. A long, slender spindle would be particularly subject to flexible mode vibrations, and hence would not exhibit rigid body behavior.
3) The SDE connections between the rigid bodies are meant to represent the stiffnesses of the joints and/or struts. However, most joints do not exhibit uniform stiffness for every direction, and if this is found to be significant it must be taken into account.

Finally, the third recommendation is to perform more extensive model validation. Development of a versatile modeling environment is a time consuming task; model validation is even more so. It was decided to focus the bulk of this research on model development, and as a result, there is still a broad field to investigate in terms of experimental validation. Several methods of intuitive validation were presented that were closely related to model development, but there are also a large number of experimental analyses that have yet to be evaluated.

BIBLIOGRAPHY

Aini, R., Rahnejat, H., and Gohar, R., "A Five Degrees of Freedom Analysis of Vibrations in Precision Spindles.", Int. Journal of Machine Tool and Manufacturing, Vol. 30, No. 1, pp. 1-18, 1990.

Allemag, R. J., Brown, D. L., "A Correlation Coefficient for Modal Vector Analysis.", 1st International Modal Analysis Conference, Orlando, Florida, pp 110-116, 1982.

Amsterdam, J., "Automated Modeling of Physical Systems", Automated Modeling, ASME, DSC-Vol. 41, 1992.

Barhorst, A. A., "Symbolic Equation Processing Utilizing Vector/Dyad Notation", Journal of Sound and Vibration, Vol. 208, No. 5, pp. 823-839, 1997.

Bianchi, G., Paolucci, CF., Van den Braembussche, P., Van Brussel, H., "Towards Virtual Engineering in Machine Tool Design", Annals of the CIRP, Vol. 45, No. 1, pp. 381-384, 1996.

Billings, S. A., "Identification of nonlinear Systems - A Survey", IEE Proc. D, Vol. 127, pp. 272-285, 1980.

Box, G., Jenkins, G., Time Series Analysis: Forecasting and Control, Holden-Day, Inc., 1976.

Carne, T. G. et. al., "Finite Element Analysis and Modal Testing of a Rotating Wind Turbine", International Journal of Analytic and Experimental Modal Analysis, Vol. 3, p. 33, 1988.

Cesareo, G., Nicolo, F., "DYMIR: A Code for Generating Dynamics Models of Robots", IEEE International Conference on Robotics, Atlanta, GA, 1984.

Chen, B., "Computer-Aided Derivation of State Space Models for Linear Dynamic Systems", M.S. Thesis, Oregon State University, 1996.

Cheung, Y. K., Leung, A. Y. T., Finite Element Methods in Dynamics, Science Press, Kluwer Academic Publishers, 1991.

Chung, Y., "Model Based Adaptive Compensation of Dynamic Characteristics of In-Process Sensors.", Ph.D. Thesis, University of Wisconsin-Madison, 1993.

Codourey, A., Burdet, E., "A Body-Oriented Method for Finding a Linear Form of the Dynamic Equation of Fully Parallel Robots", Institute of Robotics, ETH Zurich, 1997.

Comparin, R. J., "The Dynamic Analysis of a Machine Tool Spindle", M.S. Thesis, Ohio State University, 1983.

Dally, J. W., Riley, W. F., McConnell, K. G., Instrumentation for Engineering Measurement, John Wiley & Sons, Inc., 1993.

Dasgupta, S., Anderson, B. D. O., Kaye, R. I., "Identification of Physical Parameters in Structured systems." Automatica, Vol. 24, No. 2, pp. 217-225, 1988.

Datel, FLJ-D5 Digital-Programmable High-Order Low-Pass Filter Product Data Sheet, Datel, 1987.

DeCarlo, R. A., Linear Systems, Prentice-Hall, Inc., 1989.

Denavitt, J., Harternberg, R. S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", ASME Journal of Applied Mechanics, Vol. 22, pp. 215-221, 1955.

Doebelin, E. O., Measurement Systems: Application and Design, McGraw-Hill, Inc., 1990.

Doeblin, O. E., System Modeling and Response: Theoretical and Experimental Approaches, John Wiley and Sons, Inc., 1980.

Ewins, D. J., Modal Testing: Theory and Practice, John Wiley & Sons, Inc., 1984.

Faessler, H., "Computer-Assisted Generation of Dynamical Equations for Multibody Systems", Intl. Journal of Robotics Research, Vol. 5, No. 3, 1986.

Fagan, M. J., Finite Element Analysis: Theory and Practice, Longman Scientific & Technical, Wiley Harlow, 1992.

Fichter, E. F., "A Stewart Platform-Based Manipulator: General Theory and Practical Construction", International Journal of Robotics Research, Vol. 5, No. 2, 1986.

Franklin, G. F.,Powell, J. D., Emami-Naeini, A., "Feedback Control of Dynamic Systems.", Addison-Wesley, 1994.

Friswell, M. I., Mottershead, J. E., Finite Model Updating in Structural Mechanics, Kluwer Academic Publishers, 1995.

Fritzen, C. P., "Identification of Mass, Damping and Stiffness Matrices of Mechanical Systems.", ASME Journal of Vibrations, Acoustics, Stress and Reliability in Design, Vol. 108, pp. 9-16, 1986.

Gaonkar, G., Ravichandran, S., Nagabhushanam, J., Reddy, T. S. R., "A Study of Symbolic Processing and Computational Aspects in Helicopter Dynamics", Journal of Sound and Vibration, Vol. 137, No. 3, pp. 495-507, 1990.

Garcia de Jalon, J., Bayo, E., Kinematic and Dynamic Simulation of Multibody Systems, Springer-Verlag, 1994.

Ginsberg, J. H., Advanced Engineering Dynamics, Cambridge University Press, 1995.

Gysin, H., "Ein Beitrag zur Verbesserung von Finite-Elemente-Modellen anhand modaler Parameter", Ph.D. Thesis, ETH Nr. 9849.

Hale, A. L., Meirovitch, L., "Derivation of the Equations of Motion for Complex Structures by Symbolic Manipulation", Computers and Structures, Vol. 9, pp. 639-649, 1978.

Hebsacker, M.; Statische Auslegung und Optionen von Hexaglide-Maschinen. M.S. Thesis, IWF-ETHZ.

Heidenhain Corporation, "Measuring Systems for Machine Tool Inspection and Acceptance Testing", Schaumburg IL.

Heisel, U., "Machine Tool Design Requirements for High-Speed Machining", Annals of the CIRP, Vol. 45, No. 1, 1996.

Honegger, M., Codourey, A., Burdet, A., "Adaptive Control of the Hexaglide, a 6 dof Parallel Manipulator", Proc. of the ICRA, 1997.

Hong, S. W., Curtiss, H. C., "An Analytic Modeling and System Identification Study of Rotor/Fuselage Dynamics at Hover", Proc. of Conference on Flying Qualities and Human Factors, San Francisco, CA, 1993.

Huang, T., Whitehouse, D. J., Wang, J., "The Local Dexterity, Optimal Architecture and Design Criteria of Parallel Machine Tools", Annals of the CIRP, Vol. 47, No. 1, 1998.

Inman, D. J., Engineering Vibration, Prentice Hall, 1994.

Isermann, R., "Estimation of physical parameters for dynamic processes with application to an industrial robot", International Journal of Control, Vol. 55, No. 6, pp. 1287-1298, 1992.

Isermann, R., "Process Fault Detection Based on Modeling and Estimation Methods – A Survey." Automatica, Vol. 20, No. 4, pp. 387-404, 1984.

Jansen, H., Notes from Multi-Particle Dynamics Lectures, 1997.

Jitpraphai, T., "Model Based Visualization of Vibrations in Mechanical Systems", M.S. Thesis, Oregon State University, 1997.

Kane, T. R., Levinson, D. A., Dynamics: Theory and Applications, McGraw-Hill, Inc., 1985.

Kane, T. R., Likins, P. W., Levinson, D. A., Spacecraft Dynamics, McGraw-Hill, Inc., 1983.

Karnopp, D., Rosenberg, R., System Dynamics: A Unified Approach, John Wiley & Sons, Inc., 1975.

Kistler Instrument Corp., Advanced Instrumentation for a World of Applications, Kistler Instrument Corp., 1995.

Kreuzer, E. J., Schiehlen, W. O., "Generation of Symbolic Equations of Motion for Complex Spacecraft using Formalism NEWEUL", Aeronautical and Astronautical society, Vol. 83.

Leuridan, J. M., "Direct System Parameter Identification of Mechanical Structures with Application to Modal Analysis", M. S. Thesis, University of Cincinnati, 1981.

Levinson, D. A., "The Derivation of Equations of Motion of Multiple-Rigid-Body-Systems using Symbolic Manipulation", Journal of Spacecraft Rockets, Vol. 14, No. 8, pp. 479-487, 1977.

Lieh, J., Haque, I., "Symbolic Closed-Form Modeling and Linearization of Multibody Systems Subject to Control", Journal of Mechanical Design, Vol. 113, pp. 124-132, 1991.

Ljung, L., "System identification in a MIC perspective", Modeling, Identification and Control, Vol. 15, No. 3, pp. 153-159, 1994.

Macala, G. A., "SYMBOD: A Computer Program for the Automatic Generation of Symbolic Equations of Motion for Systems of Hinged-Connected Rigid Bodies", AIAA 21st Aerospace Science Meeting, Reno, NV, 1983.

Maciejowski, J. M., Multivariable Control Design, Addison-Wesley, 1989.

Marion, J. B., Thornton, S. T., Classical Dynamics of Particles and Systems, Harcourt Brace Jovanovich, Inc., 1988.

Matsubara, M., et. al., "Computational Modelling of Precision Spindles Supported by Ball Bearings", Int. Journal of Machine Tool and Manufacturing, Vol. 28, No. 4, pp. 429-442, 1988.

Miller, D. G., White, F., "A Treatment of the Impact of Rotor-Fuselage Coupling on Helicopter Handling Qualities", Proc. of 43rd Annual National Forum of the American Helicopter Society, 1987.

Mitchell, J. S., "An Introduction to Machinery Analysis and Monitoring", Pen Well Publishing Co., 1981.

Mottershead, J.E., Friswell, M. I., "Model updating in structural dynamics: A Survey", Journal of Sound and Vibration, Vol. 167, No. 2, 1993.

National Instruments, AT-MIO E Series User Manual, National Instruments Corp., 1995.

Nickel, T., "Physical Parameter Estimation for Monitoring and Diagnosis of Mechanical Systems", Ph.D. Thesis (to be published), Oregon State University, 1998.

Novak, A., Wiklund, H., "On-Line Prediction of the Tool Life", Annals of the CIRP, Vol. 45, No. 1, pp. 93-96, 1996.

O'Reilly, J., "Observers for Linear Systems", Springer-Verlag, pp. 115-119, 1994.

Pandit, S. M., Modal and Spectrum Analysis: Data Dependent Systems in State Space, John Wiley & Sons, Inc., 1991.

Precision, 88 Series 8 Pole, 8 Zero Programmable Filter Modules, Precision Filters, Inc., 1989.

Rake, H., "Step Response and Frequency Response Methods", Automatica, Vol. 16, No. 2, pp. 519-526, 1980.

Reddy, V. R., Sharan, A. M., "The Finite Element Modeled Design of Lathe Spindles: The Static and Dynamic Analysis", ASME Journal of Vibrations, Acoustics, Stress and Reliability in Design, Vol. 109, pp. 407-415, 1987.

Rehsteiner, F., Weikert, S., "Analysis of Tool Trajectories (2D Tool Paths) in Machine Tools", Proceedings ASPE 1996 Annual Meeting, Monterey, CA, pp. 288-293, 1996.

Rosenthal, D. E., Sherman, M. A., "High Performance Multibody Simulations via Symbolic Equation Manipulation and Kane's Method", Journal of the Astronautical Sciences, Vol. 34, No. 3, pp. 223-239, 1986.

Sandor, G. N., Erdman, A. G., Advanced Mechanism Design: Analysis and Synthesis, Prentice-Hall, 1984.

Schaechter, D. B., Levinson, D. A., "Interactive Computerized Symbolic Dynamics for the Dynamicist", Journal of the Astronautical Sciences, Vol. 36, No. 4, pp. 365-388, 1988.

Scheck, F., Mechanics, Springer-Verlag, 1994.

Segalman, D. J., Dohrmann, C. R., "A Method for Calculating the Dynamics of Rotating Flexible Structures, Part 1: Derivation", Journal of Vibrations and Acoustics, Vol. 118, pp. 313-317, 1996.

Shabana, A. A., Theory of Vibration, Springer-Verlag, 1991.

Shin, Y. C., Wang, K. W., Chen, C. H., "Dynamic Analysis and Modeling of a High Speed Spindle System", Transactions of NAMRI/SME, pp. 298-304, 1990.

Sorensen, H. W., "Parameter Estimation: Principles and Problems", Dekker, 1980.

Spiewak, S. A., "Acceleration Based Indirect Force Measurement in Metal Cutting Processes", International Journal of Machine Tools and Manufacturing, Vol. 35, No. 1, pp. 1-17, 1995.

Spiewak, S. A., "Emerging Trends in Automatic Supervision of Machine Tools", Proc. Intl. Conf. on Monitoring and Automatic Supervision in Manufacturing, Warsaw, pp. 49-60, 1995.

Spiewak, S. A., Di Corpo, J., "Adaptive Compensation of Dynamic Characteristics for In-Process Sensors.", ASME Journal of Engineering for Industry, Vol. 113, pp. 198-206, 1991.

Srinivasan, K., Kulkarni, P. K., "Cross-Coupled Control of Biaxial Feed Drive Servomechanism", ASME Journal of Engineering for Industry, Vol. 112, pp. 225-232, 1990.

Szafarczyk, C., "Automated Supervision in Manufacturing Systems. Principles, Classification and Terminology", Proceedings of the 3rd International Conference on Automatic Supervision, Monitoring and Adaptive Control in Manufacturing, pp. 7–22, 1990.

Szafarczyk, M., Automatic Supervision in Manufacturing, Springer-Verlag, pp. 115-119, 1994.

Takata, S., Sata, T., "Model Referenced Monitoring and Diagnosis – Application to the Manufacturing System", Computers in Industry, 1986.

Thomson, W. T., Theory of Vibration with Applications, Prentice-Hall, 1981.

Tosatti, L., Bianchi, G., Fassi, I., Boer, C. R., Jovane, F., "An Integrated Methodology for the Design of Parallel Kinematic Machines (PKM)", Annals of the CIRP, Vol. 47, No. 1, 1998.

Tu, J. F., "On-line Preload Monitoring for High-speed Anti-friction Spindle Bearings Using Robust State Observers", Ph.D. Thesis, University of Michigan, 1991.

Valenti, M., "Machine Tools Get Smarter", Mechanical Engineering, ASME, Vol. 117, No. 11, pp. 70-75.

Van Brussel, H., et.. al., "Comparative Assessment of Harmonic, Random, Swept Sine and Shock Excitation Methods for the Identification of Machine Tool Structures with Rotating Spindles", Annals of CIRP, Vol. 24, pp. 291-296, 1975.

Warnecke, H., Neugebauer, R., Wieland, F., "Development of Hexapod Based Machine Tool", Annals of the CIRP, Vol. 47, No. 1, 1998.

Weaver, W. Jr., Johnston, P. R., Structural Dynamics by Finite Elements, Prentice-Hall, 1987.

Weck, M., Handbook of Machine Tools, John Wiley, 1984.

Weikert, S., Rehsteiner, F., Brisbine, B., Spiewak, S. A., "Model Based Minimization of Dynamic Tool Path Errors", Proc. Second SAE Aerospace Manufacturing Technology Conference, Long Beach CA, 1998.

Weikert, S., Rehsteiner, F., Brisbine, B., Spiewak, S. A., "Modeling and Assessment of Machine Tool Dynamics and Accuracy", Proc. First SAE Aerospace Manufacturing Technology Conference, Seattle WA, pp. 91-101, 1997.

Wellstead, P. E., "Non Parametric Methods of System Identification", Automatica, Vol. 17, No. 1, pp. 53-69, 1981.

Willsky, A. S., "A Survey of Design Methods for Failure Detection in Dynamic Systems", Automatica, Vol. 12, pp. 601-611, 1976.

Wolfram Research, Inc., *Mathematica - Control System Professional*, Wolfram Research, Inc., 1996.

Wolfram Research, Inc., *Mathematica*, Wolfram Research, Inc., 1991.

Zatarain, M., Lejardi, E., Egana, F., "Modular Synthesis of Machine Tools", Annals of the CIRP, Vol. 47, No. 1, 1998.

APPENDICES

Appendix A: Transformation Matrices

In order to obtain a consistent description of a system comprising several rigid bodies depicted in Fig. 3.1, the transformation of coordinates from local to global reference frames is essential. Since this process has a profound impact on the accuracy of representation, it is worthwhile to describe it in more detail. Coordinate transformations are accomplished by the combination of four matrices: three rotation matrices (one about each axis) and a translation matrix. Since this is a general formulation, it can be applied to any rigid body in the system, and hence provides the ability to express all local positions in terms of global coordinates.

Two specific transformations are required for each rigid body. The first is purely rotational, and is used in the kinetic energy to modify the inertia tensor. The second involves the combination of rotations and translations. This is used to express the local connection points of the SDEs in terms of global coordinates for use in the calculation of potential and damping energies.

The spatial rotations required to arrive at a particular orientation are dependent on the orientation of each rotation axis, the amount of rotation about each axis, and (although not as apparent) the sequence of rotations [Marion, Thornton, 1988; Ginsberg, 1995]. The two most common sequences are referred to as body-fixed rotations and space-fixed rotations. For body-fixed rotations, each rotation is about a particular axis at the preceding step of the sequence. In contrast, each step of a space-fixed rotation is about a global axis (e.g., about the original coordinate system). Selection of which order of rotations to use is not critical, since is has been shown that they have a simple relationship; namely, that one sequence of rotations is equivalent to the inverse order of the other sequence [Ginsberg, 1995]. Space-fixed rotations were chosen in this research merely for ease of interpretability.

The matrices representing rotations about the three axes are of the traditional form. For the x-axis

$$\mathbf{R}_{\Theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta_i) & -\sin(\Theta_i) \\ 0 & \sin(\Theta_i) & \cos(\Theta_i) \end{bmatrix} \tag{A.44}$$

Similarly, for rotation about the y-axis and z-axis

$$\mathbf{R}_{\Phi} = \begin{bmatrix} \cos(\Phi_i) & 0 & \sin(\Phi_i) \\ 0 & 1 & 0 \\ -\sin(\Phi_i) & 0 & \cos(\Phi_i) \end{bmatrix}, \quad \mathbf{R}_{\Psi} = \begin{bmatrix} \cos(\Psi_i) & -\sin(\Psi_i) & 0 \\ \sin(\Psi_i) & \cos(\Psi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.45,46}$$

The angles Θ_i, Φ_i, and Ψ_i refer to rotation of the i-th rigid body with respect to the global frame (see Fig. 3.6). For a rotation sequence of x, y, then z, the space-fixed expression for the (rotational only) transformation in Eq. (3.16) (used to modify the inertia tensor) becomes

$$\boldsymbol{\lambda}_{i,R} = \mathbf{R}_{\Theta}\mathbf{R}_{\Phi}\mathbf{R}_{\Psi} \tag{A.47}$$

The inclusion of translation for the transformation of connection points belong to a relatively new class of *homogeneous* transformation matrices [Denavitt, Harternberg, 1955; Sandor, Erdman, 1984]. The combination of translation and rotation requires slight modification of the rotation matrices given in Eqs. (A.44), (A.45), and (A.46). They include an extra row and column for utility purposes

$$\mathbf{R}'_{\Theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Theta_i) & -\sin(\Theta_i) & 0 \\ 0 & \sin(\Theta_i) & \cos(\Theta_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.48}$$

The remaining two, $\mathbf{R}'_{\phi}$ and $\mathbf{R}'_{\varphi}$, follow the same form. The matrix form for translation only is

$$\mathbf{T}_{XYZ} = \begin{bmatrix} 1 & 0 & 0 & X_i \\ 0 & 1 & 0 & Y_i \\ 0 & 0 & 1 & Z_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.49}$$

Where X_i, Y_i, and Z_i refer to global coordinate positions of the i-th rigid body center of mass. The complete homogeneous (translational and rotational) transformation matrix for the i-th rigid body is given by the combination of the modified rotation matrices and Eq. (3.49)

$$\boldsymbol{\lambda}_{i,TR} = \mathbf{T}_{XYZ}\mathbf{R}'_{\Theta}\mathbf{R}'_{\Phi}\mathbf{R}'_{\Psi} \tag{A.50}$$

This transformation is used in the definition of the SDE deflections in Eqs. (3.22) and (3.23).

Appendix B: Analysis of a Stewart Platform in *Mathematica*

Model of Stewart Platform Generating Equations of Motion

- Development History

Started:	10/1/97:	B. Brisbine
last Rev.:	6/11/98:	B. Brisbine

1. Call packages (Check if loaded succesfully) & clear numerical constants and variables

1.1 Load packages.

```
<< Utilities`CleanSlate`
```

- Clear all things

```
CleanSlate[];
```

```
<< Utilities`Notation`
```

```
<< "LinearAlgebra`MatrixManipulation`"
```

```
<< "Calculus`VectorAnalysis`"
```

```
<< "ControlSystems`Master`"
```

2. Set properties of the working environment

```
Off [ General::spell ]
```

```
Off [ General::spell1 ]
```

■ 3. Vectors of coordinates and system parameters

■ 3.1 Total coordinates ($q_i^* = q_{i,0} + q_i$).

■ 3.1.1 Platform coordinates.

```
pTrans = {x1°[t], y1°[t], z1°[t]};
pRot = {θ1°[t], ϕ1°[t], φ1°[t]};
```

■ 3.1.2 Leg-base coordinates.

```
legTrans = {xb1°[t], xb2°[t], xb3°[t], xb4°[t], xb5°[t], xb6°[t]};
```

■ 3.1.3 Combine all system coordinates.

```
(*Coord = Join[pTrans, pRot, legTrans];*)
Coord = Join[pTrans, pRot];
n = Dimensions[Coord];
MatrixForm[Coord]
```

■ 3.1.4 Define the system's active degrees of freedom (total DOF).

form: {x, y, z, θ, ϕ, φ, xb1, xb2, xb3, xb4, xb5, xb6}

```
(*DOF = {1,1,1,1,1,1,1,1,1,1,1,1};*)
DOF = {1, 1, 1, 1, 1, 1};
```

■ 3.2 Nominal values of platform (NC programmed path).

```
Symbolize[x1₀]; Symbolize[y1₀]; Symbolize[z1₀];
Symbolize[θ1₀]; Symbolize[ϕ1₀] Symbolize[φ1₀];

L = 0.838; g = 9.81;
NomVal = {0, 0, 0.86, 0, 0, 0};
(*NomVal={x1₀,y1₀,z1₀,θ1₀,ϕ1₀,φ1₀};*)
pDOF = Dimensions[NomVal];
MatrixForm[NomVal]
```

■ 3.3 Generalized coordinates (incremental values).

```
pGenTrans = {x1[t], y1[t], z1[t]};
pGenRot = {θ1[t], ϕ1[t], φ1[t]};
legGenTrans = {xb1[t], xb2[t], xb3[t], xb4[t], xb5[t], xb6[t]};
pGenTransD = ∂t pGenTrans;
pGenRotD = ∂t pGenRot;
legGenTransD = ∂t legGenTrans;
(*GenCoord= Join[pGenTrans, pGenRot, legGenTrans];
GenCoordD=Join[pGenTransD, pGenRotD, legGenTransD];
transGenCoordD = Join[pGenTransD, legGenTransD];*)
GenCoord = Join[pGenTrans, pGenRot];
GenCoordD = Join[pGenTransD, pGenRotD];
transGenCoordD = Join[pGenTransD];
GenCoordDD = ∂t GenCoordD;
MatrixForm[GenCoord]
```

■ 3.4 Mass vector for platform and servo motors.

format: $\{m_{px}, m_{py}, m_{pz}, m_{sm1}, m_{sm2}, m_{sm3}, m_{sm4}, m_{sm5}, m_{sm6}\}$

```
(*massV = {mP, mP,mP,mL, mL, mL,mL, mL, mL};*)
massV = {mP, mP, mP};
trans = Dimensions[massV];
```

■ 3.5 Inertia matrix for platform.

```
Symbolize[JPxx]; Symbolize[JPyy]; Symbolize[JPzz];

inertiaM = DiagonalMatrix[{JPxx, JPyy, JPzz}];
MatrixForm[inertiaM]
```

■ 3.6 Leg and servo motor stiffness and damping vectors.

```
Symbolize[KLS]; Symbolize[KSS]; Symbolize[BLS];
Symbolize[BSS];

KLS = {KLS, KLS, KLS, KLS, KLS, KLS};
KSS = {KSS, KSS, KSS, KSS, KSS, KSS};
BLS = {BLS, BLS, BLS, BLS, BLS, BLS};
BSS = {BSS, BSS, BSS, BSS, BSS, BSS};
```

■ 4. Transformation matrices

■ 4.1 Type x transformation matrix : rotation about the X axis.

```
TMx[θ_] := {{1, 0, 0, 0}, {0, Cos[θ], -Sin[θ], 0}, {0, Sin[θ], Cos[θ], 0}, {0, 0, 0,
1}}
MatrixForm [TMx[θ]]
```

■ 4.2 Type y transformation matrix : rotation about the Y axis.

```
TMy[ϕ_] := {{Cos[ϕ], 0, Sin[ϕ], 0}, {0, 1, 0, 0}, {-Sin[ϕ], 0, Cos[ϕ], 0}, {0, 0, 0,
1}}
MatrixForm [TMy[ϕ]]
```

■ 4.3 Type z transformation matrix : rotation about the Z axis.

```
TMz[φ_] := {{Cos[φ], -Sin[φ], 0, 0},
  {Sin[φ], Cos[φ], 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}}
MatrixForm[TMz[φ]]
```

■ 4.4 Type t transformation matrix : translation only.

```
TMt[x_, y_, z_] := {{1, 0, 0, x}, {0, 1, 0, y}, {0, 0, 1, z}, {0, 0, 0, 1}}
MatrixForm[TMt[x, y, z]]
```

■ 4.5 Platform transformation matrix: total rotation only.

```
TMtrP = TMx[Coord[[4]]] . TMy[Coord[[5]]] . TMz[Coord[[6]]];
TMrP = TMtrP[[Range[1, 3], Range[1, 3]]] /.
  {θ1°[t] → NomVal[[4]], ϕ1°[t] → NomVal[[5]], φ1°[t] → NomVal[[6]]};
```

■ 4.6 Platform transformation matrix : total rotation & translation.

```
TMP = TMt[Coord[[1]], Coord[[2]], Coord[[3]]] . TMtrP;
MatrixForm[TMP]
```

■ 5. Leg endpoint positions (For PE of leg "springs")

■ 5.1 Define position vectors, where legs are connected to the tool platform.

```
(*LP1 = {p1x, p1y, p1z, 1};
LP2 = {p2x, p2y, p2z, 1};
LP3 = {p3x, p3y, p3z, 1};
LP4 = {p4x, p4y, p4z, 1};
LP5 = {p5x, p5y, p5z, 1};
LP6 = {p6x, p6y, p6z, 1};*)
LP1 = {-0.0738, -0.1278, -0.075, 1};
LP2 = {0.1475, 0, -0.075, 1};
LP3 = {0.1475, 0, -0.075, 1};
LP4 = {-0.0738, 0.1278, -0.075, 1};
LP5 = {-0.0738, 0.1278, -0.075, 1};
LP6 = {-0.0738, -0.1278, -0.075, 1};
vectLP = Transpose[{LP1, LP2, LP3, LP4, LP5, LP6}];
m = Dimensions[vectLP];
MatrixForm [vectLP]
```

■ 5.2 Transform all coordinates to inertial frame (base).

```
Symbolize[vectLPt₀];
```

```
vectLPt = TMP . vectLP;
vectLPt₀ =
 vectLPt /. {x1°[t] → NomVal[[1]], y1°[t] → NomVal[[2]], z1°[t] → NomVal[[3]],
   θ1°[t] → NomVal[[4]], ϕ1°[t] → NomVal[[5]], φ1°[t] → NomVal[[6]]};
MatrixForm[vectLPt₀]
```

■ 5.3 Determination of guideway positions by inverse kinematics.

■ 5.3.1 Calculate total guideway position vectors.

```
Symbolize[xb₀]; Symbolize[yb₀]; Symbolize[zb₀];
```

```
(*xb°=Array[temp2,m[[2]];
For[i=1,i≤m[[2]],i++,temp2[i]=Coord[[pDOF[[1]]+i]];*)
(*yb₀={-0.1278,0,0,0.1278,0.1278,-0.1278};*)
yb₀ = {-0.3818, -0.2525, 0.2525, 0.3818, 0.1292, -0.1292};
(*yb₀={yb1,yb2,yb3,yb4,yb5,yb6};*)
zb₀ = {0, 0, 0, 0, 0, 0};
(*vectLB={xb°,yb₀,zb₀,{1,1,1,1,1,1}};
MatrixForm[vectLB]*)
```

■ 5.3.2 Calculate nominal values for guideway leg positions from platform nominal values.

```
Symbolize[vectLB₀];

legPos = {-1, 1, 1, -1, -1, -1};
xb₀ = Array[temp1, m[[2]]];
For[i = 1, i ≤ m[[2]], i++, temp1[i] = vectLPt₀[[1, i]] + legPos[[i]]
    √(L² - (vectLPt₀[[2, i]] - yb₀[[i]])² - (vectLPt₀[[3, i]] - zb₀[[i]])²)];
vectLB₀ = {xb₀, yb₀, zb₀, {1, 1, 1, 1, 1, 1}};
MatrixForm[vectLB₀]
```

■ 6. Potential energy

■ 6.1 Leg (base-platform) springs.

■ 6.1.1 Deflection of leg springs.

```
DefLS = Array[temp3, m[[2]]];
(*For[i=1,i≤m[[2]],i++,temp3[i]=
   √((vectLPt[[1,i]]-vectLB[[1,i]])²+(vectLPt[[2,i]]-vectLB[[2,i]])²+
       (vectLPt[[3,i]]-vectLB[[3,i]])²)-L];*)
For[i = 1, i ≤ m[[2]], i++, temp3[i] =
  √((vectLPt[[1, i]] - vectLB₀[[1, i]])² + (vectLPt[[2, i]] - vectLB₀[[2, i]])² +
      (vectLPt[[3, i]] - vectLB₀[[3, i]])²) - L];
MatrixForm[DefLS]
```

■ 6.1.2 Calculation of leg spring potential energy.

$$ULS = \sum_{i=1}^{m[2]} \frac{1}{2} KLS[[i]]\, DefLS[[i]]^2;$$

■ 6.2 Servo (leg-motor) springs.

■ 6.2.1 Calculation of servo spring potential energy.

```
(*USS=∑_{i=1}^{m[[2]]} 1/2 KSS[[i]] (Coord[[pDOF[[1]]+i]]-xb₀[[i]])²;*)
USS = 0;
```

■ 6.3.1 Calculation of gravitational potential energy.

```
UG = g massV[[3]] Coord[[3]];
```

■ 6.4 Calculation of total potential energy, U_t.

```
Symbolize[Ut];
```

$$U_t = ULS + USS + UG$$

■ 6.5 Taylor Series expansion of total potential energy (U_t).

■ 6.5.1 Verify that the first term is zero.

```
U1 = Chop[
  Ut /. {x1'[t] → NomVal[[1]], y1'[t] → NomVal[[2]], z1'[t] → NomVal[[3]],
     θ1'[t] → NomVal[[4]], ϕ1'[t] → NomVal[[5]], φ1'[t] → NomVal[[6]]
     (*, xb1'[t]→xb0[[1]], xb2'[t]→xb0[[2]], xb3'[t]→xb0[[3]],
      xb4'[t]→xb0[[4]], xb5'[t]→xb0[[5]], xb6'[t]→xb0[[6]]*)} /.
   {√L² → L}]
```

■ 6.5.2 Verify that the second term is zero.

```
U2 = Chop[ Sum[GenCoord[[i]] ∂Coord[i] Ut, {i, 1, n[1]}] /.
    {x1'[t] → NomVal[[1]], y1'[t] → NomVal[[2]], z1'[t] → NomVal[[3]],
     θ1'[t] → NomVal[[4]], ϕ1'[t] → NomVal[[5]], φ1'[t] → NomVal[[6]]
     (*, xb1'[t]→xb0[[1]], xb2'[t]→xb0[[2]], xb3'[t]→xb0[[3]],
      xb4'[t]→xb0[[4]], xb5'[t]→xb0[[5]], xb6'[t]→xb0[[6]]*)} /.
   {√L² → L}]
```

■ 6.5.3 Calculation of the third term.

```
A3 = Simplify[Array[temp4, {n[[1]], n[[1]]}]];
For[i = 1, i ≤ n[[1]], i++,
 For[j = 1, j ≤ n[[1]], j++, temp4[i, j] = ∂Coord[i],Coord[j] Ut /.
     {x1'[t] → NomVal[[1]], y1'[t] → NomVal[[2]], z1'[t] → NomVal[[3]],
      θ1'[t] → NomVal[[4]], ϕ1'[t] → NomVal[[5]], φ1'[t] → NomVal[[6]]
      (*, xb1'[t]→xb0[[1]], xb2'[t]→xb0[[2]], xb3'[t]→xb0[[3]],
       xb4'[t]→xb0[[4]], xb5'[t]→xb0[[5]], xb6'[t]→xb0[[6]]*)} /.
    {√L² → L}]];
MatrixForm[A3]
```

$$U3 = Chop\left[\frac{1}{2}\sum_{i=1}^{n[1]} GenCoord[[i]] \sum_{j=1}^{n[1]} GenCoord[[j]]\, A3[[i, j]]\right]$$

■ 7. Kinetic energy

■ 7.1 Calculation of translational kinetic energy (platform and leg bases).

```
Symbolize[T_trans];
```

$$T_{trans} = \sum_{i=1}^{trans[1]} \frac{1}{2} \text{massV}[i] \; \text{transGenCoordD}[i]^2$$

■ 7.2 Calculation of rotational kinetic energy (platform).

```
Symbolize[T_rot];
```

$$T_{rot} = \frac{1}{2} \text{pGenRotD} \cdot \text{TMrP} \cdot \text{inertiaM} \cdot \text{Transpose}[\text{TMrP}] \cdot \text{pGenRotD}$$

■ 7.3 Calculation of total kinetic energy, T_t.

```
Symbolize[T_t];
```

$$T_t = T_{trans} + T_{rot}$$

■ 8. Damping energy

■ 8.1 Leg dampers.

$$\text{DampLS} = \sum_{i=1}^{m[2]} \frac{1}{2} \text{BLS}[i] \; (\partial_t \text{DefLS}[i])^2;$$

■ 8.2 Servo dampers.

```
(*DampSS=∑_{i=1}^{m[2]} 1/2 BSS[i] (∂_t (Coord[pDOF[1]+i]-xb_o[i]))^2*)
DampSS = 0;
```

■ 8.3 Total damping energy, Damp_t.

```
Symbolize[Damp_t];
```

$$\text{Damp}_t = \text{DampLS} + \text{DampSS};$$

■ 8.4 Taylor Series expansion of damping energy.

```
Damp1 =
 Damp_t /. {x1'[t] → NomVal⟦1⟧, y1'[t] → NomVal⟦2⟧, z1'[t] → NomVal⟦3⟧,
    θ1'[t] → NomVal⟦4⟧, ϕ1'[t] → NomVal⟦5⟧, φ1'[t] → NomVal⟦6⟧
    (*,xb1'[t]→xb_o⟦1⟧,xb2'[t]→xb_o⟦2⟧,xb3'[t]→xb_o⟦3⟧,
     xb4'[t]→xb_o⟦4⟧,xb5'[t]→xb_o⟦5⟧,xb6'[t]→xb_o⟦6⟧*)} /.
  {√L² → L, x1''[t] → x1'[t], y1''[t] → y1'[t], z1''[t] → z1'[t],
   θ1''[t] → θ1'[t], ϕ1''[t] → ϕ1'[t], φ1''[t] → φ1'[t](*,
    xb1''[t]→xb1'[t],xb2''[t]→xb2'[t],xb3''[t]→xb3'[t],
    xb4''[t]→xb4'[t],xb5''[t]→xb5'[t],xb6''[t]→xb6'[t]*)}
```

■ 9. External input force and moment (disabled)

■ 10. The Lagrangian equations of motion

■ 10.1 Calculation of equations of motion.

```
EOM1 = Array[temp5, n⟦1⟧];
For[i = 1, i ≤ n⟦1⟧, i++, temp5[i] =
  ∂t (∂GenCoordD[i] Tt) - ∂GenCoord[i] Tt + ∂GenCoordD[i] Damp1 + ∂GenCoord[i] U3 - FM[i]];

Dimensions[EOM1]
```

■ 10.2 View the equations of motion.

... sas/bb : Massaging the first equation

... sas/bb : Massaging the second equation

... sas/bb : Massaging the third equation

... sas/bb : Massaging the fourth equation

... sas/bb : Massaging the fifth equation

... sas/bb : Massaging the sixth equation

■ 10.2.1 Test for some simple cases.

```
EOM1[[1]] /. {yl$_o$ → L, φl$_o$ → 0}

Simplify[EOM1[[2]] /. {yl$_o$ → L, φl$_o$ → 0}]

Simplify[EOM1[[6]] /. {yl$_o$ → L, φl$_o$ → 0}]
```

■ 11. State Space formulation

■ 11.0 Inspect the equations (disabled)

■ 11.1 Define equations for the State Space model (sas).

```
EOM1[[1]]

auxEq = Array[temp6, n[[1]]]; For[i = 1, i ≤ n[[1]], i++,
 temp6[i] = Part[Solve[EOM1[[i]] == 0, GenCoordDD[[i]]], 1]]
```

■ 11.2 Define components of the State Space model.

■ 11.2.1 State Space vector.

```
Symbolize[xSS]; Symbolize[xSS1]; Symbolize[xSS2]; Symbolize[xSS3];
Symbolize[xSS4]; Symbolize[xSS5]; Symbolize[xSS6]; Symbolize[xSS7];
Symbolize[xSS8]; Symbolize[xSS9]; Symbolize[xSS10]; Symbolize[xSS11];
Symbolize[xSS12]; Symbolize[xSS13]; Symbolize[xSS14];
Symbolize[xSS15]; Symbolize[xSS16]; Symbolize[xSS17];
Symbolize[xSS18]; Symbolize[xSS19]; Symbolize[xSS20];
Symbolize[xSS21]; Symbolize[xSS22]; Symbolize[xSS23]; Symbolize[xSS24];

xSS = {xSS1[t], xSS2[t], xSS3[t], xSS4[t],
  xSS5[t], xSS6[t], xSS7[t], xSS8[t], xSS9[t], xSS10[t], xSS11[t],
  xSS12[t](*,xSS13[t],xSS14[t],xSS15[t],xSS16[t],xSS17[t],xSS18[t],
   xSS19[t],xSS20[t],xSS21[t],xSS22[t],xSS23[t],xSS24[t]*)};
```

The state variables correspond to : xl[t], yl[t], zl[t], θl[t], φl[t], φl[t], xbl[t], xb2[t], xb3[t], xb4[t], xb5[t], xb6[t], xl'[t], yl'[t], zl'[t], θl'[t], φl'[t], φl'[t], xbl'[t], xb2'[t], xb3'[t], xb4'[t], xb5'[t], xb6'[t].

■ 11.2.2 Vector of input signals (forces acting on the platform and on the legs).

```
Symbolize[uSS];

uSS = {FM[1], FM[2], FM[3], FM[4], FM[5],
  FM[6](*,FM[7],FM[8],FM[9],FM[10],FM[11],FM[12]*)};
```

- 11.2.3 Vector of output signals (position of platform and legs).

```
Symbolize[ySS];

ySS = {xSS1[t], xSS2[t], xSS3[t], xSS4[t], xSS5[t],
  xSS6[t](*, xSS7[t], xSS8[t], xSS9[t], xSS10[t], xSS11[t], xSS12[t]*)};
```

■ 11.3 Construct the State Model.

- 11.3.1 Substitute state variables (from the State Space vector) into formulas derived in 11.1.

```
(*auxEqss = auxEq/.{x1[t]→xSS1[t], y1[t]→xSS2[t], z1[t]→xSS3[t],
    θ1[t]→xSS4[t], ϕ1[t]→xSS5[t], φ1[t]→xSS6[t], xb1[t]→xSS7[t],
    xb2[t]→xSS8[t], xb3[t]→xSS9[t], xb4[t]→xSS10[t], xb5[t]→xSS11[t],
    xb6[t]→xSS12[t], x1'[t]→xSS13[t], y1'[t]→xSS14[t],
    z1'[t]→xSS15[t], θ1'[t]→xSS16[t], ϕ1'[t]→xSS17[t], φ1'[t]→xSS18[t],
    xb1'[t]→xSS19[t], xb2'[t]→xSS20[t], xb3'[t]→xSS21[t],
    xb4'[t]→xSS22[t], xb5'[t]→xSS23[t], xb6'[t]→xSS24[t]};*)
auxEqss = auxEq /. {x1[t] → xSS1[t], y1[t] → xSS2[t], z1[t] → xSS3[t],
   θ1[t] → xSS4[t], ϕ1[t] → xSS5[t], φ1[t] → xSS6[t], x1'[t] → xSS7[t],
   y1'[t] → xSS8[t], z1'[t] → xSS9[t], θ1'[t] → xSS10[t],
   ϕ1'[t] → xSS11[t], φ1'[t] → xSS12[t]};
```

- 11.3.2 Build the function describing the 'State part' of the model and the 'Output part' of the model. These functions, linear in this case, appear in the following equations:

$$x' = f_{SS}(x, u)$$
$$y = h_{SS}(x, u)$$

```
Symbolize[fSS]; Symbolize[hSS];
```

```
(*fss={xss13[t],xss14[t],xss15[t],
   xss16[t],xss17[t],xss18[t],xss19[t],xss20[t],xss21[t],xss22[t],
   xss23[t],xss24[t],GenCoordDD[[1]] DOF[[1]]/.auxEqss[[1]],
   GenCoordDD[[2]] DOF[[2]]/.auxEqss[[2]],
   GenCoordDD[[3]] DOF[[3]]/.auxEqss[[3]],
   GenCoordDD[[4]] DOF[[4]]/.auxEqss[[4]],
   GenCoordDD[[5]] DOF[[5]]/.auxEqss[[5]],
   GenCoordDD[[6]] DOF[[6]]/.auxEqss[[6]],
   GenCoordDD[[7]] DOF[[7]]/.auxEqss[[7]],
   GenCoordDD[[8]] DOF[[8]]/.auxEqss[[8]],
   GenCoordDD[[9]] DOF[[9]]/.auxEqss[[9]],
   GenCoordDD[[10]] DOF[[10]]/.auxEqss[[10]],
   GenCoordDD[[11]] DOF[[11]]/.auxEqss[[11]],
   GenCoordDD[[12]] DOF[[12]]/.auxEqss[[12]]}*)
fss = {xss7[t], xss8[t], xss9[t], xss10[t],
  xss11[t], xss12[t], GenCoordDD[[1]] DOF[[1]] /. auxEqss[[1]],
  GenCoordDD[[2]] DOF[[2]] /. auxEqss[[2]],
  GenCoordDD[[3]] DOF[[3]] /. auxEqss[[3]],
  GenCoordDD[[4]] DOF[[4]] /. auxEqss[[4]],
  GenCoordDD[[5]] DOF[[5]] /. auxEqss[[5]],
  GenCoordDD[[6]] DOF[[6]] /. auxEqss[[6]]}

Dimensions[fss]

hss = yss
```

- 11.3.3 Convert the above nonlinear (or linear as in this case) State Space form to the 'standard' linear form.

$$x' = A\,x + B\,u$$
$$y = C\,x + J\,u$$

```
systemSS = Linearize[fss, hss, {{xss1[t], 0}, {xss2[t], 0}, {xss3[t], 0},
   {xss4[t], 0}, {xss5[t], 0}, {xss6[t], 0}, {xss7[t], 0}, {xss8[t], 0},
   {xss9[t], 0}, {xss10[t], 0}, {xss11[t], 0}, {xss12[t], 0}(*,
    {xss13[t],0},{xss14[t],0},{xss15[t],0},{xss16[t],0},{xss17[t],0},
    {xss18[t],0},{xss19[t],0},{xss20[t],0},{xss21[t],0},
    {xss22[t],0},{xss23[t],0},{xss24[t],0}*)}, {{FM[1], 0},
   {FM[2], 0}, {FM[3], 0}, {FM[4], 0}, {FM[5], 0}, {FM[6], 0}}];

systemSS
```

■ 11.4 Test at various configurations.

- 11.4.1 First, test the observability (symbolically -- it takes about 12 hours--disabled)

```
Observable[systemSS]
```

11.4.2 Test at simple configuration

Substitute in numerical values for the parameters.

```
numSS = systemSS /. {(*mL→1,*)mP → 10.84, K_LS → 500000, B_LS → 0.001,
   (*K_SS→1000,B_SS→0.1,*) JP_xx → 0.024, JP_yy → 0.035, JP_zz → 0.043};
```

Obtain the eigenfrequencies (Hz).

```
MatrixForm[ Abs[Eigenvalues[numSS[[1]]]] / (2 π) ]

ReviewForm[numSS];
```

Obtain transfer function of the system from the State Space model.

```
TF = TransferFunction[s, numSS];

Dimensions[TF[[2]]]
```

Obtain the TF corresponding to x1 input and x1 output, x1 input and y1 output...

```
MatrixForm[Chop[TF[[2]] /. {s -> 1}]]

aux1 = Simplify[Chop[TF[[2]][[1]][[1]]]
aux2 = Simplify[Chop[TF[[2]][[3]][[3]]]
```

Get elements of the TF.

```
(*res=FactorRational[TransferFunction[s,aux]]*)

(*ZeroPoleGain[TransferFunction[s,aux]]*)
```

Plot Bode plot.

```
BodePlot[TransferFunction[s, aux1], PlotPoints -> 1000];
BodePlot[TransferFunction[s, aux2], PlotPoints -> 1000];
```

Test static compliance, numerically (using Final Value Theorem).

```
Limit[aux1, s -> 0]
Limit[aux2, s -> 0]
```