# Flexible Multibody Dynamics: A New Approach Using Virtual Work and Graph Theory

by

Pengfei Shi

A thesis

presented to the University of Waterloo

in fulfilment of the

thesis requirement for the degree of

Doctor of Philosophy

in

Systems Design Engineering

Waterloo, Ontario, Canada, 1998

The University of Waterloo requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

# Abstract

A new approach to flexible multibody dynamics is presented. Its most prominent feature is that it extends the existing graph-theoretic (GT) method for multibody dynamics to include flexible bodies. This is accomplished by extending the traditional form of system graph and by using the novel idea of adopting virtual work as a through variable. The validity of virtual work (VW) as a through variable is demonstrated philosophically, mathematically, and with general examples, for the graph-theoretic models of elements presented in the thesis. An additional advantage of the new approach is that it can reduce the number of system equations as compared with conventional absolute or joint coordinate formulations for multibody systems with closed loops. The new VW graph-theoretic approach encompasses most existing graph-theoretic approaches to multibody dynamics.

New GT elements are created. They include the flexible body element, the flexible arm element, and the dependent VW element. Terminal equations for conventional multibody elements (rigid bodies, joints, forces) are derived in terms of VW. Construction of a system graph is explained and demonstrated with examples. In addition to the VW through variable, the conventional across and through variables for each element in the system all satisfy the topological cutset and circuit equations from the system graph.

A systematic procedure for formulating system equations, including kinetic and kinematic constraint equations, is put forward that preserves the methodical nature of the traditional GT method. A symbolic-numeric computer package (DynaFlex) is developed for a formulation in which joints are selected into the tree of the system graph.

The three-dimensional kinematics of a Bernoulli-Euler beam is revisited so that a suitable model is developed for it to be included in the new graph-theoretic approach. It is found that using a commonly-used first-order deformation field causes some first-order inertial force terms to be missed from system equations. A new remedy of using a complete second-order deformation field is proposed, and a methodical approach to

generating a deformation field that is complete up to any order is given. The use of the proposed complete second-order deformation field is validated in Chapter 7.

Various other issues relating to symbolic implementation of the new approach, Rayleigh-Ritz discretization of the deformation variables of a Bernoulli-Euler beam, numerical solution of differential-and-algebraic equations, and future research directions are discussed.

## Nomenclature

| | |
|---|---|
| $a$, $A$ | scalar |
| $\mathcal{R}$ | rotation transformation matrix |
| $\underline{a}$ | three-dimensional vector |
| $\hat{i}$, $\hat{j}$, $\hat{k}$ | unit vectors |
| $\{\underline{e}^i\}$ | column of unit vectors (vectrix) for reference frame $X^iY^iZ^i$ |
| $\mathbf{q}$, $\mathbf{A}$ | matrix, including column matrix |
| $\tilde{A}$ | skew-symmetric matrix of column matrix $\mathbf{A}$ |
| $\delta$ | the operator of variation |
| $\delta W$ | virtual work |
| $\dot{(\,)}$ | time derivative with respect to the inertial frame |
| $\overset{\circ}{(\,)}$ | time derivative with respect to a body reference frame |
| $\{\,\}$ | column matrix with entries displayed |
| $[\,]$ | non-column matrix with entries displayed |
| $\triangleq$ | given as, defined as |
| $O(n)$ | to the $n$-th order |
| $en$ | a graph edge numbered as $n$ |

# Acknowledgments

First, I would like to thank my supervisors Dr. John McPhee and Dr. Glenn Heppler, for granting me such a good opportunity to come to the University of Waterloo to further my academic growth, and for their valuable comments, suggestions, and advice that have made my research work here a rewarding one.

I would also like to thank Mr. Christian Lange of University of Graz, Germany, for his initial MOBILE source code for the Rigid Spatial Slider Crank Mechanism.

I would like to thank my colleague Ms. Lisa Li for her encouragement and assistance in everyday life that pushed me through the initial stages of my living and study here.

I would like to thank my parents who, with due love, have never ceased inspiring me for constant pursuit of knowledge.

Finally, I would like to thank my wife and son for their bearing with me spending so little time with them during my doctorate program.

*'Learning without thought is labour lost;*

*thought without learning is perilous.'*

*Confucius, 551–479 B.C.*

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

A multibody system is defined as a set of material bodies, rigid or flexible, interconnected with mechanical joints so that they can move relative to each other in a certain fashion. In general, such a system is very complex and difficult to analyze by hand. A main goal of multibody dynamics is to study and produce general-purpose algorithms that can be automated on a computer to systematically generate and solve system equations so as to predict the motion of the bodies involved and the forces (torques) required to effect certain motions of the system.

Multibody systems are found in many areas of applications: robotics, mechanisms, and biosystems. To design and control them, high demands have been made on the continued improvement in the equation formulation and the simulation fidelity of such systems, especially when light-weight components that move at high speeds are involved.

Since large rotations are inherently involved in general multibody systems, their dynamic equations are highly nonlinear in nature, and their analysis is one of the most difficult problems in modern mechanics [1].

Flexible multibody systems are distinguished from their rigid body counterparts by their flexibility, and from structures by the gross motion of the bodies that comprise the

systems. The flexibility and the gross motion exhibited in flexible multibody systems give rise to various difficulties in modelling and simulation. The key issues in the study of such systems are the proper modelling of the strain-causing deformation, and a fast and reliable solution of system equations, either symbolically or numerically. In spite of numerous studies that have been done and the extensive research efforts devoted, there are still many unresolved issues.

The study of flexible multibody dynamics has been around for some time. There are, however, still disagreements between dynamicists about the best dynamic laws to employ to generate system dynamic equations. There are disagreements about the consistency of different approximation methods. There are disagreements about which dynamic formulation procedures are the most effective and most efficient. There are even disagreements about the interpretation and application of the results of the analyses.

In spite of the disagreements, all approaches to flexible body systems still follow a basic framework that underlies the study. First, one assumes how the body would deform under certain loading cases. This assumption, or idealization, of a deformation pattern is in effect a physical discretization that turns a problem with an infinite number of degrees of freedom into that with only a finite number. Following the idealization of the deformation pattern comes the generation of system equations, for which there are the traditional pure mechanics approach and the less traditional graph-theoretic approach. The solution of system equations, which produces the system behavior in some cases, completes the analysis of a multibody system for most cases.

To physically discretize beams, they are categorized into Bernoulli-Euler beams, Timoshenko beams, and possibly other beam types as requirements arise. With their respective assumed deformation pattern, the configuration of a Timoshenko beam can be specified by six degrees of freedom and that of a Bernoulli-Euler beam by four only, instead of an infinite number. Although it is a common sense that the closer the assumed pattern of

deformation to the actual deformation, the more accurate the predicted behavior of the system can be, it has not been easy to understand what is a good suitable mathematical representation for a given physical deformation field (one stated in words), not to mention how to devise a more accurate mathematical approximation to a deformation field, as can be seen later in the thesis.

Various dynamics laws have been used to generate equations in either differential or variational form for multibody systems. It is especially tempting to employ the Principle of Virtual Work or its like, such as Jourdain's Principle or Gaussian Principle because they can eliminate joint reactions even for closed loop systems when the system is small enough that can be handled with a pen-and-paper. Unfortunately, this advantage that works by hand has not been fully explored in computer-aided analysis of multibody systems till now when the successful marriage of graph theory with Principle of Virtual Work has made it reality.

The graph-theoretic (GT) approach with force and torque as the through variable has been developing since early seventies [60, 61], and they are good enough for dealing with rigid body systems today. This approach is noted for its clear concepts, well-established theory, systematic procedure for formulating system equations and potential to deal with complex systems that may involve electrical, thermal, hydraulic and other types of components. Its systematic procedure for generating system equations is especially appealing in dealing with multibody dynamics. Thus, in seeking to preserve this prominent feature of GT approach and to extend it to systems involving flexible bodies, a novel GT approach that makes use of virtual work as the through variable was born and is presented in this thesis.

In some cases of engineering importance, merely predicting the system behavior is not enough. Control of systems, or sensitivity analysis of system behavior to some parameters, is often required. In such a case, generation of symbolic system equations is desirable

for it can speed up the numerical solution of the equations, leaving more time for more advanced control schemes; it can also provide an easier way to compute the gradients needed for an up-coming sensitivity study. In addition, this approach of symbolically generating system equations and solving them numerically is the closest to that a human researcher would follow if he could solve a problem by hand; it produces special-purpose simulation codes that work more efficiently than those used in a general-purpose pure numerical package [4].

Numerical solution of system dynamic equations, which include kinetic and constraint kinematic equations, and usually come in the form of second-order ordinary differential equations coupled with algebraic equations (DAEs), is no easy task at all, when there are still difficulties even solving some ordinary differential equations in practice. In spite of this, different strategies were proposed in the literature to reduce or convert the DAEs into pure ordinary differential equations (ODEs) and then solve them as such, and they have met with varying degrees of success. Unfortunately, these traditional approaches to solving DAEs are now called into question [2] as regard to their ability to preserve the numerical properties of the original DAEs, and computational mathematicians are still struggling to find feasible and reliable ways for solving DAEs [3] directly.

There are eight chapters in this thesis. In the next chapter, a brief literature review is presented on the issues mentioned above, and the contributions I have made to multibody dynamics and graph-theoretic methods to it is outlined. In Chapter 3, a philosophical justification is produced for using virtual work as a through variable in my new GT approach, and related issues are discussed. Then new GT models of elements commonly encountered in modelling multibody systems, including flexible bodies, are presented. New elements, such as flexible body elements, flexible arm elements and dependent virtual work elements, are discussed. A mathematical proof that virtual work indeed satisfies the vertex postulate for the new elemental graph-theoretic models is given.

Following that, in Chapter 4, the general terminal equations for flexible bodies presented in Chapter 3 are tailored to a Bernoulli-Euler beam. An in-depth look into the traditional assumed displacement field for the beam is conducted, a conclusion is drawn as to its suitability for being used to obtain a second order elastic rotation matrix, and a new complete second order elastic rotation matrix and thus displacement field is proposed. This new displacement field is then employed to evaluate the beam's terminal equations, including the virtual work terminal equation. The effect of using the proposed complete second order displacement on the beam's inertial and elastic forces is discussed in detail. In Chapter 5, a general procedure is presented for formulating system equations with the new graph-theoretic approach. How to construct a system graph and what types of formulation trees that can be selected are discussed in detail. The chapter ends with a few simple examples that serve to illustrate the proposed approach and formulation procedure. Material presented up to this point is complete on the new GT approach, but to handle large and complex problems, this approach is automated in a combined symbolic-numerical computer package named DynaFlex, with the Maple computation package [110] serving as the programming language for generating symbolic system equations and Fortran for numerically solving them. In Chapter 6, DynaFlex is explained in some detail. It is capable of automatically generating motion equations for general three-dimensional flexible multibody systems, given only a description of the system as input, and generating a numerical approximation to the solution of these equations. Specifically the formulation procedure and related issues in its computer implementation with Maple are examined, and the numerical methods that can be employed to solve these symbolic equations are introduced. In Chapter 7, examples that have been solved with the computer package DynaFlex are presented. These examples include both rigid and flexible three-dimensional systems with closed loops. Results obtained are all compared with those recorded in the literature or generated from established multibody computer

packages. Experiments with the symbolic generation of motion equations and their numerical solutions are conducted and some observations are made. The thesis concludes with Chapter 8, a closure, where a few conclusions are drawn from the project done so far and related areas for future research are identified.

# Chapter 2

# Literature Review

This chapter contains a survey of the mainstream developments in the research on modelling and simulation of flexible multibody systems. Many of them are also involved in rigid multibody systems.

Two major approaches are used in the research: the conventional pure mechanics approach, which makes direct use of the diverse laws and principles under the framework of rigid body dynamics, vibration theory and continuum mechanics to derive the system equations; the other is the so-called vector-network (VN) approach or graph-theoretic (GT) approach, which takes basic concepts from linear graph theory and blends them with fundamental principles of mechanics, resulting in a systematic and methodical formulation of the system equations of motion.

Due to the complexity involved in the formulation of system equations for flexible bodies, the derivation of system equations is cumbersome and error-prone. On the other hand, computers have developed to the point where they can be asked to perform the tedious labor previously done by hand. As a result, the computer-aided approach to equation generation was born, and has been gaining momentum in recent years. The background research survey will also provide a glimpse at the development of this rela-

tively new trend, commonly known as multibody dynamics.

Numerical solution of dynamic equations of multibody systems is another important area of research that is attracting more and more attention of even computational mathematicians. The approaches employed by different schools of researchers to deal with the DAEs that govern the motion of multibody systems will be outlined.

## 2.1 Conventional Approaches in Flexible Multibody Dynamics

Two basic methods are currently in use to describe the configuration of an elastic component in flexible multibody dynamics: the floating frame method and the inertial frame method. Each gives rise to a different group of formulations, resulting in different forms of dynamic equations for the same multibody system under study.

In the floating frame method, which is the basis of most computer packages, the component configuration is defined using a floating coordinate system superimposed with another set of coordinates of the component relative to this floating system. Shadow beams [20] and the body mean axes [21] are two commonly used floating frames. With the assumption of small strains, the use of a shadow beam allows a simple expression for the total potential energy of the component, but the expression for the kinetic energy takes a rather cumbersome form. The resulting equations of motion are nonlinear and highly coupled in the inertia terms due to the presence of Coriolis and centrifugal effects as well as inertia due to the rotation of the shadow beam. In contrast, body mean axes permit the decoupling of translational rigid body motion from the rotational rigid body motion and the elastic deformation [21, 55]. Regardless of which choice of floating frame, the deformation of a component is discretized either using the finite element method or the component mode method in order to render the final dynamic equations of motion

tractable.

The inertial frame method describes the elastic component configuration directly in terms of the coordinates of the component referred to the inertial frame, but still it is imperative to extract the deformation of the component from this inertial set of coordinates so that the internal elastic forces or the strain energy of the component can be calculated, which is needed to derive the kinetic equations of the component. The resulting equations of motion from the inertial frame description have a constant mass matrix while the stiffness operator emanating from the potential energy functional becomes nonlinear [20, 52].

### 2.1.1 The Shadow Beam Frame Description

#### Finite Element Discretization

Winfrey[23], Erdman[24] and Imam[25] were among the first to apply finite element methods to flexible mechanism problems, but their works were of an elastodynamic nature, uncoupling artificially rigid body motion from elastic deformation.

Song and Haug[22] were the first to treat each member in a planar mechanism as being flexible and combine kinematic constraint equations with kinetic equations to obtain a coupled system of equations that precisely represent all planar degrees of freedom, an approach that is still in use today for flexible multibody dynamics analysis. Shadow beam frames and the finite element method were used in representing the flexibility of the members. Lagrange's equations were employed to derive the kinetic equations.

The set of three papers by Turcic et al. [26, 27, 28] marked the first systematic efforts to investigate the dynamics of flexible multibody systems both analytically and experimentally. In [26], a mathematical model using Lagrange's equations and quadratic finite beam elements for a general two-dimensional complex elastic mechanism were presented, and the model could be extended to the three-dimensional case. In [27], the equations

derived in [26], with the addition of a what is today called geometric stiffening matrix obtained from a quasi-static axial force analysis, were applied to two practical examples, including a four-bar system with three flexible links. The authors found that it is important to include the vibration effects in the analysis of high-speed mechanisms and that much lower stresses and strains would have resulted if the quasi-static axial force analysis had not been used. Experiments were conducted with the flexible four-bar system and results were shown in[28]. It was demonstrated that the experimental and analytical strains exhibited good agreement, and the inclusion of the geometric stiffening matrix in the theoretic model considerably improved its comparison with the experimental results.

Sung, Thompson, Xing, and Wang[38] conducted a detailed experimental study of slider-crank mechanisms and four-bar linkages with various geometric dimensions. Their experimental data compared favorably with the results of a computer simulation that used a finite element model which included elastic geometric nonlinearities. The agreement provided a high degree of confidence in the mathematical modelling procedure.

Bakr and Shabana[29] introduced a two-dimensional Timoshenko beam element into flexible multibody dynamics for the first time, accounting for the rotatory inertia and shear deformation effects at the same time.

Sincarsin and Hughes[32, 33], using Newton-Euler dynamics mated with the finite element method, modelled the dynamics of flexible multibody systems with open loops, producing system equations that are amenable to the introduction of constraint and control forces but are valid only for low speed motion due to the exclusion of geometric nonlinearities. A coherent symbology for multibody dynamics analysis was developed in this paper as well.

Han and Zhao[30] presented a general treatment for modelling the dynamics of a flexible multibody system, using a lumped mass finite element approach that is capable of treating an arbitrary combination of both rigid and flexible bodies connected together

by joints that permit translations and rotations, in a general tree configuration, with an extension to handle closed loops indicated.

### Component Mode Discretization

Hurty, who is considered the pioneer in this field, developed the Component Mode Synthesis technique[34, 35] for structural dynamics. This approach approximates the deformation of a flexible body with a linear combination of a finite number of generalized elastic coordinates each multiplied by a mode shape, which may include the vibrational modes, static modes and other modes of the body. The Craig-Bampton method differs only slightly from Hurty's and the two give the same numerical results[36]. Gladwell also developed a variant of Hurty's method which has come to be called the Branch Mode Analysis [37]. In general, these methods tend to differ only in the selection of the various component modes.

Haug, Wu and Kim[39] used a variational approach to derive dynamic equations of flexible machines, modelling the elastic component deformation with static and vibrational modes. The vibrational modes were taken from a structural analysis of the component, which was carried out using the finite element method and the linear strain-displacement relations with a certain specified set of boundary conditions.

Ryan[40] studied the conventional modal modelling approach and concluded that unless the interdependence of displacements is accounted for, this approach is hazardous in general, although it works for some special cases. In [40], Ryan recommended retaining fully nonlinear generalized elastic forces or using discrete rigid sub-bodies (capable of large rotation and translation) to avoid these problems.

Using Lagrange's equations, Shabana[41] compared the component mode approach with the finite element method, with both of the final two sets of dynamic equations expressed in terms of invariants that depend only on the assumed displacement field. While

the theoretical development shows that the use of the finite element method significantly reduces the number of independent invariants of motion, numerical results show that they agree well with each other

Xia and Menq[42] proposed a new component mode method in which the system modes, instead of the modes of its individual components, are found as functions of system configurations, using an interpolation technique. These system modes account for the effects of both the coupling between the rigid body motions and the coupling among deformations of different components. A flexible two-link manipulator was used to illustrate the proposed approach.

A detailed examination was conducted by Shin and Yoo [68] on the effects of mode selection, scaling, and orthogonalization. In their conclusion, Shin and Yoo recommended the selection of vibrational normal modes and attachment modes along with mode scaling because the combination was found to enhance numerical efficiency.

DADS [44] and ADAMS [45], two widely-used multibody computer packages that use a purely numerical approach, were developed along the shadow beam approach. They both require vibrational modes of the flexible body that result from a finite element analysis of the body.

### 2.1.2 Body Mean Axes

This type of floating frame was considered to be first introduced by McDonough[21]. The frame is defined such that the body possesses zero linear and angular momenta as seen from the frame. This definition implies that the kinetic energy of the body, as observed from the frame, is minimized. McDonough derived the dynamic equations of a flexible body using classical Newton-Euler equations, and transformed them to the body mean axes, producing equations that are partially decoupled between the rigid body motion and the displacements due to deformation. No illustrative examples were given.

Cavin[53] elaborated along this same line, using Hamilton's principle combined with a finite element discretization scheme to turn the equations into a computationally feasible form. Unfortunately again, no example problems were actually solved.

Agrawal and Shabana[54] explored and implemented the body mean axis description in conjunction with a finite element discretization. They solved three problems numerically, including a spatial one.

Koppens, Sauren, Veldpaus, and Campen [55], using a planar beam, showed that the body mean axis yields simpler final dynamic equations as compared with the shadow beam frame description. Pure mathematical modes were used in his elaboration to discretize the displacements of the flexible component relative to the floating frames.

So far, no known computer package has been created based on this approach.

### 2.1.3 The Inertial Frame Description

A co-rotational formulation of the finite element method was first introduced into transient dynamic analysis of large translational and rotational displacement problems by Belytschko and Hsieh[43], who used a planar, constant triangular element and an Euler-Bernoulli beam element with material nonlinearities. Later, Belytschko and Schwer[47] extended this formulation into treating spatial frames in large displacement, small strain problems, with node orientations described by unit vectors.

In a set of two papers, Simo and Vu-Quoc[48, 49] introduced this finite element method into flexible multibody dynamics. With all the quantities referred to the inertial frame, the resulting set of kinetic equations are linear in time derivative terms and nonlinear only in the stiffness part of the equations. In addition, large strains are automatically included in the model. Planar example problems were solved using computer codes designed typically for nonlinear structural dynamics.

Geradin, Cardona, and Granville[50] used the rotation vector to describe node rota-

tions in three-dimensional flexible multibody problems. Later Cardona [51] introduced the component synthesis technique to create superelements for modelling mechanism members. Based on their work, this team developed MECANO, a computer program for the analysis of flexible mechanical systems.

Proceeding in a mathematically less complicated approach in modelling three-dimensional rotation, Avello and Garcia de Jalon[52] used unit vectors to specify nodal orientations, and developed a three-dimensional nonlinear Timoshenko beam finite element capable of handling finite displacements with small linear elastic strain. Constraint equations on the unit vectors were adjoined to the kinematic and kinetic equations of the system.

In [20], [50] and [52], the rotation vectors are regarded as field variables and discretized in the same way as nodal displacements are. As a result, the intrapolated rotation in an element might not agree with the rotation that is calculated from the intrapolated displacement field, leading to singularity problems when slender structures are considered.

### 2.1.4 Modelling Geometric Stiffening

For a rotating elastic beam, the equations of motion obtained by the linear theory of elasticity predict the beam to become less resistant to bending, which is in contradiction with experimental observations. This inconsistency prompted researchers to look for the shortcomings in the existing dynamic modelling method and to find a way to account for the missing stiffness.

This is not a new issue. As early as the mid-seventies, Vigneron [8] and Kaza [9] had already elaborated on this topic, but their statements were made largely for Bernoulli-Euler beams since they all used the traditionally employed deformation field approximation for Bernoulli-Euler beams.

In the past decade or so, interest in the modelling of the geometric stiffening effect

has been on the increase, especially after the well-known paper[10] was presented, demonstrating that existing multibody computer codes can produce incorrect results because of their neglect of dynamic stiffening. This issue arises only with the floating frame description, for with the inertial frame description this effect is automatically included in the formulation.

Some researchers [11, 12, 13] assert that in flexible multibody systems, the coupling between the elastic transverse and axial deformations becomes significant. Therefore to correct the loss of stiffness, some higher-order terms in generalized elastic coordinates, third order and even higher, in the strain energy expression must be retained. This gives rise to some nonlinear stiffness terms in the equations of motion. Worth special mention is the work by Hanagud and Sarkar because their approach is the most general and accurate, and can be extended to general three-dimensional rotating bodies [58].

Hurty[14] and Banerjee[15] consider the centrifugal field as an external pseudo-potential field and find the stiffness due to this field a linear one and add it to the stiffness obtained from the linear-elastic strain energy.

On the other hand, Padilla[16] believes that the missing stiffness is due to premature linearization, which refers to the linearization that starts prior to the calculation of partial derivatives in the process of deriving motion equations. For example, any linearization done prior to the calculation of partial velocities in Kane's method, or partial derivatives of kinetic and potential energy expressions in Lagrange's method, would be premature. Padilla suggests the use of nonlinear strain-displacement relations to prevent loss of any terms in the equations of motion.

Mayo[17] investigated the geometric stiffening, using the well-known slider-crank mechanism, compared the results to [18], and concluded that the effect of the nonlinear terms of the strain-displacement relations on transverse vibrations may be important when either significant axial forces or large transverse vibrations are present.

The latest paper by Sadigh and Misra[19] presented an analytical development based on Kane's method. They show that, in general, the equations of motion of any system with elastic vibrations and rigid body motions might have some terms missing if the elastic deformation variable is expressed as a linear combination of the generalized coordinates. Three existing remedies for this deficiency, namely using the nonlinear strain energy expression, a pseudo-potential field due to the inertia forces, and the nonlinear strain-displacement relations, are compared. The third is recommended as the most precise and straightforward, which coincides with the statement made in [58]. A method based on the nonlinear strain-displacement relations is presented which can be used either to generate the correct form of the partially linearized equations or to correctly compensate for the missing terms in a prematurely linearized set of equations. Illustrative examples are used extensively to demonstrate the above ideas.

Damaren and Sharf[56], in investigating the dynamics of the two-link Space Shuttle Remote Manipulator System, classified the sources of nonlinearities, and numerically tested the effects of the various types of nonlinearities on the robot system. Their results show that all terms are required for accurate simulation of fast maneuvers.

## 2.2 Formulations Using a Graph-Theoretic Approach

The graph-theoretic approach in dynamic analysis, otherwise known as the vector-network method, is a simple, systematic and versatile scheme for modelling dynamic systems using principles from linear graph theory as its framework and laws of mechanics as its foundation. Ever since its advent[60] more than twenty years ago, it has been developing steadily thanks to the persistent efforts by an increasing number of researchers. Now it is ready to attack the last bastion of mechanical system analysis: dynamics of flexible multibody systems.

### 2.2.1 Graph-Theoretic Approach for Rigid-Body Systems

The first systematic application of graph theory methods to dynamics resulted in the "vector-network method" of Andrews and Kesavan[60] [61]. This branch-chord formulation, with force and displacement as through and across variables respectively, was used to systematically derive the equations of motion, in terms of absolute coordinates, for a three-dimensional system of unconstrained particle masses. From this approach, the computer program VECNET was born[60].

Li[69] extended this work, using forces and moments as through variables and displacements and rotations as across variables, to the analysis of two-dimensional mechanical systems, by developing an efficient substitution procedure for formulating the governing differential and algebraic equations (DAEs). Ormrod[62] implemented Li's idea in what is known as ADVENT.

Along the same line, McPhee[65] developed TRIDENT to deal with three-dimensional unconstrained rigid bodies while Andrews, Richard and Anderson[63] pushed the method one step further to treat general two- and three-dimensional systems, producing the computer program RESTRI.

Lo[70] expanded vector-network theory by introducing work and change in energy, and linear impulse and change in momentum as through variables. This extension was made to facilitate the formulation of dynamic equations in the expert system named KINDY for particle dynamics.

McPhee[64] explored the different selection schemes for the spanning tree in the linear graph, and gave the criteria for selecting a tree that produces a set of motion equations in any desired set of coordinates. The GT approach was combined with the projection method [103] so that a smaller set of system equations can result. In [75], the same author contrasted the existing formulations that employ more or less graph-theoretic concepts

with the vector-network approach.

### 2.2.2 Graph-Theoretic Approach for Flexible Multibody Systems

Publications in this category are scarce so far.

Richard and Tennich[66] dealt with flexible systems, using forces and moments, and displacements and rotations as through and across variables, respectively. In establishing terminal equations for flexible components, the change in moments of inertia of the components was ignored, and the Euler equations for rigid bodies were used to model the rotational motion, accounting only for the change in moments of applied forces due to the deformation of the components. The concept of flexible-arm elements is introduced, but terminal equations of the flexible component are assumed to be the same as those for rigid arms except that the deformation displacement of the component containing the flexible arm is included, disregarding the relative and the Coriolis accelerations of the flexible arms. In addition, deformation of components is calculated using traditional structural dynamics, without considering the fact, which is essential to flexible multibody dynamics, that the components undergo large-scale translations and rotations and that rigid body motions are coupled with elastic deformations.

In a later work, Tennich[67] created a graph for a flexible system, again using force as through variable and displacement as across variable. It is acceptable to take kinetic equations of a single flexible body resulting from finite element analysis to be the terminal equations of the body. Yet, the term $\mathbf{F}_1$, "defined as the vector of total force and torque applied to a body and obtained using the cutset equation" in the general terminal equation (12) for a body, is clearly incorrect because forces acting at different locations on a body contribute differently to the body's dynamic equations of motion, and their contributions should be included through conversion matrices post-multiplied by the forces. This means that kinetic equations (29) and (30) can not be generated with any cutset. As a result,

for the example, only circuit equations are applied to generate kinematic equations and cutset equations are not employed. Thus, his method is an ad hoc combination of GT method and finite element analysis that precludes the use of any coordinates other than the absolute ones.

## 2.3 Symbolic Formulation of System Equations

Symbolic formulation of system equations has several advantages over the more traditional numerical approach. First of all, dynamic equations for a given system with the same topological property need only be generated once when the other parameters are changed for reason of seeking a better design, a better control scheme and so on. Secondly, symbolic equations provide ease for follow-up analysis on these equations, such as obtaining gradients of certain variables for sensitivity analysis. Furthermore, by generating the system equations in symbolic form first and then solving them numerically, this approach saves the time needed for setting up the system equations at each time step in the integration that a pure numerical approach requires. It also eliminates all the multiplications by 0's and 1's, and thus holds the promise of making real-time simulation for complicated systems a reality.

Examples of the first generation of symbolic packages are collected in [59]. Unfortunately, all of them were designed for rigid body systems only. In addition, not all of them were capable of dealing with systems with closed loops. The dynamic laws that were used in these packages range from Newton-Euler equations to D'Alembert Principle and to Jourdain's Principle for reason of manipulation efficiency.

Recent development in symbolic formulation has seen more active research in modelling of flexible body systems or so-called hybrid systems, systems that contain both rigid and flexible bodies. Representatives are by Lieh [86], Piedboeuf [87], and Valembois, Fisette and Samin [88].

Lieh, using the Principle of Virtual Work mated with the generalized coordinate partitioning technique, presented a general set of hand-derived symbolic equations that are ready to be expanded fully into symbolic system equations of minimal number. This approach was implemented in Maple. Three-dimensional Bernoulli-Euler beams are considered, but their angle of twist is ignored in the formulation. Besides, the analyst has to decide what vibration modes to select in discretizing the beams and to provide them if the modes are not readily available.

Employing the Principle of Virtual Power, a variant of Principle of Virtual Work, Piedboeuf developed a symbolic computer package called SYMOFROS, implemented also in Maple. This package was originally intended for manipulators only, but is currently being extended to handle systems with closed loops. Three-dimensional Bernoulli-Euler beams are considered. Elastic deformation variables are approximated over the entire domain of the beam. Different shape function types are available for this purpose, including Taylor series and splines. Foreshortening effects are included in the displacement field of the beam to compensate for the exclusion of axial deformation from the model. The curvature approach is adopted to compute the elastic rotation matrix that is complete up to the second order in the generalized elastic coordinates. This second-order elastic rotation matrix gives rise to a complete second-order deformation field approximation. The second-order deformation field is used for calculating inertial forces in the system kinetic equations, which produces terms complete up to first order. A first order deformation field adopted from the second order one is used for calculating the elastic forces, which produces terms complete to first order too.

Taking a quite different approach, Valembois, Fisette, and Samin [88] using the Newton-Euler equations, first established the partial differential equations for a general beam whose cross section has six degrees of freedom and then discretized these equations by means of the weighted residual method [90]. Due to the employment of the Newton-

Euler equations, the system kinetic equations are complete to the first order, even though only a first order elastic rotation matrix and deformation displacement field are used in the formulation. The deformation variables of the beam are approximated over the entire domain of the beam using Taylor series as shape functions. Using Taylor series or other types of series, such as Chebyshev and Legendre [89], to approximate deformation variables over the entire domain of a beam has the advantage that the shape functions do not change with the beam under different and varying boundary conditions because the displacement field thus approximated can be forced to satisfy them [91, 92, 93], as the system moves. This problem-independent approximation also does well when a system is such that obtaining sensible vibrational modes is impossible. The main drawback of this Taylor series approach is that as the number of monomials increases, condition of system matrices tends to get worse, imposing numerical difficulties, as reported in [88].

## 2.4 Numerical Methods for Multibody Dynamics

Except for very simple multibody systems, the system equations have to be solved numerically to determine system behavior. System equations fall into one of three forms in terms of their structure: (1) ordinary differential equations, (2) index 3 differential-algebraic equations, i.e., system equations that contain Lagrange multipliers or unknown reaction forces, or (3) index 2 differential-algebraic equations, i.e., system equations that are free of Lagrange multipliers or unknown reaction forces. The Lagrange Multiplier free (LMF) form of system equations was first obtained by Kamman and Huston [96].

Of the three forms, ODEs are the most thoroughly investigated and thus are the easiest to solve. This does not mean, however, that all ODEs resulting from multibody dynamics are easy to solve.

To solve the index 3 DAEs, different strategies have been used. The most popular one is to convert them into ODEs and solve them thereafter [6]. This approach works for

most cases. Even in this underlying ODE approach, there exist many different techniques. Comparative studies have been done on them, e.g., in [113].

Parallel to the underlying ODE approach, research on direct solution of the DAEs has been developing steadily, though more slowly. Two basic techniques have been collected and examined systematically in [3]. They are the Implicit Runge-Kutta (IRK) and the Backward Differentiation Formula (BDF). These two methods as presented in [3] discretize the original set of DAEs without adding extra sets of other equations. Although much effort has been devoted to research on these methods, currently they can only handle a few classes of DAEs well, generally only those whose index [3] is lower than three, and as a rule of thumb, the lower the index of the DAEs, the easier they can be solved. It is also true that there are public domain DAE solvers available, such as DASSL by Petzold and RADAU5 by Hairer and Wanner, but dynamicists seem still reluctant to make use of them for various reasons.

In [98, 97], the overdetermined approach was proposed, in which the original index 3 DAEs, after being converted into index 1 ones, are joined with one or two of the constraint equations at velocity and acceleration level. These equations are then discretized with a BDF method and the resulting algebraic equations solved with Newton iteration. Führer [2] further examined this proposal and designed a new iteration procedure that makes use of the special structure of the problem and improves the numerical properties. Since it solves more equations than the original DAEs, this method has not been greeted with much enthusiasm, not at least by the dynamicists.

On another front in an attempt to speed up the numerical solution of system equations so that real-time simulation can be achieved for large scale problems, parallel methods have been emerging [99, 100]. A discussion on the topic is outside the scope of this thesis.

## 2.5 Contributions of this Thesis

Two major contributions are made in this thesis. First, it presents a novel graph-theoretic approach. This approach makes use of virtual work [115] as a through variable for a new set of elemental graphs, follows basically the same systematic procedure as that of the existing GT approaches, and generates system equations, including kinetic and kinematic constraint equations, for rigid, flexible and hybrid systems. Thus, it is a truly general approach as far as multibody systems are concerned. Second, it presents a new complete second-order elastic rotation matrix for Bernoulli-Euler beams, along with a corresponding complete second-order deformation displacement field.

In employing virtual work as the through variable, the added advantage of eliminating joint reactions from system equations (and thus reducing their number) emerges when the zero-virtual-work property of ideal joints is explored and used in the formulation process.

Using the complete second-order elastic rotation matrix, instead of the traditional first-order approximation, helps to recover all the missing first-order terms in the inertia forces and all the second-order terms in elastic force for the beam. As will be seen later, in the example of the three-dimensional spin-up beam with an off-set mass, the recovery of the first-order inertia terms can be of crucial importance. Depending on whether the foreshortening effect is explicitly included in the displacement field, the recovered second-order elastic force terms can have substantial impact as well.

Other contributions include a detailed examination of the traditional deformation field approximation for Bernoulli-Euler beams, a Maple implementation of the proposed GT formulation computer package DynaFlex, that automatically generates symbolic system equations, a discussion of some of the abnormalities in Maple's functions, employment of Chebyshev and Legendre polynomials as shape functions for discretizing deformation variables of flexible bodies, comparison of different ODE and DAE solution techniques as

they are applied to multibody dynamics, and the discovery of certain anomalies in results from the DADS software package.

# Chapter 3

# Virtual Work and Graph-Theoretic Models of Components

## 3.1 Introduction

The graph-theoretic approach to modelling physical systems is simple, systematic and versatile, and the basic concepts of graph theory have permeated into various disciplines of engineering [72, 73, 74]. The vector-network technique, as linear graph theory applied to mechanical systems has come to be known, is a unifying concept, for it shows that systems theory concepts that apply to a wide variety of systems are equally applicable to the dynamic analysis of mechanical systems. Application of linear graph theory to the dynamics of particles and rigid-body systems has been a success, as can be seen from the literature review.

However, difficult problems arise when it comes to tackling systems that have flexible

components. So far the vector-network technique has been using force (and moment) as through variables and displacement (and rotation) as across variables. This association of graph-theoretic variables works well because of the rigidity assumption for the system components under study, and the resulting simplicity of the dynamic equations of motion. To be specific, force and moment summations are used directly to set up the dynamic equations using the Newton-Euler equations in existing GT approaches. This summation procedure is the essential feature in the successful application of the graph theory with forces and moments used as though variables. Furthermore, the D'Alembert form and the definiteness in the number of the dynamic equations makes it possible to design an associated isomorphic linear graph and to count the number of unknowns involved and the number of system equations available just by examining the structure of the system graph.

For systems having flexible components, however, the dynamic equations are far more complicated and their formulation far more involved. Some practices that are popular and efficient in modelling rigid body dynamics do not translate conveniently to flexible body dynamics. For example, forces and moments are rarely used directly to set up dynamic equations. Instead, concepts such as energy and work are often utilized. In other words, energy-based dynamic formulations are often favored. Furthermore, the number of dynamic equations at the beginning of the formulation may be different from the number of the dynamic equations one finally obtains, as in the case of using the Principle of Virtual Work or the Principle of Virtual Power. The final number of dynamic equations for flexible systems depends on the number of generalized coordinates used to model the small deformations of flexible components, which does not occur in rigid body dynamics, in addition to other coordinates that experience large motions.

On the other hand, from the viewpoint of the graph theory, for a true and appropriate linear graph to exist which represents a mechanical system, both kinetically and kine-

matically, it is necessary that the dimensions of the chosen through variables associated with each edge in the graph be the same so that it is possible that the vertex postulate be satisfied. The same requirement applies to across variables for a similar reason.

In view of these facts and based on the summative nature of the Principle of Virtual Work [101], it is proposed here that the quantity of virtual work [115] be employed as a through variable in a graph-theoretic formulation of the dynamic equations for flexible multibody systems, with displacements (rotations) still serving as across variables.

With virtual work as a through variable, the vertex postulate can be satisfied if the elements involved in the system are modelled properly, as is demonstrated later in this chapter.

## 3.2 Validity of Virtual Work as a Through Variable

Ever since Firestone [76] pointed out the imperfections of the old mechanical-electrical analogy and outlined a new analogy almost six decades ago, the popular notions of through and across variables have nearly always been associated with some sort of measuring instrument and their physical connection pattern relative to system components. For example, in an analysis of electrical systems, current is used as the through variable because, in addition to satisfying the vertex postulate (commonly known in electrical engineering as Kirchhoff's current law), it can be measured by an ammeter placed in series with system elements. Voltage drop is used as the across variable because, in addition to satisfying the circuit postulate (commonly known as Kirchhoff's voltage law), it can be measured by a voltmeter placed in parallel with system elements. For thermal systems, heat flow rate is the traditional through variable because, beside satisfying the vertex postulate corresponding to the thermal energy conservation, it can be measured by a calorimeter connected in series with system elements; temperature is used as the across variable for the reason that, beside satisfying the circuit postulate, it can be measured by

a thermometer connected in parallel with system elements. In hydraulic systems, much in the same manner, fluid flow rate is regarded as the through variable because, apart from meeting the vertex postulate corresponding to the conservation of matter, it can be measured by a flow meter placed in series with the system elements; pressure is used as the across variable because, apart from meeting the circuit postulate, it can be measured by a pressure meter placed in parallel with the system element. For mechanical systems, force is used as the through variable because force satisfies the vertex postulate and at the same time there also exists an instrument that, when connected in series with the system element, measures force; displacement is used as the across variable because displacement satisfies the circuit postulate and at the same time there also exists an instrument that, when connected in parallel to the system element, measures displacement. The notion of associating system variables with connection patterns of measuring instruments is important because a system graph is constructed by replacing each instrument, in either the network of instruments for measuring the through variable or the network of instruments for measuring the across variable in a 1-to-1 correspondence, by a line segment connected between its terminals and assigning a direction to the segment in accordance with the polarity of the instrument.

While this operational approach of the graph-theoretical method, first proposed by Trent in his pioneering paper [77], has helped the growth of the application of graph theory to physical systems modelling, it should be noted the qualities required of variables for them to be eligible for either through or across variables have never been fully investigated in the literature. Trent [77] did define a through variable as one that satisfies the incidence law (an equivalent of the vertex postulate) and across variable as that which satisfies the mesh law (an equivalent of the circuit postulate), but he lost no time to point out the required manner in which instruments are applied, at least conceptually, to measure these quantities. The influence of the operational approach has been so profound that

the necessary existence of the corresponding instruments has been regarded by many as an indispensable part of the definitions for both the through variable and the across variable.

Unfortunately, apart from the variables for which there exist measuring instruments, there are numerous other important quantities associated with various physical systems for which engineers have not yet designed the measuring instruments, leaving their status unclear. Virtual work is just one example of the quantities in the latter category. This situation has certainly encumbered to a great degree the further development of the graph-theoretical method of engineering problems. However, on the other hand, one observes that in the process of formulating system equations using the linear graph method, only three sets of equations are needed, namely the cutset equations, the circuit equations and the terminal equations, and the formulation process has nothing to do with the connection pattern of the measuring instruments once the system graph has been constructed.

This observation clearly reveals that the method is composed of two distinct steps: first constructing the system graph, and then generating the cutset and circuit equations and possibly manipulating them in conjunction with the terminal equations. As a result, as long as a system graph can be constructed such that the cutset and circuit equations obtained from it for any choice of tree conform to the physical laws that the chosen through variable and across variable must respectively satisfy, these chosen through and across variables are satisfactory choices no matter what their physical explanations are, and how (or even whether) they can be measured, practically or conceptually. To support the statement, let us take the dual graph of a mechanical system [79] as an example. In the dual graph, force is used as the across variable and displacement as the through variable, but there is no instrument available that connected in parallel measures force or connected in series measures displacement. The same statement is true of the dual graphs of other physical systems. Therefore it is only natural to define the through variable as a

variable that satisfies the vertex postulate and the across variable as that which satisfies the circuit postulate, regardless of their measurability.

It is important to emphasize at this point that only when the right terminal equations, the right system graph and the properly chosen through and across variables come together can a correct system model be guaranteed. For mechanical systems, when force is used as the through variable and displacement as the across variable, the vector-network is required to ensure that a correct system model is obtained. On the other hand, when force is used as the across variable and displacement as the through variable, the dual graph is required to ensure a correct system model is obtained, although in this case terminal equations are the same as those for the vector-network. The same holds for other engineering systems when the roles of traditional through and across variables are reversed.

Admittedly, it is no easy task and, even worse, there is no general rule to follow in constructing, for an arbitrarily chosen pair of through and across variables, a system graph that would generate, for any tree selection, the correct system equations by the use of cutset and circuit equations in conjunction with a proper set of terminal equations. Besides, there is the possibility that the isomorphism, usually exhibited in the operational approach, between the physical system and the correct system graph might be sacrificed.

The process of creating a correct system graph often involves some degree of art, or less pretentiously, contriving, and because of this some graph-theoretic analysts choose to shun mentioning this aspect of the method. Looking back on the history of the development of science, instances involving contriving or trial and error are not a rare occurrence. The most familiar example to dynamic analysts is the construction of Lyapunov functions for dynamic stability studies of engineering systems, and the involvement of some degree of trial and error in the method does not in the least make it any less important.

To be sure, with some choice of complementary variables, the operational approach

does help a great deal and provides a systematic way of constructing system graphs, as has been evidenced by the numerous engineering problems that have been solved using the traditional complementary variables. In spite of this, it must be pointed out that the operational approach is not totally applicable to mechanical systems. A closer look at Trent [77] would reveal that he also used art in constructing the system graph even for such a simple mechanical system as a mass under the action of two forces: he represented the D'Alembert force with an edge and there has not existed any serial instrument for measuring D'Alembert forces!

To conclude, in the graph-theoretic analysis of physical systems, any variable, regardless of its measurability, can be used as either a through variable or an across variable as long as a linear system graph along with the terminal equations can be created, in which the variable satisfies either the vertex postulate or the circuit postulate.

As a direct inference, for mechanical systems the quantity of virtual work can undoubtedly be used as a through variable provided a system graph can be created where the virtual work satisfies the vertex postulate corresponding to the Principle of Virtual Work. Interestingly enough, Koenig, Tokad and Kesavan [72] also mentioned that virtual work can be used as a through variable but the authors stopped short of any further elaboration.

With virtual work used as the through variable, the graph-theoretical models for common components of mechanical systems can be established and are presented later in this chapter. The system graph is simply the union of the subgraphs for each of the components that make up the system and the subgraphs are connected in the same way as the components are connected in the physical system.

## 3.3 Tellegen's Theorem with Virtual Work as Through Variable

Tellegen's theorem is well known to analysts in electrical engineering and to those who have experience with the graph-theoretic method of general engineering problems. It states that the summation of the products of the through and across variables associated with each edge in a graph equals zero [73]. This theorem holds as a result of the orthogonality between the circuit and the cutset matrix of the system graph, and independent of the across and through variables. This theorem, for some systems such as electric and hydraulic ones, amounts to the statement of energy conservation in the system. The energy interpretation, however, does not always hold. For example, in thermal systems heat flow which itself is a form of energy is used as the through variable and temperature is used as the across variable. It is obvious that the product of the through and across variable is no longer energy, let alone the energy interpretation. In spite of this, Tellegen's theorem is still true in this case, that is, the summation of the products of the heat flow and the temperature associated with each edge is equal to zero. Only the physical interpretation of such an equation is not obvious now. More examples of this nature can be found in non-physical systems, such as the transport system and the financial system, also known as flow problems in linear graph theory, where commodity flows [80] are used as the through variable. Therefore it is obvious that whatever quantity is used as the across variable, the equation formulated after Tellegen's theorem is not necessarily related to energy.

With virtual work chosen as the through variable for mechanical systems, Tellegen's Theorem can be shown to hold true as well, that is, the products of the virtual work and the displacement (either translational or rotational) associated with each edge sum to zero. As for thermal systems, though, there is no obvious energy interpretation for

such an equation, and other meaningful physical interpretations are not readily found. In addition, the product now takes a vectorial form rather than the traditional scalar form as seen in electric or mechanical systems when force and displacement are used as the complementary variables. These features, among others, have made the choice of virtual work as the through variable all the more complicated.

## 3.4 Graph-Theoretic Models of Common Elements

In applying the graph-theoretic method to a system, each element (also called component) is mathematically modelled separately, independent of other system elements, in the form of one or more terminal (or constitutive) equations. Each element is then graphically represented by one or more graph edges between certain vertices or points of interconnection of the element to other system elements. The result is a linear graph from which the equations describing the interconnection of the system elements can be obtained in a simple methodical way, using theorems from graph theory.

There are two sets of interconnection equations used in this study: the "fundamental-cutset" and the " fundamental-circuit" equations, usually called the cutset and circuit equations, respectively, for short. The terminal equations and graphs are so designed that the cutset equations are equivalent to the principle of virtual work as it is applied to flexible body dynamics; the circuit equations are equivalent to kinematic (displacement, velocity, acceleration) summations traditionally obtained from vector polygons. It must be emphasized here that each element in a system is represented by both the terminal equations and the graph edge(s) associated with it. In other words, the terminal equations and the graph edges together constitute the graph-theoretic (mathematical) model of the element, from which the system model is constructed using concepts of graph theory.

Elements commonly encountered in the modelling of mechanical systems can be classified into the following categories:

- $e1$: body elements, including rigid and flexible bodies (BE)
- $e2$: arm elements, including body arms and inertially-fixed arms (AE)
- $e3$: motion driver elements (MDE)
- $e4$: springs, dampers and actuators (SDA) between bodies
- $e5$: generalized force drivers (FDE), including applied forces and torques, acting either on joints or on a body from the inertial frame
- $e6$: joint elements (JE)
- $e7$: dependent virtual work elements (VWE)

For ease of presentation in this chapter, the category an element belongs to is identified by the first subscript number of the letter $e$. When there are more than one element of the same type appearing in a graph, a second subscript number will be added to distinguish them. All elements are defined by their terminal equations and graphs. In the following subsections, the new graph-theoretic models of the elements listed above are given and discussed in detail. These models enable one to systematically generate system equations using virtual work as a through variable and the displacement (and rotation) as the across variable.

### 3.4.1 Body Elements

Body elements are used to model solid bodies in the system. Depending on the flexibility assumption for the body under consideration, two types of body elements are possible. They are the rigid body element and the flexible body element.

**The Rigid Body Element**

This element models a rigid body.

**Terminal Equation:**

$$\delta W_{e1} = -M_1 \underline{\ddot{r}}_1 \cdot \delta \underline{r}_1 - \left[ \underline{\mathbf{I}}_1 \cdot \underline{\dot{\omega}}_1 + \underline{\omega}_1 \times (\underline{\mathbf{I}}_1 \cdot \underline{\omega}_1) \right] \cdot \delta \underline{\theta}_1 \quad (3.1)$$

$$\underline{F}_1 = -M_1 \underline{\ddot{r}}_1 \quad (3.2)$$

$$\underline{T}_1 = -\underline{\mathbf{I}}_1 \cdot \underline{\dot{\omega}}_1 - \underline{\omega}_1 \times (\underline{\mathbf{I}}_1 \cdot \underline{\omega}_1) + \underline{T}_d \quad (3.3)$$

where $\delta W_{e1}$, $\underline{F}_1$ and $\underline{T}_1$ are respectively the virtual work, force and torque defined for the element; $M_1$ is the mass of the body; $\underline{r}_1$ is the position vector of the center of mass; $\underline{\mathbf{I}}_1$ is the moment of inertia dyadic of the body about its center of mass; $\underline{\omega}_1$ is the angular velocity of the body relative to the inertial reference; the dot superscript denotes the time derivative with respect to the inertial reference frame; $\delta \underline{\theta}_1$ represents the virtual angular displacement of the body. $\underline{T}_d$ collects all the dependent torques [63] for the body. When there are couple moments acting on the body, such as the ones from prismatic joints or applied couple moments, the summation excludes them. These couple moments are modelled separately. Reaction couple moments, always as unknowns, are associated with corresponding joint edges (see Section 3.4.6) and applied couple moments, always given, are modelled as force drivers (see Section 3.4.4).

It is clear that the virtual work defined for the element is the virtual work of the inertial force system of the body.

**Terminal Graph:** a single edge that originates at an inertial reference point and terminates at the mass center of the body (see $e1$ in Fig.3.1).

**The Flexible Body Element**

This element models a flexible body and is a totally new element.

**Terminal Equation:**

$$\delta W_{e1} = -\int_V \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} \, dV + \int_V \delta \mathbf{r}^T (\mathbf{f}_b - \gamma \ddot{\mathbf{r}}) dV \quad (3.4)$$

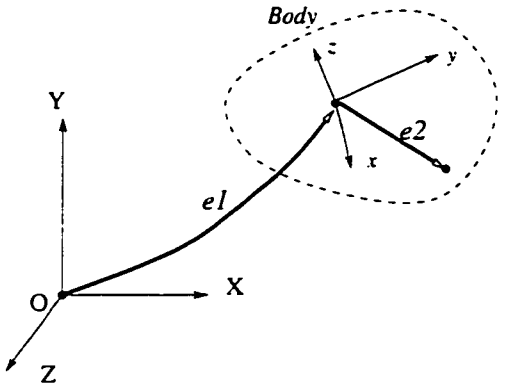



Figure 3.1: Terminal Graphs of Body and Arm Element

$$\underline{F}_1 = -M_1 \ddot{\underline{r}}_1 \tag{3.5}$$

$$\underline{T}_1 = -\underline{\rho}_c \times M_1 \ddot{\underline{r}}_p - \dot{\underline{H}}_p + \underline{T}_d \tag{3.6}$$

where $\delta W_{e1}, \underline{F}_1$ and $\underline{T}_1$ are the virtual work, force and torque defined for the body; $\delta\boldsymbol{\varepsilon} = [\delta\varepsilon_{xx},\ \delta\varepsilon_{yy},\ \delta\varepsilon_{zz},\ 2\delta\varepsilon_{xy},\ 2\delta\varepsilon_{yz},\ 2\delta\varepsilon_{zx}]^T$ is the column matrix of the varied strain components; $\boldsymbol{\sigma} = [\sigma_{xx},\ \sigma_{yy},\ \sigma_{zz},\ \sigma_{xy},\ \sigma_{yz},\ \sigma_{zx}]^T$ is the column matrix of the stress components; $\mathbf{f}_b$ is the body force; $-\gamma\ddot{\mathbf{r}}$ is the component column matrix of the inertial force per unit volume of the body. The force and torque associated with the body edge are designed according to Greenwood [78]. $\underline{\rho}_c$ and $\underline{r}_p$ are respectively the position vector of the mass center of the body relative to the origin of the body reference and the position vector of the origin of the body reference relative to the inertial frame (see Fig.3.2); $\underline{H}_p \triangleq \int_V \underline{\rho}_i \times \dot{\underline{\rho}}_i \gamma dV$ is the angular momentum of the flexible body with respect to the origin of the body fixed frame, where $\underline{\rho}_i$ is the position vector of the infinitesimal mass element $\gamma dV$ with respect to the body reference. $\underline{T}_d$ again collects all the dependent torques on the body.

It is seen that the first term in the virtual work expression is the negative of the virtual work due to internal elastic forces (Shames 1985 [101]) while the second term expresses

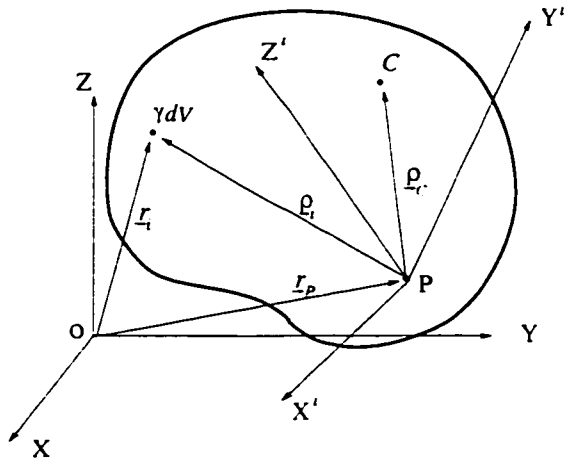



Figure 3.2: Reference Frames for a Flexible Body

the virtual work of body forces, including inertial forces.

**Terminal Graph:** a single edge that starts at the inertial reference point and terminates at the origin of the body reference frame (see edge $e1$ in Fig.3.1).

Note that these terminal equations reduce to those used for their rigid body counterparts when zero deformation and hence zero strains are assumed for the body under consideration. It is also noted that the VW terminal equation defined here holds for the most general cases where the deformation field and stress are independent of any constitutive law as long as the deformation is small [101]; it will be tailored for application to Bernoulli-Euler beams later in Chapter 4.

### 3.4.2 Body Arms

Body arms are usually used to define the locations of imparted actions (e.g., joint reactions, external forces and torques) to the body relative to the center of mass (for rigid bodies) or to the origin of the local frame (for flexible bodies). They are also used to locate and orient other reference frames on the body. Depending on the flexibility assumption for the body on which the arm resides, two types of body arms are possible.

They are the rigid body arms and the flexible body arms.

**The Rigid Body Arm**

The rigid body arm element is used to locate the imparted action to a rigid body, relative to the center of mass of the body.

**Terminal Equations:**

$$\dot{\underline{r}}_2 = \underline{\omega}_1 \times \underline{r}_2 \tag{3.7}$$

$$\delta\underline{r}_2 = \delta\underline{\theta}_1 \times \underline{r}_2 \tag{3.8}$$

$$\ddot{\underline{r}}_2 = \dot{\underline{\omega}}_1 \times \underline{r}_2 + \underline{\omega}_1 \times (\underline{\omega}_1 \times \underline{r}_2) \tag{3.9}$$

$$\mathcal{R}_2 = Constant \tag{3.10}$$

$$\underline{\omega}_2 = 0 \tag{3.11}$$

$$\delta\underline{\theta}_2 = 0 \tag{3.12}$$

$$\dot{\underline{\omega}}_2 = 0 \tag{3.13}$$

where $\underline{r}_2$ is the body arm vector starting from the center of mass and ending at the distal point of application in the same body; $\underline{\omega}_1$ is the angular velocity of the body (and hence body-fixed frames as well); $\mathcal{R}_2$ is the relative rotation transformation matrix of the arm element, i.e., the rotation matrix of a body-fixed frame whose origin is at the distal point with respect to the body-fixed frame at the center of mass; $\underline{\omega}_2$ is the corresponding relative angular velocity of the arm element, i.e., the angular velocity of a body fixed frame whose origin is at the distal point with respect to the body fixed frame at the center of mass.

**Terminal Graph:** an edge originating from the mass center and terminating at the distal point in the same body (see edge $e2$ in Fig.3.1).

It is useful to point out that the through variables associated with arm elements are generally system unknowns and always dependent on other through variables via the

cutset equations.

When an arm is not fixed on a rigid body, henceforth denoted a sliding arm, its graph is the same as that of the fixed arm above. Its terminal equations are defined as follows

$$\dot{\underline{r}}_2 = \overset{\circ}{\underline{r}}_2 + \underline{\omega}_1 \times \underline{r}_2 \tag{3.14}$$

$$\delta\underline{r}_2 = \overset{\circ}{\delta}\underline{r}_2 + \delta\underline{\theta}_1 \times \underline{r}_2 \tag{3.15}$$

$$\ddot{\underline{r}}_2 = \overset{\circ\circ}{\underline{r}}_2 + 2\underline{\omega}_1 \times \overset{\circ}{\underline{r}}_2 + \dot{\underline{\omega}}_1 \times \underline{r}_2 + \underline{\omega}_1 \times (\underline{\omega}_1 \times \underline{r}_2) \tag{3.16}$$

$$\mathcal{R}_2 = Constant \tag{3.17}$$

$$\underline{\omega}_2 = 0 \tag{3.18}$$

$$\delta\underline{\theta}_2 = 0 \tag{3.19}$$

$$\dot{\underline{\omega}}_2 = 0 \tag{3.20}$$

where the notations here have the same meaning as before except that the circle superscript denotes the time derivative relative to the rigid body's reference frame, and $\overset{\circ}{\delta}\underline{r}_2$ means the variation of $\underline{r}_2$ with the rigid body's frame held fixed.

This sliding arm is usually employed to locate a slider block, an external force, or a particle that moves on a rigid body, in the rigid body's frame.

**The Flexible Body Arm**

The flexible body arm element is used to locate the imparted action to a flexible body, with respect to the body-fixed frame.

**Terminal Equations:**

$$\dot{\underline{r}}_2 = \overset{\circ}{\underline{u}} + \underline{\omega}_1 \times \underline{r}_2 \tag{3.21}$$

$$\delta\underline{r}_2 = \overset{\circ}{\delta}\underline{u} + \delta\underline{\theta}_1 \times \underline{r}_2 \tag{3.22}$$

$$\ddot{\underline{r}}_2 = \overset{\circ\circ}{\underline{u}} + 2\underline{\omega}_1 \times \overset{\circ}{\underline{u}} + \dot{\underline{\omega}}_1 \times \underline{r}_2 + \underline{\omega}_1 \times (\underline{\omega}_1 \times \underline{r}_2) \tag{3.23}$$

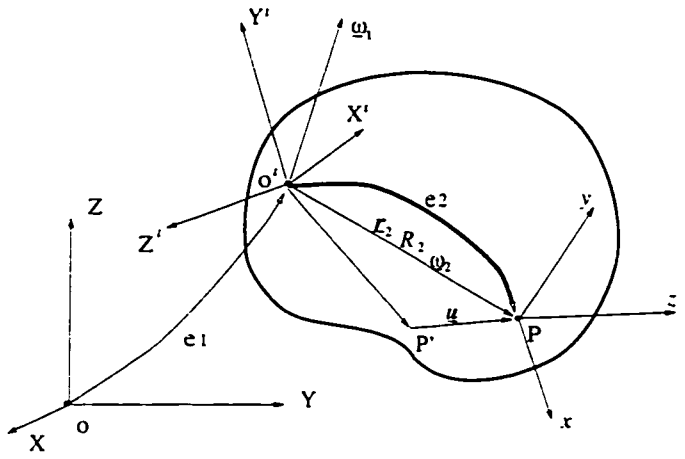



Figure 3.3: Flexible Arm on a Flexible Body

$$\mathcal{R}_2 \quad : \quad \text{depends on the deformation field of the flexible body} \tag{3.24}$$

$$\underset{\sim}{\omega_2} \quad : \quad \text{depends on the deformation field of the flexible body} \tag{3.25}$$

$$\delta\underset{\sim}{\theta_2} \quad : \quad \text{depends on the deformation field of the flexible body} \tag{3.26}$$

where $\underset{\sim}{u}$ is the elastic deformation of a distal point $P$ (see Fig.3.3) that is expressed in the body reference $X^iY^iZ^i$; $\underset{\sim}{\omega_1}$ is the angular velocity of the body reference frame; $\underset{\sim}{r_2}$ is the flexible body arm vector that starts from the origin of the body reference and ends at the distal point; $\mathcal{R}_2$ is the elastic rotation transformation matrix of a body-fixed reference frame $Pxyz$ whose origin is at the distal point with respect to the body frame of the flexible body; $\underset{\sim}{\omega_2}$ is the corresponding angular velocity of the reference frame at the distal point relative to the body frame. From reference [111], $\tilde{\omega}_2 = \frac{d\mathcal{R}_2}{dt}\mathcal{R}_2^T$, where $\tilde{\omega}_2$ is the skew-symmetric matrix of vector $\underset{\sim}{\omega_2}$ expressed in the $X^iY^iZ^i$ frame, and $\delta\tilde{\theta}_2 = \delta\mathcal{R}_2\mathcal{R}_2^T$.

It is important to note that for both $\overset{\circ}{\delta}\underset{\sim}{u}$ and $\delta\underset{\sim}{\theta_2}$ (the varied angular displacement of the distal frame) the variation is carried out with the body reference frame held fixed.

**Terminal Graph:** an edge $e2$ that starts from the origin of the body reference frame and terminates at the distal point, see Figs.3.3 and 3.1.

### 3.4.3 Motion Drivers

This element corresponds to a motion imposed onto a point or reference frame on a rigid or flexible body.

**Terminal Equation:**

$$\delta W_{e3} = \underline{F}_3 \cdot \delta \underline{r}_3 + \underline{T}_3 \cdot \delta \underline{\theta}_3 \tag{3.27}$$

where $\underline{F}_3$ and $\underline{T}_3$ are the force and the torque needed to generate the specified motions. Both $\delta \underline{r}_3$ and $\delta \underline{\theta}_3$ are referred to an inertial frame.

**Terminal Graph:** an edge going from the ground to the driven element.

It is important to point out that although the motion of a motion driver is specified, $\delta \underline{r}_3$ and $\delta \underline{\theta}_3$ in the above expression are not taken to be zero. They are evaluated as any other kinematic quantities of the system without considering the fact that both $\underline{r}_3$ and the rotation are specified. The effect of the motion driver is to be accounted for with system kinematic loop closure equations.

It must be noted too that the above graph-theoretic model is valid only when the motion driver is applied from an inertial base. In the case of motion drivers that are applied between bodies of a system, as is common in robotic applications, the graph-theoretic model takes the form similar to that presented for the spring-damper-actuator (see Section 3.4.8).

### 3.4.4 Force Drivers

Elements of this type represent external forces or torques applied to a rigid or flexible body.

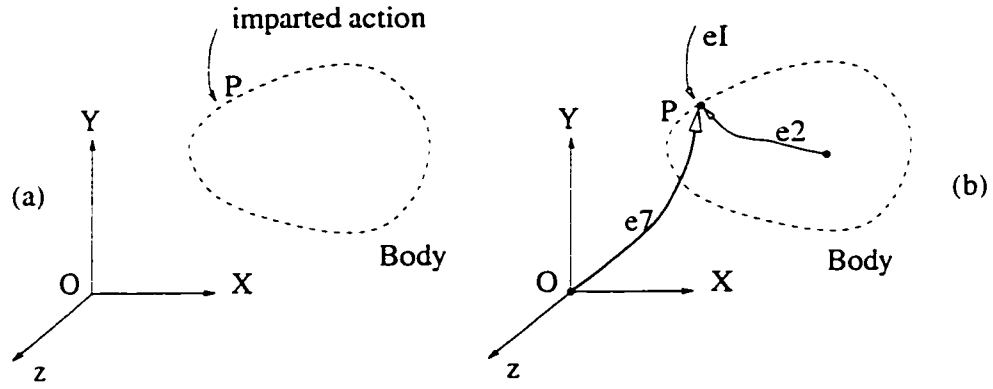



Figure 3.4: Imparted Action and Dependent Virtual Work Element

**Terminal Equation:**

$$\delta W_{e5} = \underset{\sim}{F}_5(t) \cdot \delta \underset{\sim}{r}_5 + \underset{\sim}{T}_5(t) \cdot \delta \underset{\sim}{\theta}_5 \tag{3.28}$$

where $\underset{\sim}{F}_5(t)$ and $\underset{\sim}{T}_5(t)$ are the applied force and torque; $\delta \underset{\sim}{r}_5$ is the virtual displacement of the point of application of the force; $\delta \underset{\sim}{\theta}_5$ is the virtual rotation of a local frame at the point of application of the torque. Both $\delta \underset{\sim}{r}_5$ and $\delta \underset{\sim}{\theta}_5$ are referred to an inertial frame.

### 3.4.5 Dependent Virtual Work Elements

This element, often referred to as a dependent element for short, is also a completely new element. It does not correspond to any physical component in a usual sense, but it has its counterpart in the traditional GT approach, i.e., the dependent torque driver [63]. Similar to the dependent torque, the dependent virtual work element is designed so that any actions imparted to a body, such as joint reactions or actions from springs, can be included in the modelling in terms of its virtual work.

**Terminal Equations:**

$$\delta W_{e7} = \underset{\sim}{F}_I \cdot \delta \underset{\sim}{r}_7 + \underset{\sim}{T}_I \cdot \delta \underset{\sim}{\theta}_7 \tag{3.29}$$

$$\underset{\sim}{F}_7 = 0 \tag{3.30}$$

$$\underset{\sim}{T}_7 = 0 \tag{3.31}$$

where $\underset{\sim}{F}_I$ and $\underset{\sim}{T}_I$ are the force and the torque exerted by the imparted action; $\delta\underset{\sim}{r}_7$ and $\delta\underset{\sim}{\theta}_7$ are respectively the variation of the position vector and the angular position of the application point of the imparted action, all referred to the inertial frame; $\underset{\sim}{F}_7$ and $\underset{\sim}{T}_7$ are the force and torque for this dependent virtual work element.

**Terminal Graph:** an edge $e7$ going from the inertial reference point to the application point of the imparted action, as shown in Fig.3.4 where $eI$ represents the imparted action and $e2$ is a body arm element locating the imparted action.

It is noted that the virtual work expressions for motion driver, force driver and the dependent element all have the same form.

### 3.4.6 Ideal Joints

This element represents a physical joint, see Fig.3.5. In fact, the dependent virtual work element was first conceived [116] during an examination of the virtual work equation of an ideal joint.

**Terminal Graph:** to correctly incorporate an ideal joint in a graph-theoretic system model using virtual work as the through variable, the strategy used here is to first release the joint and replace the effect of the joint with its reaction forces on the bodies it connected, as usually done in the pure-mechanics modelling of systems. Then, the joint reactions acting on each of the bodies are modelled with a dependent virtual work element (see Section 3.4.5). Thus, two virtual work elements that start from the origin of an initial frame are first used to model the joint reactions through their virtual work. A third edge called the joint edge is used to account for the kinematic characteristics of the joint, as has been done in the traditional GT approach.

In short, three edges are needed to model an ideal joint, see Fig.3.5(c). Two of them, $e71$ and $e72$, are dependent virtual work elements, each starting from the origin of an inertial frame and terminating at either of the joint's connection points on the bodies it connects. The third edge $e6$, a joint edge, joins the two points of connection only. Also in this figure, $e21$ and $e22$ are two arm elements locating the joint connection point on each of the bodies.

**Terminal equations:** The terminal equations for the dependent elements are the same as those given in Section 3.4.5. The terminal equations for the third edge is

$$\begin{cases} \delta W_{e6} \triangleq 0 \\ \text{plus the conventional force and kinematic constraint equations [63]} \end{cases} \tag{3.32}$$

It helps understand the model to emphasize that from a pure mechanics point of view, the effect of joint reactions is accounted for through their virtual work.

It is important to point out here that due to the zero-virtual-work property of an ideal joint and the way the present graph-theoretic approach models it, one always has

$$\delta W_{e71} + \delta W_{e72} = 0 \tag{3.33}$$

To prove this, it is first written by definition

$$\delta W_{e71} = \underset{\sim}{F}_6 \cdot \delta \underset{\sim}{r}_b + \underset{\sim}{T}_6 \cdot \delta \underset{\sim}{\theta}_b \tag{3.34}$$

$$\delta W_{e72} = (-\underset{\sim}{F}_6) \cdot \delta \underset{\sim}{r}_a + (-\underset{\sim}{T}_6) \cdot \delta \underset{\sim}{\theta}_a \tag{3.35}$$

in which $\underset{\sim}{F}_6$ and $\underset{\sim}{T}_6$ are the joint force and torque, $\delta \underset{\sim}{r}_a$ and $\delta \underset{\sim}{r}_b$ are the virtual displacements of the joint connection points, and $\delta \underset{\sim}{\theta}_a$ and $\delta \underset{\sim}{\theta}_b$ are the virtual rotational displacements of frames at the joint connection points.

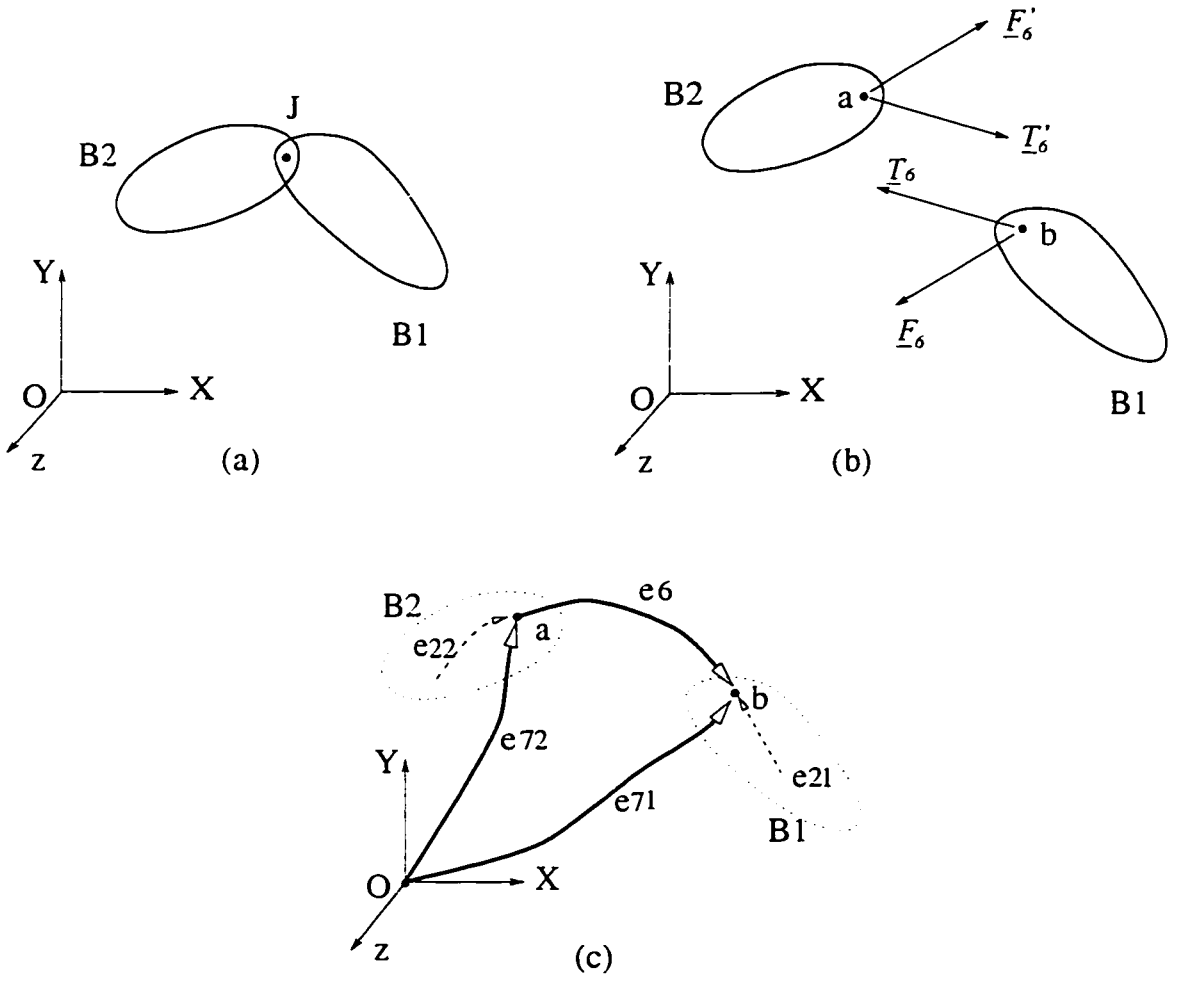



Figure 3.5: (a) The Ideal Joint; (b) Exposed Joint Reactions; (c) Graph for the Joint

Thus, it follows that

$$\delta W_{e71} + \delta W_{e72} = \underline{F}_6 \cdot (\delta \underline{r}_b - \delta \underline{r}_a) + \underline{T}_6 \cdot (\delta \underline{\theta}_b - \delta \underline{\theta}_a) \quad (3.36)$$

$$= \underline{F}_6 \cdot \delta \underline{r}_{joint} + \underline{T}_6 \cdot \delta \underline{\theta}_{joint} \quad (3.37)$$

For an ideal joint, its motion space is orthogonal to its reaction-force space. Therefore, equation (3.33) results.

Since joint reactions appear only in virtual work expressions for dependent elements, and these virtual work expressions will appear in the system virtual work equation(s), equation (3.33) enables the present GT approach to eliminate joint reactions (thus any Lagrange multipliers) from the system virtual work equation(s) right from its start, as long as the dependent virtual work expressions appear in pairs, without having to resort to any numerical or projection procedure that has been in use in existing approaches [102, 103]. The result is a set of system kinetic equations that are free of any joint reactions (thus any Lagrange multipliers), as discussed in Chapter 5.

### 3.4.7 Free Joints

For the current graph-theoretic approach to be able to incorporate free-floater systems, like spacecraft and satellite systems, the free joint element is introduced here. A free joint is defined as a joint that allows six degrees of freedom.

**Terminal Graph:** The terminal graph of a joint consists of two edges: one is a dependent virtual work element edge, the other a joint edge. The virtual work edge starts from the origin of an inertial frame and ends at any point of the floater body. The joint edge runs the same way, as shown in Fig.3.6.

**Terminal equations:**

$$\underline{F}_6 = 0 \quad (3.38)$$

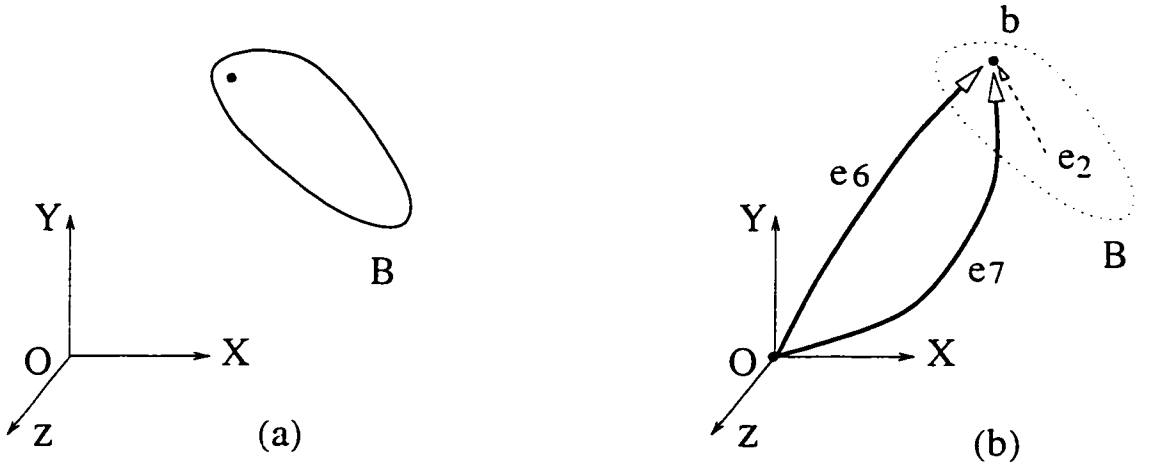



Figure 3.6: (a) Free Joint; (b) Terminal Graph for Free Joint

$$\underset{\sim}{T}_6 = 0 \tag{3.39}$$

$$\delta \underset{\sim}{W}_{e6} = 0 \tag{3.40}$$

$$\delta \underset{\sim}{W}_{e7} = 0 \tag{3.41}$$

This model of free joint element can be derived from the previous ideal joint model. Indeed, we only need to regard either of the two bodies as an inertial reference frame, and set joint reaction forces there to zero.

### 3.4.8 Spring-Damper-Actuator Elements

This element defines a spring-damper-actuator (SDA) that connects any two bodies and is actually a compound element, see Figure 3.7(a). Each individual element, i.e., a spring, damper, or force actuator, can be regarded as a special case of it.

**Terminal Graph:** Along the same spirit followed in modelling the ideal joint, three edges are used to represent this compound element graphically. Two of them are the dependent elements. The third is an edge that joins the two connection points of the spring, as shown in Figure 3.7(b).

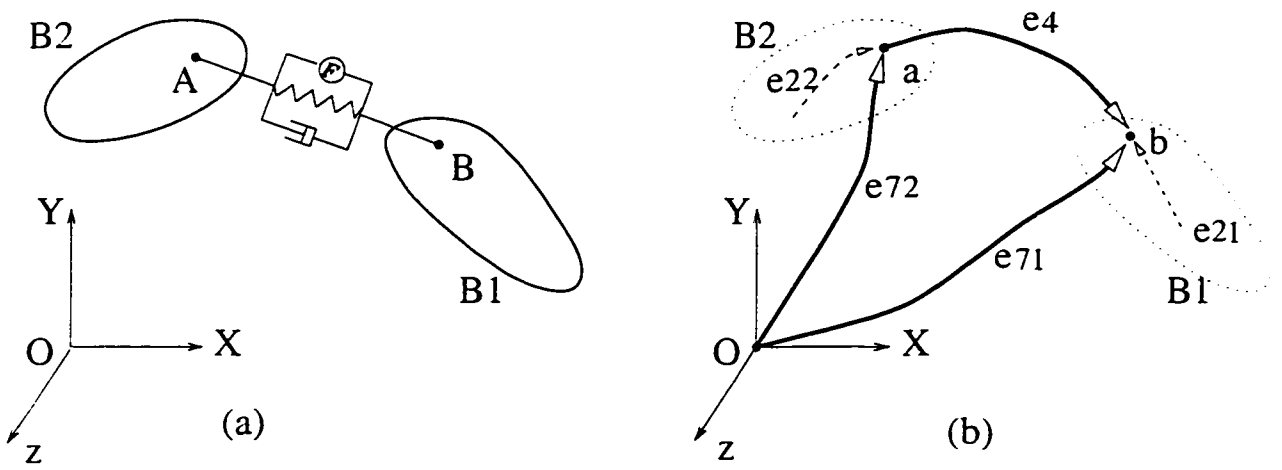



Figure 3.7: (a) Spring-Damper-Actuator and Body in Connection; (b) Terminal Graph for Spring-Damper-Actuator

**Terminal Equations:** The terminal equations for the dependent elements are given in Section 3.4.5. The terminal equations for the third edge take the following form:

$$\delta W_{e4} = 0 \tag{3.42}$$

and for translational spring-damper-actuators

$$\underset{\sim}{F}_4 = -\left[k\,(r_4 - r_{40}) + d\,(\underset{\sim}{v}_4 \cdot \hat{u}_4) + F(t)\right]\hat{u}_4 \tag{3.43}$$

$$\underset{\sim}{T}_4 = 0 \tag{3.44}$$

and for rotational spring-damper-actuators

$$\underset{\sim}{F}_4 = 0 \tag{3.45}$$

$$\underset{\sim}{T}_4 = -\left[k_r\,(\theta_4 - \theta_{40}) + d_r\,\dot{\theta}_4 + T(t)\right]\hat{u}_4 \tag{3.46}$$

$\delta W_{e4}$, $\underset{\sim}{F}_4$ and $\underset{\sim}{T}_4$ are respectively the virtual work, force and torque defined for the spring

edge $e4$; in equation (3.43), $r_4$ and $r_{40}$ are respectively the deformed and undeformed lengths of the translational spring; $d$ and $\underline{v}_4$ are the damping coefficient and the velocity of the translational element; $F(t)$ is the magnitude of the actuator force parallel to the compound element; $\hat{u}_4$ is the unit vector that defines the direction of the element in space. A parallel explanation applies to the torque expression $\underline{T}_4$.

## 3.5 Vertex Postulate Satisfied

Since the traditional employment of displacements and rotations as the across variables has been preserved in the present investigation, it is only needed to prove that the graph-theoretic models presented in the last section satisfy the vertex postulate so that correct system equations can indeed be obtained using these elements. To achieve this end, let us first take a look at a system of rigid bodies. A body of this system is isolated and shown with all the possible elements connected that can impart action to it, as shown in Fig.3.8. The corresponding free-body diagram is depicted in Fig.3.9, where $\underline{F}^i$ and $\underline{T}^i$ are the inertia force and couple moment about the center of mass of the rigid body, respectively.

Let us first calculate the virtual work of all the forces acting on the body by giving the body frame $X^iY^iZ^i$ a virtual displacement and rotation denoted by $\delta\underline{r}^i$ and $\delta\underline{\theta}^i$ respectively, and assume that as a result, the virtual displacement and rotation at points $P_j \quad (j = 1, 4)$ are $\delta\underline{r}_j$ and $\delta\underline{\theta}_j \quad (j = 1, 4)$ respectively. Thus, applying the Principle of Virtual Work to the body, one has

$$\begin{aligned} VW^i &= \delta\underline{r}^i \cdot \underline{F}^i + \delta\underline{\theta}^i \cdot \underline{T}^i + \delta\underline{r}_1 \cdot \underline{F}_1 + \delta\underline{\theta}_1 \cdot \underline{T}_1 \\ &+ \delta\underline{r}_2 \cdot \underline{F}_2 + \delta\underline{\theta}_2 \cdot \underline{T}_2 + \delta\underline{r}_3 \cdot \underline{F}_3 + \delta\underline{\theta}_3 \cdot \underline{T}_3 + \delta\underline{r}_4 \cdot \underline{F}_4 \qquad (3.47) \\ &= 0 \qquad (3.48) \end{aligned}$$

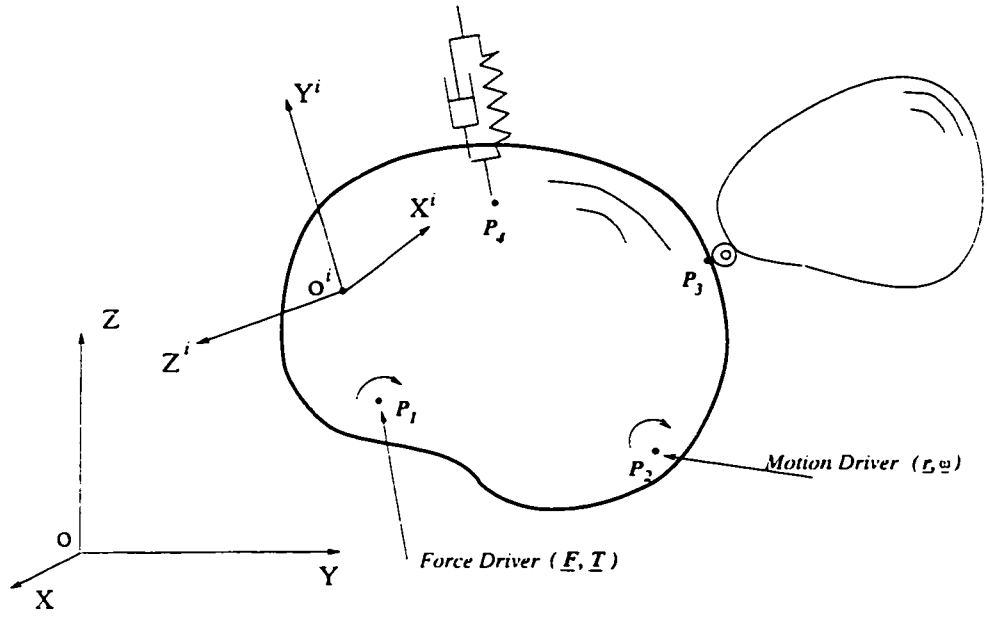



Figure 3.8: A Rigid Body in Connection

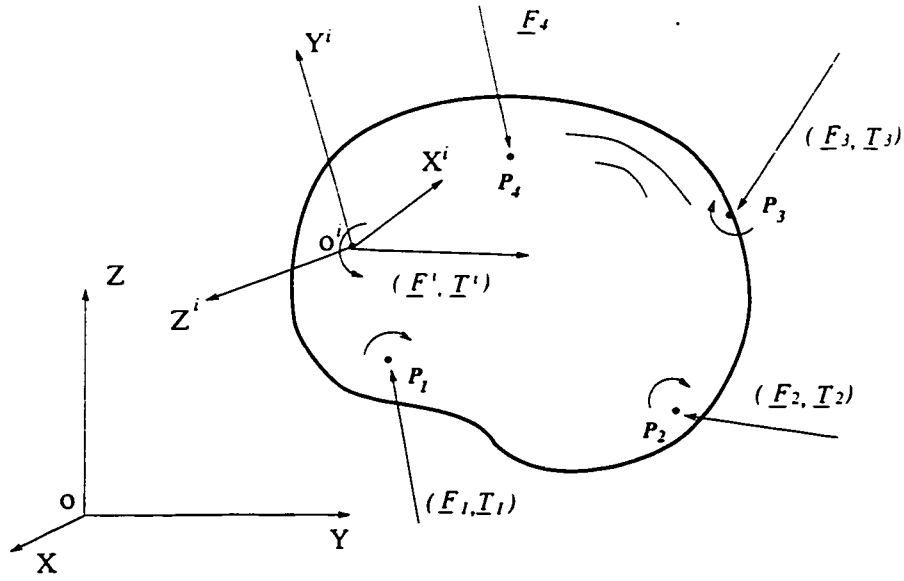



Figure 3.9: A Rigid Body under Action

The virtual work done by the same force system is now calculated using the graph-theoretic models presented in last section to show that the vertex postulate is satisfied, i.e., (3.47) can be generated from a cutset equation of the graph.

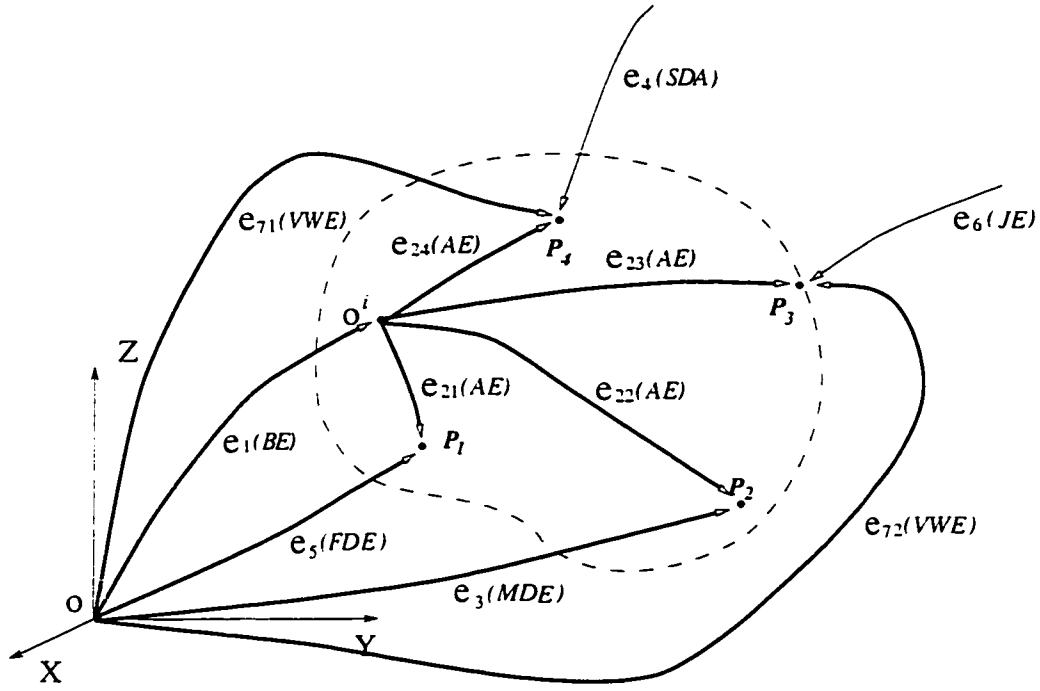



Figure 3.10: The Graph for Rigid Body under Actions

To start with, the graph for the body depicted in Fig.3.8 is constructed, and shown in Figure 3.10. The vertex expression for node $O^i$ reads

$$\delta W_{e1} - \delta W_{e21} - \delta W_{e22} - \delta W_{e23} - \delta W_{e24} \tag{3.49}$$

It is then noted that, as mentioned earlier, the virtual work of a body arm takes whatever

value is required to satisfy the vertex equation of a node, that is

$$\begin{cases} \delta W_{e21} &= -\delta W_{e5} \\ \delta W_{e22} &= -\delta W_{e3} \\ \delta W_{e23} &= -\delta W_{e72} - \delta W_{e6} \\ \delta W_{e24} &= -\delta W_{e71} - \delta W_{e4} \end{cases} \tag{3.50}$$

They are in fact the vertex equations for nodes $P_1$, $P_2$, $P_3$ and $P_4$.

Substituting this set of equations into expression (3.49) gives

$$\delta W_{e1} + \delta W_{e5} + \delta W_{e3} + \delta W_{e72} + \delta W_{e6} + \delta W_{e4} + \delta W_{e71} \tag{3.51}$$

which reduces to

$$\delta W_{e1} + \delta W_{e5} + \delta W_{e3} + \delta W_{e72} + \delta W_{e71} \tag{3.52}$$

since it has been defined that the virtual work associated with the joint edge and SDA edge is zero.

Using the forces illustrated in Figure 3.9, the virtual displacements and rotations assumed before, and the defined VW terminal equations for the elements involved, expression (3.52) can be rewritten as

$$\delta \underline{r}^i \cdot \underline{F}^i + \delta \underline{\theta}^i \cdot \underline{T}^i + \delta \underline{r}_1 \cdot \underline{F}_1 + \delta \underline{\theta}_1 \cdot \underline{T}_1 + \delta \underline{r}_2 \cdot \underline{F}_2 + \delta \underline{\theta}_2 \cdot \underline{T}_2 + \delta \underline{r}_3 \cdot \underline{F}_3 + \delta \underline{\theta}_3 \cdot \underline{T}_3 + \delta \underline{r}_4 \cdot \underline{F}_4 \tag{3.53}$$

which is exactly the same as equation (3.47). Thus it sums to zero, too. This in turn reduces expression (3.49) to zero. It follows that the vertex postulate for node $O^i$ is satisfied for the graph-theoretic models presented in the last section.

As a by-product, it is noted that expression (3.52) sums to zero as well. This equation in fact indicates that the vertex postulate for node $O$ holds for the sub-graph associated

with the body under consideration and can thus be generated from a cutset equation with any of the edges incident on node $O$ as a defining edge, including the rigid body edge (for a body tree). At the same time, the satisfaction of the vertex postulate at node $O$ for the sub-graph also leads to the satisfaction of the vertex postulate at the same node for the system graph, i.e., when all bodies in the system are considered.

One needs only to substitute the first two terms in equations (3.47) and (3.53) with

$$-\int_V \delta\boldsymbol{\varepsilon}^T \boldsymbol{\sigma}\, dV + \int_V \delta\mathbf{r}^T(\mathbf{f}_b - \rho\,\ddot{\mathbf{r}})dV \tag{3.54}$$

and repeat the same steps as before to show that the vertex postulate holds when a flexible body is considered.

Thus, it is concluded that using virtual work as the through variable satisfies the vertex postulate when system elements are properly modelled as in Section 3.4, and that cutset equations of any tree selection give the same results as when the Principle of Virtual Work is applied to the free-body diagram of the corresponding part of the system.

## 3.6 Concluding Remarks

The validity of using of virtual work as a through variable for the new extended set of elemental graphs has been justified both philosophically and mathematically. A clear definition of a through variable has been also given for the present investigation. The graph-theoretic models of common elements encountered in mechanical systems have been presented. For convenience of reference, these models are also collected in Table 3.2.

A few notes are in order at this point. First, it is obvious that because of the ultimate goal that the proposed formulation be capable of dealing with systems involving flexible bodies, and also because of the intrinsic complexity of such systems, the terminal equations established here for a component are much more involved than those for the same

| Elements | Terminal Equation | The Graph |
|---|---|---|
| Rigid body | $\delta W_{e1} = -M_1 \underline{\ddot{r}}_1 \cdot \delta \underline{r}_1 - \left[ \underline{\mathbf{I}}_1 \cdot \underline{\dot{\omega}}_1 + \underline{\omega}_1 \times (\underline{\mathbf{I}}_1 \cdot \underline{\omega}_1) \right] \cdot \delta \underline{\theta}_1$<br>$\underline{F}_1 = -M_1 \underline{\ddot{r}}_1$<br>$\underline{T}_1 = -\underline{\mathbf{I}}_1 \cdot \underline{\dot{\omega}}_1 - \underline{\omega}_1 \times (\underline{\mathbf{I}}_1 \cdot \underline{\omega}_1) + \underline{T}_d$ | an edge originating from inertial reference point and terminating at center of mass |
| Flexible body | $\delta W_{e1} = -\int_V \delta \epsilon^T \sigma \, dV + \int_V \delta r^T (\mathbf{f}_b - \rho \ddot{\mathbf{r}}) dV$<br>$\underline{F}_1 = -M_1 \underline{\ddot{r}}_1$<br>$\underline{T}_1 = -\underline{\rho}_c \times M_1 \underline{\ddot{r}}_p - \underline{\dot{H}}_p + \underline{T}_d$ | an edge originating from inertial reference point and terminating at origin of body reference frame |
| Rigid body arm | $\underline{\dot{r}}_2 = \underline{\omega}_1 \times \underline{r}_2$<br>$\delta \underline{r}_2 = \delta \underline{\theta}_1 \times \underline{r}_2$<br>$\mathcal{R}_2 = Constant$<br>$\underline{\omega}_2 = 0$<br>$\delta \underline{\theta}_2 = 0$ | an edge originating from reference point and terminating at origin of body reference frame |
| Flexible body arm | $\underline{\dot{r}}_2 = \overset{\circ}{\underline{u}} + \underline{\omega}_1 \times \underline{r}_2$<br>$\delta \underline{r}_2 = \overset{\circ}{\delta \underline{u}} + \delta \underline{\theta}_1 \times \underline{r}_2$<br>$\mathcal{R}_2$ : depending on deformation field<br>$\underline{\omega}_2$ : depending on deformation field<br>$\delta \underline{\theta}_2$ : depending on deformation field | |
| Motion driver | $\delta W_{e3} = \underline{F}_3 \cdot \delta \underline{r}_3 + \underline{T}_3 \cdot \delta \underline{\theta}_3$ | an edge joining the driving to the driven element |
| Force driver | $\delta W_{e5} = \underline{F}_5(t) \cdot \delta \underline{r}_5 + \underline{T}_5(t) \cdot \delta \underline{\theta}_5$ | an edge joining the driving and the driven element |

Table 3.1: Graph-Theoretic Elements and Their Models

| Elements | Terminal Equation | The Graph |
|---|---|---|
| Dependent element | $\delta W_{e7} = \underline{F}_I \cdot \delta \underline{r}_7 + \underline{T}_I \cdot \delta \underline{\theta}_7$<br>$\underline{F}_7 = 0$<br>$\underline{T}_7 = 0$ | an edge joining inertial reference point to head point of an arm element |
| Ideal joint | $\delta W_{e6} = 0$ plus conventional force and kinematic constraint equations in addition to terminal equations for two dependent elements | an edge joining two points of connection plus two dependent elements |
| Spring-Damper-Actuator | $\delta W_{e4} = 0$<br>$\underline{F}_4 = -\left[k(r_4 - r_{40}) + d(\underline{v}_4 \cdot \hat{u}_4) + F(t)\right] \hat{u}_4$<br>$\underline{T}_4 = -\left[k_r(\theta_4 - \theta_{40}) + d_r\dot{\theta}_4 + T(t)\right] \hat{u}_4$<br>in addition to terminal equations for two dependent elements | an edge joining the two points of connection plus two dependent elements |

Table 3.2: Graph-Theoretic Elements and Their Models (Cont'd)

component when only rigid systems are considered.

With virtual work employed as a through variable, the graph-theoretic models now have three sets of variables defined on each edge: the kinematic quantities, the force and moment, and the virtual work. The use of virtual work as a through variable in no way excludes the force and torque that are conventionally associated with each of the edges in the system graph. In fact, owing to the way the components are modelled, not only the virtual work satisfies the vertex postulate as shown above, the force and torque defined for each edge in the system graph also satisfy the vertex postulate which conforms respectively to Newton's and Euler's equations. This can be easily seen by setting the virtual work of each edge to zero and proceeding with the traditional graph-theoretic procedure to obtain system equations in terms of the balance of forces and moments. This set of equations alone, however, is not able to produce a sufficient number of dynamic equations for flexible systems although they are for rigid body systems.

It should also be noted that the virtual work defined for an edge is not necessarily defined in terms of the force and torque and the kinematic quantities of the same edge,

as in the case of virtual work elements and body elements.

A brief comparison would show that Tellegen's theorem holds only for the whole system while the Principle of Virtual Work, in the form of cutset equations, applies at the vertex level (thus subgraph level) as well.

A few other points related to graph-theoretic models of elements are worth mentioning. First, the virtual work expressions for the motion driver, the force driver, and the dependent virtual work element are of the same nature, summing virtual works of the force and torque applied at a point. Only they are used for different components.

A simpler model can be constructed for an SDA element appearing between the inertial base and any other body in the system. This model, however, will not be as general as the one provided in Section 3.4.8. The model presented there for an inter-body SDA can be extended to model inter-body motion drivers and force drivers as well. The inter-body model and the inertial-base-applied model for the same inertial-base-applied motion driver or force driver will produce the same terms in system equations.

As a final note, although rotation of finite magnitude can be used as an across variable for two-dimensional problems, it is generally acknowledged that finite rotation is no longer a legitimate choice for across variables in the traditional sense when general three dimensional problems are considered. This is in spite of the fact that, ironically, a finite rotation meets the requirement in the "definition" of an across variable of being able to be measured in parallel in the operational approach. This leads to broad implications on the application of the traditional GT approach to three-dimensional multibody systems. The first one is that the traditional circuit equation of summative nature is impotent when it comes to calculating the orientation of a frame in space. It has to be obtained through the use of a non-traditional circuit equation (if one could borrow the same terminology) of a multiplicative nature (see Chapter 6 for details), in conjunction with the knowledge of the sequence in which the circuit edges are connected with one another. It is the non-

commutative and multiplicative characteristics of finite rotations that render them unfit for the choice of traditional across variable. As a result, they do not satisfy the Tellegen's theorem which holds only for complementary variables that satisfy the traditional circuit and cutset equations of a summative nature.

To reinstate finite rotations to an across variable, perhaps one can redefine an across variable as one for which the correct physical equations can be generated from the information contained in the circuit matrix, and not just from the traditional circuit equations of a summative nature. A similar redefinition of the through variable can also be made. These redefinitions, while guaranteeing relevant physical laws satisfied, may have no place for Tellegen's theorem and the traditional manipulation techniques of system equations [74] at all.

# Chapter 4

# Graph-Theoretic Modelling of Bernoulli-Euler Beam

In the previous chapter, the virtual work (VW) and kinematic terminal equations for general flexible bodies were given. In this chapter, the Bernoulli-Euler beam is studied in the context of multibody dynamics and its VW and kinematic equations suitable for use in the proposed graph-theoretic approach are evaluated. More complex bodies, such as plates and shells, are not studied in the current investigation.

The investigation of kinematics of beams is a fundamental ingredient in modelling flexible body dynamics. As the motion of the system under study becomes more and more complicated, three-dimensional kinematic analysis of beams is required. The term three-dimensional kinematics refers, in this chapter, to the determination of the deformation field and the relative rotation matrix, also called elastic rotation, between two sections due to elasticity of the beam that undergoes longitudinal deformation, flexural bendings as well as torsion, without considering the cause of these deformations.

There are three principle approaches to the kinematics of Bernoulli-Euler beams: the

curvature approach, the two-Euler-angle approach and the traditional assumed deformation field approach.

The curvature approach (Piedboeuf [87]) takes advantage of the differential relationship between the curvature and the rotation matrix of one coordinate system with respect to another. This relationship is analogous to the one between the angular velocity and the rotation matrix of one coordinate system relative to another. For beams with small deformations, the curvature between two cross sections due to elasticity can be expressed as a function of spatial derivatives of the deformations of the centroidal line and the twist angle of the beam. The differential equation governing the curvature and the rotation matrix is then integrated to produce the rotation matrix. The final deformation description of the centroidal line is obtained afterwards by considering the differential geometry of the beam. The deformation field is then easily determined from the centroidal line deformation and the elastic rotation matrix using the plane-remains-plane assumption for engineering beams. It is noted that Piedboeuf [87] did not consider the beam's axial deformation in his formulation of the kinematics.

The two-Euler-angle approach was first proposed by Alkire [104] after at least ten years of intensive work by helicopter blade dynamicists (Hodges and Dowell [105], Hodges and Orimiston [106]). It was also employed recently by Pai and Nayfeh [107]. This approach, as its name suggests, separates the elastic rotation into two consecutive Euler angle rotations: rotation induced by the deformation of the centroidal line and the angle of twist due to torsion. The first angle of rotation, i.e., the angle caused by the centroidal line deformation, and its corresponding rotation matrix are determined by considering the deformation of the centroidal line. The effect of the twist angle measured with respect to the deformed position of the axial axis of the beam is then added on afterwards. The deformation field of the beam can be constructed with ease, using again the plane-remains-plane assumption.

The mainstream approach to the three-dimensional kinematic analysis of beams for flexible multibody dynamics simulations is the assumed displacement field approach (Vigneron [8], Kaza and Kvaternik [9], Ider [108], Sharf [57]). In this approach, as in the two-Euler-angle approach, the starting point may be regarded to be the assumption of the deformation pattern of the beam's centroidal line and the angle of twist with respect to the deformed axial axis of the beam, but the elastic rotation is determined in different fashions. One method, claimed in Damaren and Sharf [56] as exact, makes use of and numerically integrates the differential relationship between the elastic rotation matrix and its corresponding angular velocity to give the elastic rotation matrix. In this method, however, the numerical angular velocity cannot be obtained without some approximation of its relation with the deformation field variables, and it is this approximate relation that determines the accuracy of the rotation matrix thus found. Besides, this method of obtaining the rotation matrix is in general not amenable to symbolic manipulation of equations because of the numerical integration required beforehand. Another method used is the first-order approximation of the rotation matrix in the deformation variables.

Once the displacement field and the elastic rotation matrix have been chosen, it is a straightforward matter to determine other kinematic quantities, such as the relative velocity of a generic particle or the relative angular velocity of a cross section of the beam, that are needed for use with the dynamic laws to generate system equations.

In this chapter, a closer examination is made of the traditionally used displacement field for Bernoulli-Euler beams with particular attention being paid to its efficacy when it is used to obtain an elastic rotation matrix that is of second-order in the deformation variables. Then in the second section, adopting Alkire's two-Euler angle approach, a complete second-order elastic rotation matrix for a Bernoulli-Euler beam is presented and based on it a new complete second-order displacement field is proposed. Kinematic terminal equations that are consistent with this displacement field are given in the end.

These complete $O(2)$ displacement field and rotation matrix are used in the third section to evaluate the VW of the beam's body and elastic forces (i.e., its VW terminal equations). Also in this section, the significance of using a complete second-order displacement field in the modelling is discussed qualitatively.

Before going further, let us make clear what is meant by the Bernoulli-Euler beam, i.e., the assumptions made on the beam's deformation patterns in addition to the fact that it is a slender prismatic bar. By Bernoulli-Euler beam, it is meant that the beam's cross sections remain unchanged (both in plane and out of plane) and they remain perpendicular to the reference axis as the beam deforms. From this, it is seen that the following assumptions are implied:

1. The Beam's elastic configuration is completely determined by four variables, corresponding to the axial deformation, the two lateral bendings, and torsion.

2. Shear due to bending and warping due to torsion are neglected.

3. The beam experiences small strain, but possibly large deflections.

The flexible beam is shown in Figure 4.1 as it is related to its associated reference frames, which will be referred to throughout the chapter. Frame $OXYZ$ is the inertial reference frame. $A_i X^i Y^i Z^i$ is the body reference frame for the beam whose origin is located at the center of a cross section and whose axes are aligned with the area principal axes at that point, with the $X^i$ axis directed along its undeformed centroidal axis. This also means that the $X^i$ axis remains tangent to the centroidal line as the beam deforms. Frame $B_i x_{\mathcal{R}} y_{\mathcal{R}} z_{\mathcal{R}}$ is the reference frame defined for the cross section that is a distance $S_R^i$ from the origin of the body reference frame of the beam. This cross sectional frame is a result of a translation of the body reference frame along the undeformed centroidal axis. Due to the assumption that plane sections remain plane, this frame, albeit a body-fixed one, will move along with the cross section without deformation to $C_i x_f^i y_f^i z_f^i$ as the beam

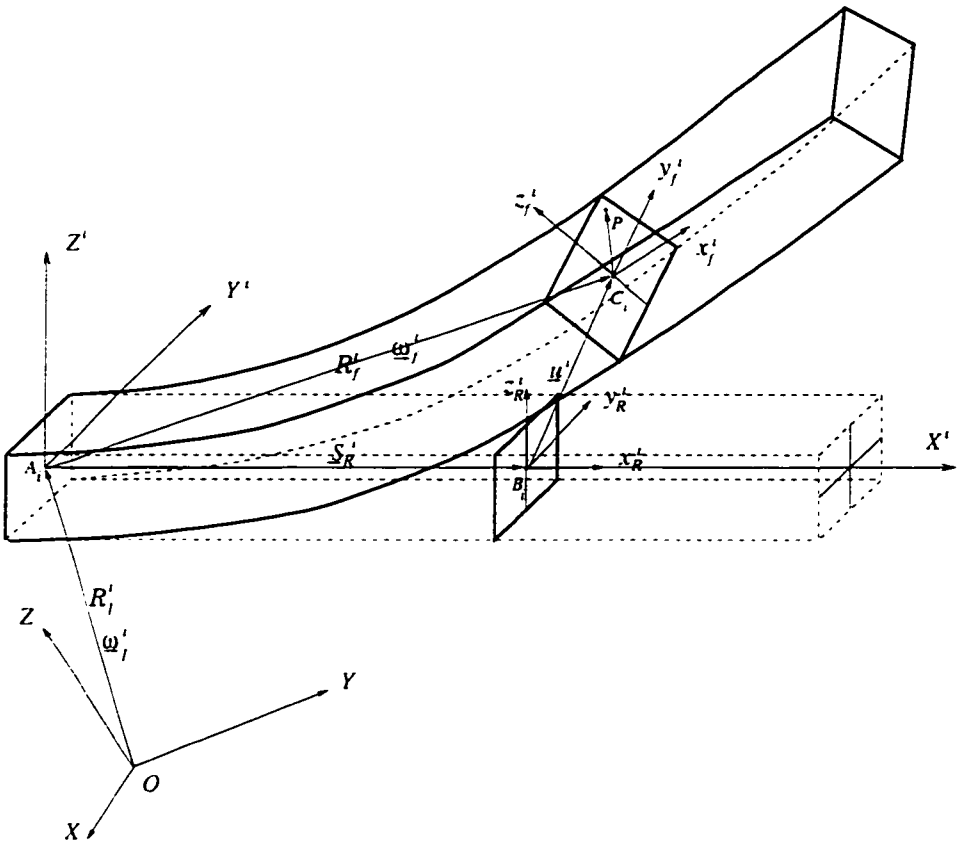



Figure 4.1: The Flexible Beam and its Reference Frames

undergoes deformation, thus defining the position and orientation of the section at any instant of time. Other symbols will be explained as they are referred to.

It is clear that any generic particle of the beam can be located by four deformation variables associated with the centroidal axis as the beam deforms. The four variables are chosen to be: $u(x,t)$ the axial deformation of the centroidal-axis minus its foreshortening along the $X^i$ axis (i.e., $u(x,t)$ is a pure axial deformation), $v(x,t)$ and $w(x,t)$ the bending deformations in the direction of $Y^i$ and $Z^i$ axes respectively, and $\phi(x,t)$ the angle of twist of the beam section the particle is on with respect to the deformed centroidal axis.

## 4.1 Traditional Assumed Displacement Field

To start, let $\underset{\sim}{\mathcal{P}}_P$ be the position vector of a generic point $P$ of the beam with respect to the origin of the body frame (see Figure 4.1), i.e.

$$\underset{\sim}{\mathcal{P}}_P \triangleq \underset{\sim}{\mathcal{P}}_{A,P} = \{\underset{\sim}{e}^i\}^T \begin{Bmatrix} X^i \\ Y^i \\ Z^i \end{Bmatrix} \tag{4.1}$$

where $\{\underset{\sim}{e}^i\} = \begin{Bmatrix} \vec{i}^i \\ \vec{j}^i \\ \vec{k}^i \end{Bmatrix}$, is the unit base vectrix of the body reference frame, and $X^i$, $Y^i$ and $Z^i$ are the current or post-deformation coordinates of the point in the body reference frame $X^iY^iZ^i$.

In dealing with Bernoulli-Euler beams using a Lagrangian description, the following displacement field of a generic point $P$ having a position of $(x, y, z)$ in the body reference frame before the beam deforms, neglecting shear due to bending and warping due to torsion, is often assumed (Vigneron [8], Kaza and Kvaternik [9], Han, Xu and Zu [31]):

$$\begin{cases} X^i &= x + u - y\,v' - z\,w' - \frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi \\ Y^i &= v - z\,\phi + y \\ Z^i &= w + y\,\phi + z \end{cases} \tag{4.2}$$

where $u$, $v$, $w$ and $\phi$ are as defined above. The primes denote a differentiation with respect to $x$. The integral accounts for the foreshortening of the centroidal line in the $X^i$ direction due to the bendings (Shames [101]).

Now let us find the rotation matrix of a cross-sectional frame $C_i x_f^i y_f^i z_f^i$ with respect to the body frame $X^iY^iZ^i$ directly from this displacement field.

Recall from continuum mechanics, when no shear stress due to bending and no warping due to torsion are considered, the principal strains at $C_i$ (see Figure 4.1) coincide with the deformed coordinate lines at the same point whose unit vectors are chosen to be those of the cross-sectional frame $x_f^i y_f^i z_f^i$. It is possible therefore to find the tangent base vectors to the deformed centroidal line at $C_i$ based on equation (4.2) and construct the orthonormal unit vectors of frame $x_f^i y_f^i z_f^i$ from them. Determining the elastic rotation matrix is then a simple matter.

The tangent base vector along the deformed $x_f^i$ axis is first found (Wemper [109]) via,

$$g_1 = \left.\frac{\partial \mathcal{P}_P}{\partial x}\right|_{\substack{y=0\\z=0}} = \{\underline{e}^i\}^T \left\{ \begin{array}{c} 1 + u' - y\,v'' - z\,w'' - \frac{1}{2}(v'^2 + w'^2) \\ v' - z\,\phi' \\ w' + y\,\phi' \end{array} \right\}\Bigg|_{\substack{y=0\\z=0}} \tag{4.3}$$

$$= \{\underline{e}^i\}^T \left\{ \begin{array}{c} 1 + u' - \frac{1}{2}(v'^2 + w'^2) \\ v' \\ w' \end{array} \right\} \tag{4.4}$$

In a similar manner, one has

$$g_2 = \left.\frac{\partial \mathcal{P}_P}{\partial y}\right|_{\substack{y=0\\z=0}} = \{\underline{e}^i\}^T \left\{ \begin{array}{c} -v' \\ 1 \\ \phi \end{array} \right\} \tag{4.5}$$

and

$$g_3 = \left.\frac{\partial \mathcal{P}_P}{\partial z}\right|_{\substack{y=0\\z=0}} = \{\underline{e}^i\}^T \left\{ \begin{array}{c} -w' \\ -\phi \\ 1 \end{array} \right\} \tag{4.6}$$

as the tangent base vectors along the deformed $y_f^i$ and $z_f^i$ axis respectively.

Now constructing unit vectors $\{\underline{e}^i_f\} = \begin{Bmatrix} \vec{\imath}^i_f \\ \vec{\jmath}^i_f \\ \hat{k}^i_f \end{Bmatrix}$ of frame $x^i_f y^i_f z^i_f$ from equations (4.4)–(4.6), one has

$$\vec{\imath}^i_f = \frac{g_1}{|g_1|} = \frac{\{\underline{e}^i\}^T}{g_1} \begin{Bmatrix} 1 + u' - \frac{1}{2}(v'^2 + w'^2) \\ v' \\ w' \end{Bmatrix} \tag{4.7}$$

$$\vec{\jmath}^i_f = \frac{g_2}{|g_2|} = \frac{\{\underline{e}^i\}^T}{g_2} \begin{Bmatrix} -v' \\ 1 \\ \phi \end{Bmatrix} \tag{4.8}$$

and

$$\hat{k}^i_f = \frac{g_3}{|g_3|} = \frac{\{\underline{e}^i\}^T}{g_3} \begin{Bmatrix} -w' \\ -\phi \\ 1 \end{Bmatrix} \tag{4.9}$$

Finally the rotation matrix from $x^i_f y^i_f z^i_f$ to body frame $X^i Y^i Z^i$ is obtained as

$$\mathcal{R}^i_f = \{\underline{e}^i\} \cdot \{\underline{e}^i_f\}^T = \begin{bmatrix} [1 + u' - \frac{1}{2}(v'^2 + w'^2)]/g_1 & -v'/g_2 & -w'/g_3 \\ v'/g_1 & 1/g_2 & -\phi/g_3 \\ w'/g_1 & \phi/g_2 & 1/g_3 \end{bmatrix} \tag{4.10}$$

As will be seen later, it suffices to settle for a second-order approximation of this rotation matrix for the present investigation. Starting with equations (4.4)–(4.6) and

using Taylor expansions, one has up to the second-order in the deformation variables

$$\frac{1}{g_1} = \frac{1}{\sqrt{[1+u'-\frac{1}{2}(v'^2+w'^2)]^2+v'^2+w'^2}} \stackrel{O(2)}{=} 1-u'+u'^2 \tag{4.11}$$

$$\frac{1}{g_2} = \frac{1}{\sqrt{1+v'^2+\phi^2}} \stackrel{O(2)}{=} 1-\frac{v'^2}{2}-\frac{\phi^2}{2} \tag{4.12}$$

$$\frac{1}{g_3} = \frac{1}{\sqrt{1+w'^2+\phi^2}} \stackrel{O(2)}{=} 1-\frac{w'^2}{2}-\frac{\phi^2}{2} \tag{4.13}$$

In view of these approximations, equation (4.10) becomes,

$$\mathcal{R}_f^i = \begin{bmatrix} [1+u'-\frac{1}{2}(v'^2+w'^2)](1-u'+u'^2) & -v'(1-\frac{v'^2}{2}-\frac{\phi^2}{2}) & -w'(1-\frac{w'^2}{2}-\frac{\phi^2}{2}) \\ v'(1-u'+u'^2) & 1-\frac{v'^2}{2}-\frac{\phi^2}{2} & -\phi(1-\frac{w'^2}{2}-\frac{\phi^2}{2}) \\ w'(1-u'+u'^2) & \phi(1-\frac{v'^2}{2}-\frac{\phi^2}{2}) & 1-\frac{w'^2}{2}-\frac{\phi^2}{2} \end{bmatrix}$$

$$\stackrel{O(2)}{=} \begin{bmatrix} 1-\frac{1}{2}(v'^2+w'^2) & -v' & -w' \\ v'-u'v' & 1-\frac{v'^2}{2}-\frac{\phi^2}{2} & -\phi \\ w'-u'w' & \phi & 1-\frac{w'^2}{2}-\frac{\phi^2}{2} \end{bmatrix} \tag{4.14}$$

The veracity of $\mathcal{R}_f^i$ as a rotation matrix can be examined by evaluating ${\mathcal{R}_f^i}^T\mathcal{R}_f^i$ as

$${\mathcal{R}_f^i}^T\mathcal{R}_f^i \stackrel{O(2)}{=} \begin{bmatrix} 1 & -u'v'+\phi w' & -\phi v'-u'w' \\ -u'v'+\phi w' & 1 & v'w' \\ -u'w'-\phi v' & v'w' & 1 \end{bmatrix} \neq [I] \tag{4.15}$$

i.e., ${\mathcal{R}_f^i}^T\mathcal{R}_f^i$ up to second order in the deformation variables does not evaluate to the unit matrix $[I]$.

This finding is disturbing since it at least shows that the $\mathcal{R}_f^i$ obtained from the assumed displacement field can not be exact to second order. What is more, it implies that despite the method used, including the numerical integration approach, the elastic

rotation matrix obtained based upon the assumed displacement field can not be correct even to the second order, much less exact. This is a new observation. In physical terms, this result indicates that the assumed displacement field given in equation (4.2) is not capable of describing the expected deformation state and thus fails to produce the expected orthogonal coordinate lines at $C_i$ in the deformed state, to the second order. From this analysis, it is also seen that the key to obtaining a consistent second and higher order elastic rotation matrices lies in the proper form of the assumed displacement field for the beam.

## 4.2 Complete $O(2)$ Displacement Field and Kinematic Terminal Equations

In this section, a complete $O(2)$ elastic rotation matrix is introduced by extending Pai and Nayfeh's approach [107]. (Pai and Nayfeh did not include the foreshortening in their approximation to the deformation displacement field of the beam, neither did they propose a systematic approach to construct a deformation displacement field that is complete up to any specified order in the deformation variables.) Then, this $O(2)$ elastic rotation matrix is utilized to construct a new complete $O(2)$ approximation to the Bernoulli-Euler beam's displacement field, followed by a discussion of the kinematic terminal equations for the beam. In the meantime, the terms missing from the traditional displacement for it to produce an $O(2)$ elastic rotation matrix are identified.

The reason why a complete $O(2)$ displacement field is needed is that it permits us to obtain system kinetic equations that are complete up to first order in the deformation variables of the beam. When kinetic equations complete up to higher orders are needed, a higher order displacement field is required. This is explained later in this chapter.

### 4.2.1 Complete $O(2)$ Elastic Rotation Matrix and Displacement Field

Referring to Fig.4.2, the rotation of the sectional frame of the beam can be divided into a sequence of two rotations, the rotation matrix of each of which is easy to find. The first rotation, called the bending rotation of the beam (Alkire [104]), is given by the angle $\phi$ between the undeformed centroidal axis and the deformed one. The other is simply the twist of the beam about $x_f^i$, the deformed centroidal axis, so as to bring the sectional frame into its final orientation.

From this breakdown of the elastic rotation and assuming the bending rotation is represented by matrix $B(\alpha)$, one has for the elastic rotation transformation matrix from the sectional frame to the body frame

$$\mathcal{R}_f^i = B^T(\alpha) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{4.16}$$

in which $\phi$ is the torsion of the section about its deformed centroidal axis $x_f^i$.

To evaluate $B(\alpha)$, it is assumed, as before, that a particle on the centroidal axis of the beam at any instant of time has a deformation displacement of $(u - \frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi,\ v,\ w)$, i.e., its current position can be expressed in the beam's reference frame $X^iY^iZ^i$ as

$$\begin{cases} X_c &= x + u - \frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi \\ Y_c &= v \\ Z_c &= w \end{cases} \tag{4.17}$$

which, on considering equations (4.4) and (4.11), gives the unit vector along its deformed

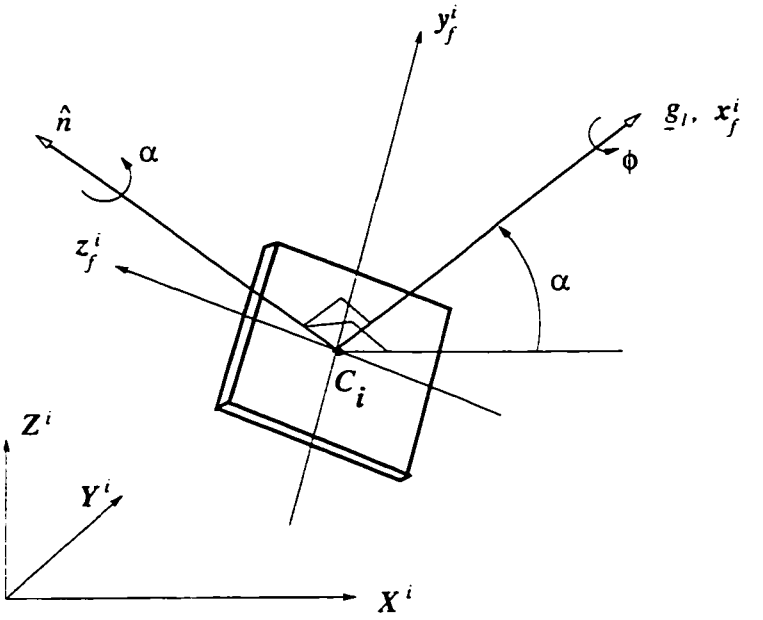



Figure 4.2: Two Euler Angles of Sectional Frame

line $x_f^i$ as

$$i_f^i = \frac{g_1}{|g_1|} = \frac{\{e^i\}^T}{\sqrt{[1+u'-\frac{1}{2}(v'^2+w'^2)]^2+v'^2+w'^2}} \begin{Bmatrix} 1+u'-\frac{1}{2}(v'^2+w'^2) \\ v' \\ w' \end{Bmatrix} \tag{4.18}$$

This expression defines $T_{11}$, $T_{12}$ and $T_{13}$ so that

$$i_f^i = \frac{g_1}{|g_1|} = \{e^i\}^T \begin{Bmatrix} T_{11} \\ T_{12} \\ T_{13} \end{Bmatrix} \tag{4.19}$$

and hence (Pai and Nayfeh [107]) by constructing the unit vector $\hat{n}$ (which is perpendicular to both $\mathbf{X}^i$ and $\mathbf{x}_f^i$ and evaluating the rotation transformation matrix for the angle of $\alpha$

about it, one has

$$B(\alpha) = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ -T_{12} & T_{11} + T_{13}^2/(1+T_{11}) & -T_{12}T_{13}/(1+T_{11}) \\ -T_{13} & -T_{12}T_{13}/(1+T_{11}) & T_{11} + T_{12}^2/(1+T_{11}) \end{bmatrix} \tag{4.20}$$

To the second order, this becomes

$$B(\alpha) \overset{O(2)}{=} \begin{bmatrix} 1 - \frac{1}{2}(v'^2 + w'^2) & v' - u'v' & w' - u'w' \\ -v' + u'v' & 1 - \frac{v'^2}{2} & -\frac{v'w'}{2} \\ -w' + u'w' & -\frac{v'w'}{2} & 1 - \frac{w'^2}{2} \end{bmatrix} \tag{4.21}$$

Therefore, the complete second order elastic rotation matrix of the cross sectional frame with respect to the body frame is

$$\begin{aligned} \mathcal{R}_f^i &= B(\alpha)^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \\ &\overset{O(2)}{=} \begin{bmatrix} 1 - \frac{1}{2}(v'^2 + w'^2) & -v' \underline{+u'v' - w'\phi} & -w' \underline{+u'w' + v'\phi} \\ v' - u'v' & 1 - \frac{v'^2}{2} - \frac{\phi^2}{2} & \underline{-\phi - \frac{v'w'}{2}} \\ w' - u'w' & \underline{\phi - \frac{v'w'}{2}} & 1 - \frac{w'^2}{2} - \frac{\phi^2}{2} \end{bmatrix} \end{aligned} \tag{4.22}$$

The underlined terms in equation (4.22) are the terms that are missing in equation (4.14) as far as the second-order elastic rotation matrix is concerned.

It can be verified that this rotation matrix is orthogonal to the second order, i.e.

$$\mathcal{R}_f^{i\,T}\mathcal{R}_f^i \overset{O(2)}{=} [I] \tag{4.23}$$

The missing terms will have a significant impact on the beam's dynamic equations when it undergoes large rigid body motion. A detailed discussion about the impact is left to later sections in this chapter, yet a brief conclusion is given here: those missing terms will cause some first-order inertial forces to be lost and some second-order terms to be lost in the elastic forces, both of which are important if an accurate and reliable prediction of the system behavior is to be obtained.

It is also noted that when the foreshortening term $-\frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi$ is not included in the beam's displacement field approximation, as in Pai and Nayfeh [107], the complete $O(2)$ elastic rotation matrix still takes the form of equation (4.22) except that now the $u$ term must be interpreted as the total axial displacement of a point on the beam's centroidal axis rather than the deformation caused purely by the stretch of the axis.

With the consistent complete second-order elastic rotation matrix in place, one is ready to construct the complete second-order displacement field for a Bernoulli-Euler beam as

$$\begin{aligned}
\underset{\sim}{\mathcal{P}}_P &= \underset{\sim}{r}_{A,C_i} + \underset{\sim}{r}_{C_i,P} \\
&= \{\underset{\sim}{e}^i\}^T \begin{Bmatrix} x + u - \frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi \\ v \\ w \end{Bmatrix} + \{\underset{\sim}{e}^i\}^T \mathcal{R}_f^i \begin{Bmatrix} 0 \\ y \\ z \end{Bmatrix}
\end{aligned}$$

$$= \{\underset{\sim}{e}^i\}^T \begin{Bmatrix} x + u - \frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi + (-v' + u'v' - w'\phi)y + (-w' + u'w' + v'\phi)z \\ v + (1 - \frac{v'^2}{2} - \frac{\phi^2}{2})y + (-\phi - \frac{v'w'}{2})z \\ w + (\phi - \frac{v'w'}{2})y + (1 - \frac{w'^2}{2} - \frac{\phi^2}{2})z \end{Bmatrix}$$

$$\triangleq \{\underset{\sim}{e}^i\}^T \{\mathcal{P}_P\} \tag{4.24}$$

where, $\underset{\sim}{r}_{A_i C_i}$ is the vector that starts at point $A_i$ and ends at $C_i$ (see Figure 4.1); $\underset{\sim}{r}_{C_i P}$ is the vector that starts at $C_i$ and ends at $P$; $u$, $v$ and $w$ are the pure axial deformation and bending deformations of the centroidal axis in the $Y^i$ and $Z^i$ directions, respectively; $\phi$ is the angle of twist with respect to the deformed centroidal line of the beam; $x$, $y$ and $z$ are the coordinates of particle $P$ in the body frame before deformation. Note that the pre-deformation coordinates $x$, $y$ and $z$ of the point $P$ are used because of the un-deformable cross-section assumption made of the beam.

This displacement field along with its attendant consistent elastic rotation matrix of equation (4.22) represents a set of exact second-order kinematic equations for a Bernoulli-Euler beam. As will be explained later in this chapter, employment of this set of second-order kinematic equations recovers all first-order inertial terms and all elastic force terms up to second order in multibody system kinetic equations.

It is important to keep in mind that, in essence, what distinguishes the proposed complete second-order deformation displacement field of equation (4.24) from the traditional one of equation (4.2) is the use of the new complete second-order elastic rotation matrix of (4.22) in (4.24) as opposed to the use of the first-order elastic rotation matrix to obtain (4.2).

To facilitate future discussions, it is also useful to point out that the displacement

field (4.24) and the elastic rotation (4.22) each can be recast into the following forms:

$$\underline{\mathcal{P}}_P = \underline{\mathcal{P}}_{P0} + \underline{\mathcal{P}}_{P1} + \underline{\mathcal{P}}_{P2} \tag{4.25}$$

and

$$\mathcal{R}^i_f = \mathcal{R}^i_{f_0} + \mathcal{R}^i_{f_1} + \mathcal{R}^i_{f_2} \tag{4.26}$$

where the subscripts 0, 1 and 2 denote the zero-th, the first- and the second-order terms of the respective expressions in the elastic deformation variables $u$, $v$, $w$ and $\phi$, and where $\{\underline{e}\} = \begin{Bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{Bmatrix}$ is the unit vector column matrix of the global frame, and $\mathcal{R}^i_1$ is the rotation transformation matrix of the beam's reference frame $X^iY^iZ^i$ with respect to the global inertial frame $XYZ$.

$$\underline{\mathcal{P}}_{P0} = \{\underline{e}^i\}^T \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \{\underline{e}\}^T \mathcal{R}^i_1 \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \tag{4.27}$$

$$\underline{\mathcal{P}}_{P1} = \{\underline{e}^i\}^T \begin{Bmatrix} u - v'y - w'z \\ v - \phi z \\ w + \phi y \end{Bmatrix} = \{\underline{e}\}^T \mathcal{R}^i_1 \begin{Bmatrix} u - v'y - w'z \\ v - \phi z \\ w + \phi y \end{Bmatrix} \tag{4.28}$$

$$\underline{\mathcal{P}}_{P2} = \{\underline{e}^i\}^T \begin{Bmatrix} -\frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi + (u'v' - w'\phi)y + (u'w' + v'\phi)z \\ -(\frac{v'^2}{2} + \frac{\phi^2}{2})y - (\frac{v'w'}{2})z \\ -(\frac{v'w'}{2})y - (\frac{w'^2}{2} + \frac{\phi^2}{2})z \end{Bmatrix}$$

$$= \{\underline{e}\}^T \mathcal{R}_1^i \begin{Bmatrix} -\frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi + (u'v' - w'\phi)y + (u'w' + v'\phi)z \\ -(\frac{v'^2}{2} + \frac{\phi^2}{2})y - (\frac{v'w'}{2})z \\ -(\frac{v'w'}{2})y - (\frac{w'^2}{2} + \frac{\phi^2}{2})z \end{Bmatrix} \quad (4.29)$$

$$\mathcal{R}_{f_0}^i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.30)$$

$$\mathcal{R}_{f_1}^i = \begin{bmatrix} 0 & -v' & -w' \\ v' & 0 & -\phi \\ w' & \phi & 0 \end{bmatrix} \quad (4.31)$$

$$\mathcal{R}_{f_2}^i = \begin{bmatrix} -\frac{1}{2}(v'^2 + w'^2) & u'v' - w'\phi & u'w' + v'\phi \\ -u'v' & -\frac{v'^2}{2} - \frac{\phi^2}{2} & -\frac{v'w'}{2} \\ -u'w' & -\frac{v'w'}{2} & -\frac{w'^2}{2} - \frac{\phi^2}{2} \end{bmatrix} \quad (4.32)$$

### 4.2.2 Kinematic Terminal Equations

Kinematic terminal equations of a flexible arm in a beam, as defined in Chapter 3, involve the position vector, with respect to the origin of the body frame, of the point that defines the flexible arm (i.e., $\underline{\mathcal{P}}_P \triangleq \underline{r}_{A_i C_i} + \underline{r}_{C_i P}$), the first and second time derivative of the position vector $\underline{\mathcal{P}}_P$ in the global frame. Although these vectors can be expressed in any reference frame, they are expressed in the global inertial frame in the current study. Kinematic terminal equations also involve the elastic rotation of a sectional frame and its angular velocity and acceleration relative to the body frame. The rotational angular velocity and acceleration are also expressed in the global inertial frame.

With the displacement field and elastic rotation matrix found earlier, obtaining these terminal equations is a simple matter. To start, let us assume that the flexible arm starts from the origin of the body frame and ends at point $P$, and it has $\{\mathcal{P}_P\}$ as its coordinate column matrix in the body frame, see equation (4.24). It is also assumed that the body frame of the beam has a rotation matrix of $\mathcal{R}_1^i$ and an angular velocity of $\underset{\sim}{\omega}_1^i$, all with respect to the global frame. Thus, the terminal equation at the displacement level for the flexible arm is

$$\underset{\sim}{r}_2 = \{\underset{\sim}{e}\}^T \mathcal{R}_1^i \{\mathcal{P}_P\} \tag{4.33}$$

where $\underset{\sim}{e}$ is the unit vector column matrix of the global frame, as defined earlier.

The terminal equation at the velocity level is

$$\dot{\underset{\sim}{r}}_2 = \{\underset{\sim}{e}\}^T \frac{d(\mathcal{R}_1^i \{\mathcal{P}_P\})}{dt} \tag{4.34}$$

and the terminal equation at the acceleration level is

$$\ddot{\underset{\sim}{r}}_2 = \{\underset{\sim}{e}\}^T \frac{d^2(\mathcal{R}_1^i \{\mathcal{P}_P\})}{dt^2} \tag{4.35}$$

The terminal equation for elastic rotation is simply the elastic rotation matrix of the section on which the point is located, that is,

$$\mathcal{R}_2 = \mathcal{R}_f^i \tag{4.36}$$

The terminal equation at the angular velocity level is

$$\underset{\sim}{\omega}_2 = \{\underset{\sim}{e}\}^T \mathcal{R}_1^i \begin{Bmatrix} \omega_{f1} \\ \omega_{f2} \\ \omega_{f3} \end{Bmatrix} \tag{4.37}$$

where $\omega_{f1}$, $\omega_{f2}$ and $\omega_{f3}$ are found (see Section 3.4) with

$$\frac{d\mathcal{R}_f^i}{dt}\mathcal{R}_f^{i\,T} = \begin{bmatrix} 0 & -\omega_{f3} & \omega_{f2} \\ \omega_{f3} & 0 & -\omega_{f1} \\ -\omega_{f2} & \omega_{f1} & 0 \end{bmatrix} \triangleq \tilde{\omega}_f \tag{4.38}$$

which is the skew-symmetric matrix of the angular velocity of the sectional frame relative to the body frame and expressed in the body frame as well. The terminal equation at the angular acceleration level is obtained as

$$\dot{\underline{\omega}}_2 = \{\underline{e}\}^T(\dot{\mathcal{R}}_1^i \begin{Bmatrix} \omega_{f1} \\ \omega_{f2} \\ \omega_{f3} \end{Bmatrix} + \mathcal{R}_1^i \begin{Bmatrix} \dot{\omega}_{f1} \\ \dot{\omega}_{f2} \\ \dot{\omega}_{f3} \end{Bmatrix}) \tag{4.39}$$

The expressions for $\delta\underline{r}_2$ and $\delta\underline{\theta}_2$ can be obtained from equations (4.34) and (4.37) respectively by substituting the time derivatives of system coordinates (including both rigid and elastic ones) with their respective variations and discarding any terms that are not functions of these coordinates.

To conclude this section, for those dynamicists who feel more comfortable with the inextensibility assumption of beams, $u = u' = 0$ can be used in equations (4.22) and (4.24) to obtain the corresponding complete $O(2)$ kinematic equations for the beam. The results are correct due to the early separation of the axial displacement of a generic point on the centroidal line into two parts: the actual axial deformation of the point denoted as $u$ and the axial displacement of the point due to the inextensibility assumption, or the foreshortening effect, as it is often called, represented by $-\frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi$. A note of caution though. This inextensibility assumption might have bigger consequences on multibody dynamics simulation than on a single beam system. Some discussions about

this will be made in the following section.

## 4.3 Virtual Work Terminal Equation

As presented earlier in Section 3.4, the virtual work terminal equation for a flexible body is defined as the negative of the virtual work done by the internal elastic forces of the body and by the body forces, including the inertial forces of the body. In this section, it is discussed how to evaluate these terms for a Bernoulli-Euler beam given the displacement field proposed in the last section.

### 4.3.1 Virtual Work of Body Forces

In this subsection an approach is first presented for finding the virtual work of the inertial force system of the beam and why a complete $O(2)$ deformation approximation should be utilized. Subsequently a formula is given for calculating the virtual work of the gravitational force acting on the beam.

Recall that the virtual work of the inertial force system of a generic body was defined in Section 3.4 as

$$-\int_V \delta\underset{\sim}{r}_P \cdot \ddot{\underset{\sim}{r}}_P \gamma dV \tag{4.40}$$

where $\underset{\sim}{r}_P$ is the absolute position vector of a generic differential volume element of the body with respect to the global frame, and $\gamma$ is the mass per unit volume. Once this quantity has been found, one can substitute it and carry out the volume integration. Instead of performing the volume integration, a simpler procedure can be pursued for the same purpose.

From the basic kinematic assumptions made earlier, a Bernoulli-Euler beam can be regarded as being composed of a series of thin differential slices cut with parallel planes, and each of the slices can be taken as rigid, see Figure 4.3. Hence, one has for the $i$-th

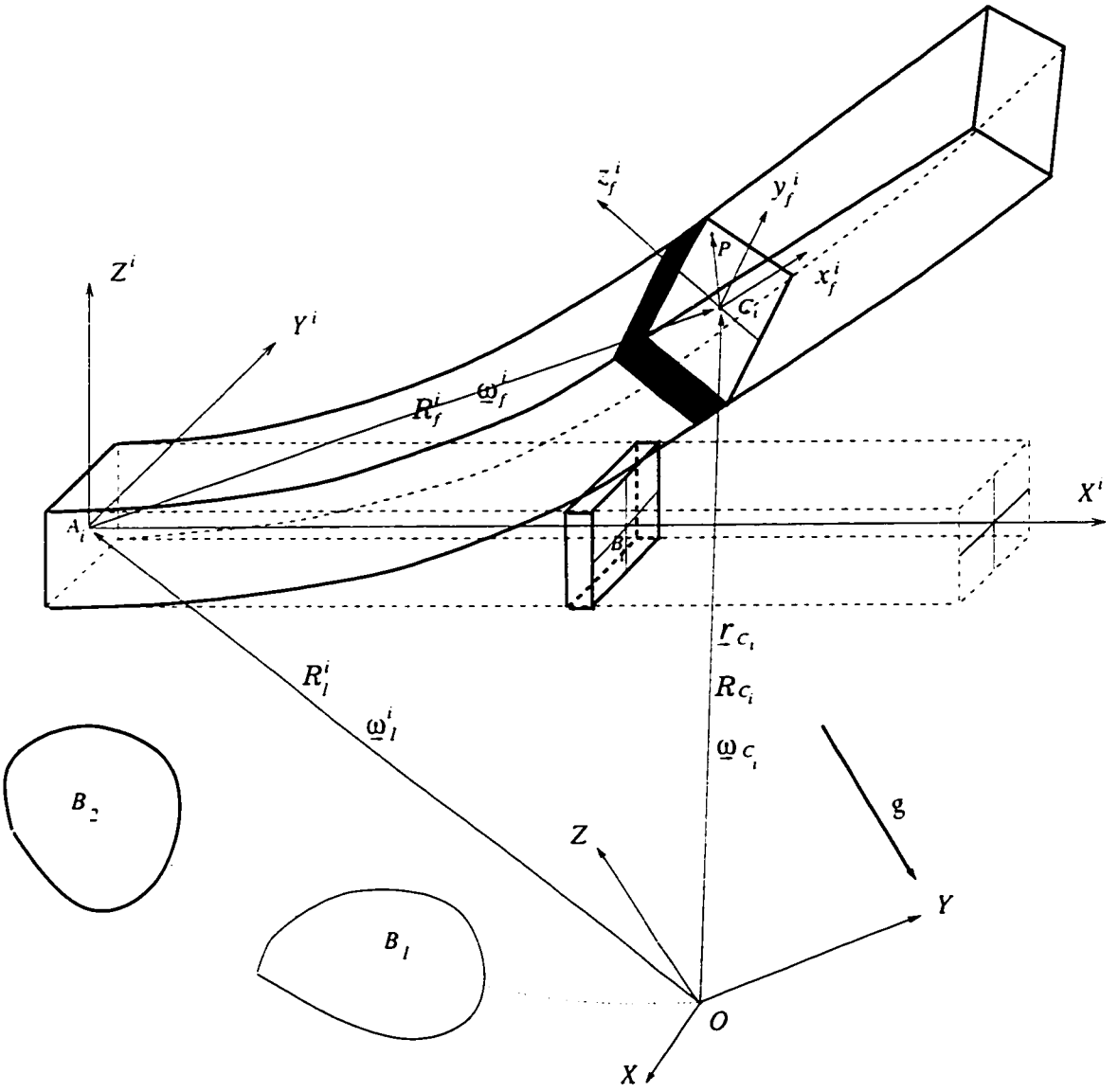



Figure 4.3: The Flexible Beam and Its Differential Slices

body

$$-\int_V \delta\underline{r}_P \cdot \gamma \ddot{\underline{r}}_P dV = -\int_0^{l_i} \left\{ \gamma_i \, \ddot{\underline{r}}_{C_i} \cdot \delta\underline{r}_{C_i} + \left[ d\underline{\mathbf{I}}_i \cdot \dot{\underline{\omega}}_{C_i} + \underline{\omega}_{C_i} \times (d\underline{\mathbf{I}}_i \cdot \underline{\omega}_{C_i}) \right] \cdot \delta\underline{\theta}_{C_i} \right\} dx \qquad (4.41)$$

in which $\gamma_i$ is the mass per unit length of the beam; $\ddot{\underline{r}}_{C_i}$ and $\delta\underline{r}_{C_i}$ are the absolute acceleration and virtual displacement of a particle on the beam's centroidal line, respectively; $d\underline{\mathbf{I}}_i$ is the inertia dyadic of the beam slice per unit length, and

$$d\mathbf{I}_i = \begin{bmatrix} \gamma(I_{yy} + I_{zz}) & 0 & 0 \\ 0 & \gamma I_{yy} & 0 \\ 0 & 0 & \gamma I_{zz} \end{bmatrix} \qquad (4.42)$$

in which $I_{yy}$ and $I_{zz}$ are the second moment of area of the cross section about the $y_f^i$ and $z_f^i$ axis respectively; $\dot{\underline{\omega}}_{C_i}$, $\underline{\omega}_{C_i}$ and $\delta\underline{\theta}_{C_i}$ are respectively the angular acceleration, velocity and virtual rotation of the slice. The integration is to be carried out along the undeformed length of the beam (Wempner [109]).

With this reformulation, the calculation of the virtual work of the inertial forces requires only finding the absolute kinematic quantities of a differential slice, substituting them and carrying out the above integration.

To find the required kinematic quantities for the above integration, one makes use of the circuit matrix of the system graph and the kinematic terminal equations of the elements involved in the system. This technique is well established and has been used in all the papers that employ the graph-theoretic approach.

What is worth further investigation though, is the impact on the system kinetic equations of employing the complete second-order deformation displacement field proposed in the previous section as opposed to the traditional first-order field. To achieve this end,

let us take a system of a single particle with mass $m$ as a simple case. It is assumed that the particle's position vector with respect to the inertial frame, up to second order in the small coordinate $q(t)$, is expressed as

$$\underline{r}(q,t) = \underline{r}_0(t) + \underline{r}_1 q(t) + \underline{r}_2 q^2(t) \tag{4.43}$$

in which $q(t)$ is a single generalized coordinate for the particle, $\underline{r}_0(t)$ can be a function of some large coordinates or some motion driver function, and $\underline{r}_1$ and $\underline{r}_2$ are not functions of time. Thus, one has

$$\delta\underline{r} = \underline{r}_1\delta q + 2\underline{r}_2 q\delta q \tag{4.44}$$

$$\ddot{\underline{r}} = \ddot{\underline{r}}_0 + \underline{r}_1\ddot{q} + 2\underline{r}_2(\dot{q}^2 + q\ddot{q}) \tag{4.45}$$

and the virtual work of its inertial force is

$$\begin{aligned} -\delta\underline{r}\cdot(m\ddot{\underline{r}}) &= -m(\underline{r}_1\delta q + 2\underline{r}_2 q\delta q)\cdot\left[\ddot{\underline{r}}_0 + \underline{r}_1\ddot{q} + 2\underline{r}_2(\dot{q}^2 + q\ddot{q})\right] \\ &= -m\Big[\underline{r}_1\cdot\ddot{\underline{r}}_0 + \underline{r}_1\cdot\underline{r}_1\ddot{q} + 2\underline{r}_2\cdot\ddot{\underline{r}}_0 q \\ &\quad + 2\underline{r}_1\cdot\underline{r}_2(\dot{q}^2 + 2q\ddot{q}) + 4\underline{r}_2\cdot\underline{r}_2 q(\dot{q}^2 + 2q\ddot{q})\Big]\delta q \qquad (4.46) \\ &\triangleq \mathcal{F}_I\delta q \qquad (4.47) \end{aligned}$$

From this simple development, one sees that if the underlined second-order term in the particle's displacement approximation (4.43) had been dropped, the inertial force $\mathcal{F}_I$ corresponding to the generalized coordinate $q$ would have lost the underlined first-order term in $q$.

The same problem would occur if the traditional first-order displacement field of a Bernoulli-Euler beam were used. To show this, let us take a look at the the first term in

equation (4.41), i.e., $\ddot{\underline{r}}_{C_i} \cdot \delta\underline{r}_{C_i}$ and let the deformation variables be discretized, using the Rayleigh-Ritz method over the entire length of the beam, as

$$\begin{Bmatrix} u(x,t) \\ v(x,t) \\ w(x,t) \\ \phi(x,t) \end{Bmatrix} = \begin{bmatrix} \mathbf{N}_u^T(x) & 0 & 0 & 0 \\ 0 & \mathbf{N}_v^T(x) & 0 & 0 \\ 0 & 0 & \mathbf{N}_w^T(x) & 0 \\ 0 & 0 & 0 & \mathbf{N}_\phi^T(x) \end{bmatrix} \begin{Bmatrix} \mathbf{q}_u(t) \\ \mathbf{q}_v(t) \\ \mathbf{q}_w(t) \\ \mathbf{q}_\phi(t) \end{Bmatrix} \triangleq \mathbf{N}\mathbf{q}_f \tag{4.48}$$

where $\mathbf{N}_u(x)$, $\mathbf{N}_v(x)$, $\mathbf{N}_w(x)$, and $\mathbf{N}_\phi(x)$ are the columns of shape functions, and $\mathbf{q}_u(t)$, $\mathbf{q}_v(t)$, $\mathbf{q}_w(t)$, and $\mathbf{q}_\phi(t)$ the columns of generalized coordinates, for $u$, $v$, $w$, and $\phi$ respectively. Although a variety of shape functions can be utilized for $\mathbf{N}_u(x)$, $\mathbf{N}_v(x)$, $\mathbf{N}_w(x)$, and $\mathbf{N}_\phi(x)$ (Meirovitch and Kwak [91] and [92]), the Taylor, Chebyshev, and Legendre polynomials (Irving and Mullineux [89]) are employed in the current investigation.

Referring to Figure 4.3, one has

$$\underline{r}_{C_i} = \underline{r}_{OA_i}(\mathbf{q}_r) + \underline{\mathcal{P}}_P\Big|_{\substack{y=0 \\ z=0}} \tag{4.49}$$

where $\underline{r}_{OA_i}(\mathbf{q}_r)$ denotes that vector $\underline{r}_{OA_i}$ is a function of $\mathbf{q}_r$, the generalized coordinates used to specify the rigid body configuration of the beam's reference frame. These generalized coordinates may contain small elastic coordinates other than those for the current beam, but they must contain large rigid coordinates for multibody systems.

Considering equation (4.25), this last equation becomes

$$\underline{r}_{C_i} = \underline{r}_{OA_i}(\mathbf{q}_r) + \underline{\mathcal{P}}_{C_i 0} + \underline{\mathcal{P}}_{C_i 1} + \underline{\mathcal{P}}_{C_i 2} \tag{4.50}$$

in which

$$\underline{\mathcal{P}}_{C,0} = \{\underline{e}\}^T \mathcal{R}_1^i \begin{Bmatrix} x \\ 0 \\ 0 \end{Bmatrix} \tag{4.51}$$

$$\underline{\mathcal{P}}_{C,1} = \{\underline{e}\}^T \mathcal{R}_1^i \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{4.52}$$

$$\underline{\mathcal{P}}_{C,2} = \{\underline{e}\}^T \mathcal{R}_1^i \begin{Bmatrix} -\frac{1}{2}\int_0^x (v'^2 + w'^2) d\xi \\ 0 \\ 0 \end{Bmatrix} \tag{4.53}$$

Again, let

$$\underline{r}_{C,0} = \underline{r}_{OA_i}(\mathbf{q}_r) + \underline{\mathcal{P}}_{C,0} \tag{4.54}$$

$$\underline{r}_{C,1} = \underline{\mathcal{P}}_{C,1} \tag{4.55}$$

$$\underline{r}_{C,2} = \underline{\mathcal{P}}_{C,2} \tag{4.56}$$

Then, considering equations (4.48), one has

$$\underline{r}_{C_i} = \underline{r}_{C,0}(\mathbf{q}_r) + \underline{r}_{C,1} + \underline{r}_{C,2} \tag{4.57}$$

$$= \underline{r}_{C,0}(\mathbf{q}_r) + \underline{\mathbf{J}}_1 \mathbf{q}_f + \underline{\mathbf{q}_f^T \underline{\mathbf{J}}_2 \mathbf{q}_f} \tag{4.58}$$

since $\underline{r}_{C,1}$ and $\underline{r}_{C,2}$ represent the first- and second-order (quadratic) terms of the position vector $\underline{r}_{C_i}$ in the deformation variables. $\underline{\mathbf{J}}_1$ is a row matrix of vectors, $\underline{\mathbf{J}}_2$ is a square

matrix of vectors, both of which are independent of $\mathbf{q}_f$, and $\underline{\mathbf{J}}_2 = \underline{\mathbf{J}}_2^T$. Thus,

$$\dot{\underline{r}}_{C_i} = \underline{\mathbf{A}}_r \dot{\mathbf{q}}_r + \underline{\mathbf{J}}_1 \dot{\mathbf{q}}_f + \underline{2\, \mathbf{q}_f^T \underline{\mathbf{J}}_2\, \dot{\mathbf{q}}_f} \tag{4.59}$$

$$\ddot{\underline{r}}_{C_i} = \ddot{\underline{r}}_{C_i,0} + \ddot{\underline{r}}_{C_i,1} + \ddot{\underline{r}}_{C_i,2} \tag{4.60}$$

in which $\underline{A}_r = \left[ \frac{\partial \underline{r}_{C_i,0}}{\partial q_{r1}}, \frac{\partial \underline{r}_{C_i,0}}{\partial q_{r2}}, \cdots, \frac{\partial \underline{r}_{C_i,0}}{\partial q_{r,N_r}} \right]$, with $N_r$ being the number of coordinates in $\mathbf{q}_r$, and

$$\delta \underline{r}_{C_i} = \underline{\mathbf{A}}_r \delta \mathbf{q}_r + \underline{\mathbf{J}}_1 \delta \mathbf{q}_f + \underline{2\, \mathbf{q}_f^T \underline{\mathbf{J}}_2 \delta \mathbf{q}_f} \tag{4.61}$$

Therefore,

$$\begin{aligned}
\ddot{\underline{r}}_{C_i} \cdot \delta \underline{r}_{C_i} &= \ddot{\underline{r}}_{C_i,0} \cdot \underline{\mathbf{A}}_r \delta \mathbf{q}_r + \ddot{\underline{r}}_{C_i,0} \cdot \underline{\mathbf{J}}_1 \delta \mathbf{q}_f && \longleftarrow \text{0-th} \\
&+ \ddot{\underline{r}}_{C_i,1} \cdot \underline{\mathbf{A}}_r \delta \mathbf{q}_r + \ddot{\underline{r}}_{C_i,1} \cdot \underline{\mathbf{J}}_1 \delta \mathbf{q}_f + \underline{2\, \ddot{\underline{r}}_{C_i,0} \cdot \mathbf{q}_f^T \underline{\mathbf{J}}_2 \delta \mathbf{q}_f} && \longleftarrow \text{1-st} \\
&+ \ddot{\underline{r}}_{C_i,2} \cdot \underline{\mathbf{A}}_r \delta \mathbf{q}_r + \ddot{\underline{r}}_{C_i,2} \cdot \underline{\mathbf{J}}_1 \delta \mathbf{q}_f + 2\, \ddot{\underline{r}}_{C_i,1} \cdot \mathbf{q}_f^T \underline{\mathbf{J}}_2 \delta \mathbf{q}_f && \longleftarrow \text{2-nd} \\
&+ 2\, \ddot{\underline{r}}_{C_i,2} \cdot \mathbf{q}_f^T \underline{\mathbf{J}}_2 \delta \mathbf{q}_f && \longleftarrow \text{3-rd}
\end{aligned} \tag{4.62}$$

It is clear now that the underlined first order term in the above equation would have been lost if the second-order displacement approximation had not been used in equation (4.48), i.e., the $\mathcal{P}_{C_i,2}$ term had not been included in it. This also means that if a first-order deformation displacement field approximation is employed instead of a complete second-order one, certain first-order terms will be lost in the beam's inertial forces.

Examining the second rotational term of equation (4.41) in a similar manner in conjuction with equation (4.26) will lead to the same conclusion.

Thus it is concluded that use of a first-order displacement field approximation in calculating inertial forces will cause some first-order terms to be lost in the system kinetic equations. The remedy is to use a second-order approximation of the displacement field

and, if higher order terms in inertial forces are required, a displacement approximation of one order higher must be used. The two-Euler angle approach introduced earlier in this chapter provides a systematic way to devise such an approximation to an arbitrary order. To do this, one only needs to include the foreshortening up to the desired order in the deformation variables in equation (4.17), expand $\mathbf{B}(\alpha)$ of equation (4.21) to the desired order, which in turn gives the $\mathcal{R}_f^i$ up to the desired order, and substitute them into the first of equation (4.24).

For the virtual work of the gravitational force acting on the beam, referring to Figure 4.3 one has

$$-\int_0^{l_i} \gamma_i g \left\{\delta r_{C_i}\right\}_3 dx \tag{4.63}$$

where $g$ is the gravitational acceleration acting in the negative direction of the global $Z$ axis, and $\{\delta r_{C_i}\}_3$ is the third ($Z$) component of $\delta \underline{r}_{C_i}$.

The virtual work of the inertial force system and the gravity discussed above is calculated automatically in symbolic form by DynaFlex using equations (4.41) and (4.63) respectively, with the deformation variables $u$, $v$, $w$, and $\phi$ discretized according to equation (4.48).

### 4.3.2 Virtual Work of Elastic Forces

In this subsection, the virtual work of the internal elastic forces of a Bernoulli-Euler beam is evaluated using the complete second-order deformation displacement approximation proposed earlier in this section, to complete the virtual work terminal equation for the beam. Owing to the use of this approximation of the deformation field, all second-order terms in the deformation variables in the elastic forces are recovered. The terms that would have been missed if the non-complete second order deformation field had been used will be identified. Non-linear strain-displacement relations are employed, but only

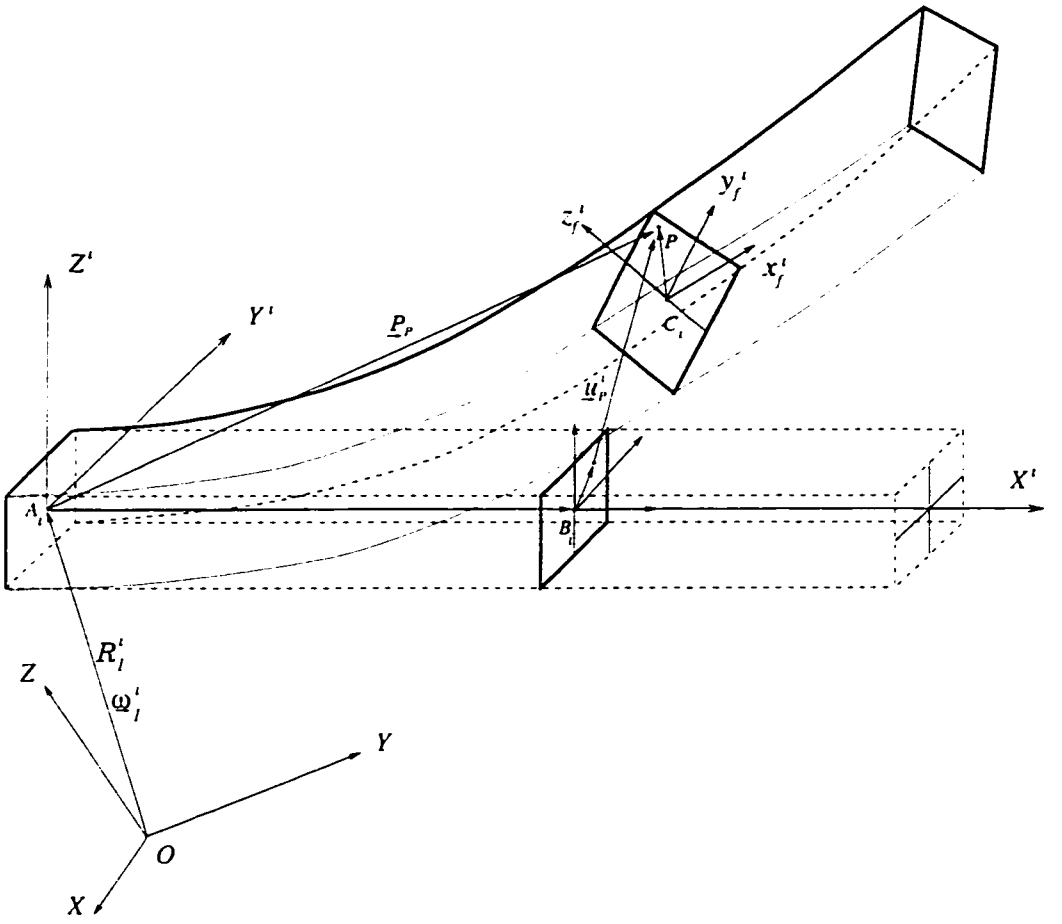



Figure 4.4: The Deformation Vector $\underline{u}_P^i$ in a Bernoulli-Euler Beam

terms up to the third-order in the deformation variables are retained in the virtual work expression for the present investigation.

### The Complete Second-Order Deformation Field

Earlier in this chapter, the complete second-order deformation displacement field of a Bernoulli-Euler beam was put forth in equation (4.24). Using this equation, the defor-

mation vector of a particle in the beam is found to be (see Figure 4.4)

$$
\begin{aligned}
\underline{u}_P^i &\triangleq \{\underline{e}^i\}^T \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} \\
&= \underline{\mathcal{P}}_P - \underline{r}_{A_i B_i} - \{\underline{e}^i\}^T \begin{Bmatrix} 0 \\ y \\ z \end{Bmatrix} \\
&= \underline{\mathcal{P}}_P - \{\underline{e}^i\}^T \begin{Bmatrix} x \\ 0 \\ 0 \end{Bmatrix} - \{\underline{e}^i\}^T \begin{Bmatrix} 0 \\ y \\ z \end{Bmatrix} \\
&= \{\underline{e}^i\}^T \begin{Bmatrix} u - \frac{1}{2}\int_0^x (v'^2 + w'^2)d\xi + (-v' + u'v' - w'\phi)y + (-w' + u'w' + v'\phi)z \\ v + (-\frac{v'^2}{2} - \frac{\phi^2}{2})y + (-\phi - \frac{v'w'}{2})z \\ w + (\phi - \frac{v'w'}{2})y + (-\frac{w'^2}{2} - \frac{\phi^2}{2})z \end{Bmatrix}
\end{aligned}
\tag{4.64}
$$

which allows us to compute the strains of the beam, and hence the virtual work of the elastic forces.

### The Non-Linear Strains

The non-linear strain-displacement relations to be used to find the virtual work read as follows (Shames [101])

$$\left\{\begin{array}{lcl} \varepsilon_{xx} & = & \dfrac{\partial u_x}{\partial x}+\dfrac{1}{2}\left[(\dfrac{\partial u_x}{\partial x})^2+(\dfrac{\partial u_y}{\partial x})^2+(\dfrac{\partial u_z}{\partial x})^2\right] \\ \varepsilon_{xy} & = & \dfrac{1}{2}\left(\dfrac{\partial u_x}{\partial y}+\dfrac{\partial u_y}{\partial x}+\dfrac{\partial u_x}{\partial x}\dfrac{\partial u_x}{\partial y}+\dfrac{\partial u_y}{\partial x}\dfrac{\partial u_y}{\partial y}+\dfrac{\partial u_z}{\partial x}\dfrac{\partial u_z}{\partial y}\right) \\ \varepsilon_{xz} & = & \dfrac{1}{2}\left(\dfrac{\partial u_x}{\partial z}+\dfrac{\partial u_z}{\partial x}+\dfrac{\partial u_x}{\partial x}\dfrac{\partial u_x}{\partial z}+\dfrac{\partial u_y}{\partial x}\dfrac{\partial u_y}{\partial z}+\dfrac{\partial u_z}{\partial x}\dfrac{\partial u_z}{\partial z}\right) \end{array}\right. \tag{4.65}$$

where $\varepsilon_{xx}$ is the normal strain in the $X^i$ direction, $\varepsilon_{xy}$ is the shear strain in a plane parallel to the $X^iA_iY^i$ plane, and $\varepsilon_{xz}$ is the shear strain in a plane parallel to the $X^iA_iZ^i$ plane. Other strain components are very small and assumed to be zero for Bernoulli-Euler beams.

Using equation (4.64), it is first computed

$$\begin{aligned} \frac{\partial u_x}{\partial x} &= u' - \frac{1}{2}(v'^2+w'^2)+(-v''+u''v'+u'v''-w''\phi-w'\phi')y \\ &\quad +(-w''+u''w'+u'w''+v''\phi+v'\phi')z && (4.66)\\ \frac{\partial u_x}{\partial y} &= -v'+u'v'-w'\phi && (4.67)\\ \frac{\partial u_x}{\partial z} &= -w'+u'w'+v'\phi && (4.68)\\ \frac{\partial u_y}{\partial x} &= v'+(-\phi\phi'-v'v'')y+(-\phi'-\frac{v''w'+v'w''}{2})z && (4.69)\\ \frac{\partial u_y}{\partial y} &= -\frac{1}{2}(\phi^2+v'^2) && (4.70)\\ \frac{\partial u_y}{\partial z} &= -\phi-\frac{v'w'}{2} && (4.71)\\ \frac{\partial u_z}{\partial x} &= w'+(\phi'-\frac{v''w'+v'w''}{2})y+(-\phi\phi'-w'w'')z && (4.72)\\ \frac{\partial u_z}{\partial y} &= \phi-\frac{v'w'}{2} && (4.73) \end{aligned}$$

$$\frac{\partial u_z}{\partial z} = -\frac{1}{2}(\phi^2 + w'^2) \tag{4.74}$$

Thus, one has

$$\begin{aligned}
\varepsilon_{xx} = {} & u' - \frac{1}{2}(v'^2 + w'^2) + (-v'' \underline{+u''v' + u'v'' - w''\phi - w'\phi'})y \\
& +(-w'' \underline{+u''w' + u'w'' + v''\phi + v'\phi'})z \\
& +\frac{1}{2}\{[u' - \frac{1}{2}(v'^2 + w'^2) + (-v'' + u''v' + u'v'' - w''\phi - w'\alpha')y \\
& +(-w'' + u''w' + u'w'' + v''\phi + v'\phi')z]^2 \\
& +[v' + (-\phi\phi' - v'v'')y + (-\phi' - \frac{v''w' + v'w''}{2})z]^2 \\
& +[w' + (\phi' - \frac{v''w' + v'w''}{2})y + (-\phi\phi' - w'w'')z]^2\}
\end{aligned} \tag{4.75}$$

Note the underlined terms are due to the use of the complete second-order approximation of the deformation field, i.e., the complete second-order elastic rotation.

For elastic forces of up to third order in deformation variables, terms higher in order than second in the strains can be ignored. Equation (4.75) then reduces to

$$\begin{aligned}
\varepsilon_{xx} \overset{O(2)}{=} {} & u' - \frac{1}{2}(v'^2 + w'^2) \\
& +(-v'' \underline{+u''v' + u'v'' - w''\phi - w'\phi'})y + (-w'' \underline{+u''w' + u'w'' + v''\phi + v'\phi'})z \\
& +\frac{1}{2}\{[u' - v''y - w''z]^2 + [v' - \phi'z]^2 + [w' + \phi'y]^2\} \\
= {} & u' + \frac{u'^2}{2} + (-v'' \underline{+u''v' + u'v'' - w''\phi - w'\phi'} - u'v'' + w'\phi')y \\
& +(-w'' \underline{+u''w' + u'w'' + v''\phi + v'\phi'} - u'w'' - v'\phi')z \\
& +\frac{1}{2}(v''y + w''z)^2 + \frac{\phi'^2}{2}z^2 + \frac{\phi'^2}{2}y^2
\end{aligned} \tag{4.76}$$

$$\begin{aligned}
= {} & u' + \frac{1}{2}u'^2 + (-v'' \underline{+u''v' - w''\phi})y + (-w'' \underline{+u''w' + v''\phi})z \\
& +\frac{1}{2}(v''y + w''z)^2 + \frac{\phi'^2}{2}z^2 + \frac{\phi'^2}{2}y^2
\end{aligned} \tag{4.77}$$

Let

$$\begin{cases} H_{11} &= u' + \frac{u'^2}{2} \\ H_{12} &= -v'' + \underline{u''v' - w''\phi} \\ H_{13} &= -w'' + \underline{u''w' + v''\phi} \\ H_{14} &= \frac{1}{2}(v''y + w''z)^2 + \frac{\phi'^2}{2}y^2 + \frac{\phi'^2}{2}z^2 \end{cases} \tag{4.78}$$

Thus, equation (4.77) takes the following form

$$\varepsilon_{xx} = H_{11} + yH_{12} + zH_{13} + H_{14} \tag{4.79}$$

Further, if one lets

$$\begin{cases} a_1 &= v''^2 + \phi'^2 \\ a_2 &= w''^2 + \phi'^2 \\ a_3 &= 2v''w'' \end{cases} \tag{4.80}$$

then

$$H_{14} = \frac{1}{2}(a_1y^2 + a_2z^2 + a_3yz) \tag{4.81}$$

In the same way, one finds $\varepsilon_{xy}$ as

$$\begin{aligned} 2\varepsilon_{xy} &= -v' + \underline{u'v' - w'\phi} + v' + (\underline{-\phi\phi' - v'v''})y + (-\phi' - \underline{\frac{v''w' + v'w''}{2}})z \\ &+[u' - \frac{1}{2}(v'^2 + w'^2) + (-v'' + u''v' + u'v'' - w''\phi - w'\phi')y \\ &+(-w'' + u''w' + u'w'' + v''\phi + v'\phi')z](-v' + u'v' - w'\phi) \\ &+[v' + (-\phi\phi' - v'v'')y + (-\phi' - \frac{v''w' + v'w''}{2})z][-\frac{1}{2}(\phi^2 + v'^2)] \\ &+[w' + (\phi' - \frac{v''w' + v'w''}{2})y + (-\phi\phi' - w'w'')z](\phi - \frac{v'w'}{2}) \end{aligned} \tag{4.82}$$

in which the underlined terms are also due to the use of the second-order elastic rotation matrix.

After terms higher than second order are dropped,

$$\begin{aligned} 2\varepsilon_{xy} &\overset{O(2)}{=} \underline{u'v' - w'\phi} + (\underline{-\phi\phi' - v'v''})y + (-\phi' - \underline{\frac{v''w' + v'w''}{2}})z \\ &\quad +[u' - v''y - w''z](-v') + [w' + \phi' y]\phi \\ &= -(\phi' + \frac{v''w' - v'w''}{2})z \end{aligned} \tag{4.83}$$

Again, let

$$H_{23} = -\frac{1}{2}(\phi' + \frac{v''w' - v'w''}{2}) \tag{4.84}$$

Then

$$\varepsilon_{xy} = zH_{23} \tag{4.85}$$

Observe that the shear strain $\varepsilon_{xy}$ does not vary with $y$ and $\varepsilon_{xy} = 0$ at the centroidal line, as expected of Bernoulli-Euler beams.

Next, one computes $\varepsilon_{xz}$ as

$$\begin{aligned} 2\varepsilon_{xz} &= -w' + \underline{u'w' + v'\phi} + w' + (\phi' - \underline{\frac{v''w' + v'w''}{2}})y + (\underline{-\phi\phi' - w'w''})z \\ &\quad +[u' - \frac{1}{2}(v'^2 + w'^2) + (-v'' + u''v' + u'v'' - w''\phi - w'\phi')y \\ &\quad +(-w'' + u''w' + u'w'' + v''\phi + v'\phi')z](-w' + u'w' + v'\phi) \\ &\quad +[v' + (-\phi\phi' - v'v'')y + (-\phi' - \frac{v''w' + v'w''}{2})z](-\phi - \frac{v'w'}{2}) \\ &\quad +[w' + (\phi' - \frac{v''w' + v'w''}{2})y + (-\phi\phi' - w'w'')z][-\frac{1}{2}(\phi^2 + w'^2)] \end{aligned} \tag{4.86}$$

and

$$\begin{aligned} 2\varepsilon_{xz} &\overset{O(2)}{=} \underline{u'w' + v'\phi} + (\phi' - \underline{\frac{v''w' + v'w''}{2}})y + (\underline{-\phi\phi' - w'w''})z \\ &\quad +[u' - v''y - w''z](-w') + [v' - \phi' z](-\phi) \end{aligned}$$

$$= (\phi' + \frac{v''w' - v'w''}{2})y \tag{4.87}$$

Let

$$H_{32} = \frac{1}{2}(\phi' + \frac{v''w' - v'w''}{2}) \tag{4.88}$$

Then

$$\varepsilon_{xz} = yH_{32} \tag{4.89}$$

Again, observe the shear strain $\varepsilon_{xz}$ does not vary with $z$ and $\varepsilon_{xz} = 0$ at the centroidal line. Also note that $H_{23} = -H_{32}$.

Since warping, a characteristic of non-circular shafts subjected to torsion, and shear effects due to bending are excluded from the deformation field (4.64), it can be shown that the direction and the magnitude of the resultant shear stress predicted from the $\varepsilon_{xy}$ and $\varepsilon_{xz}$ derived here agree with those predicted by the torsion formula for a circular shaft. To do this, let us assume that the material of the beam has $G$ as its shear modulus

$$\sigma_{xy} = 2G\varepsilon_{xy} \qquad\qquad \sigma_{xz} = 2G\varepsilon_{xz} \tag{4.90}$$

Using equations (4.83) and (4.87), the magnitude of the resultant shear stress in the $y_f^i C_i z_f^i$ plane is found (see Figure 4.5):

$$\begin{aligned} \tau &= \sqrt{\sigma_{xy}^2 + \sigma_{xz}^2} \\ &= \sqrt{\frac{1}{4}(2G)^2(\phi' + \frac{v''w' - v'w''}{2})^2y^2 + \frac{1}{4}(2G)^2(\phi' + \frac{v''w' - v'w''}{2})^2z^2} \\ &= G(\phi' + \frac{v''w' - v'w''}{2})\sqrt{y^2 + z^2} \end{aligned} \tag{4.91}$$

which is the same as predicted from the torsion formula (Popov [71]) except for the higher order term $\frac{v''w' - v'w''}{2}$, which is the result of considering the second-order geometry of the

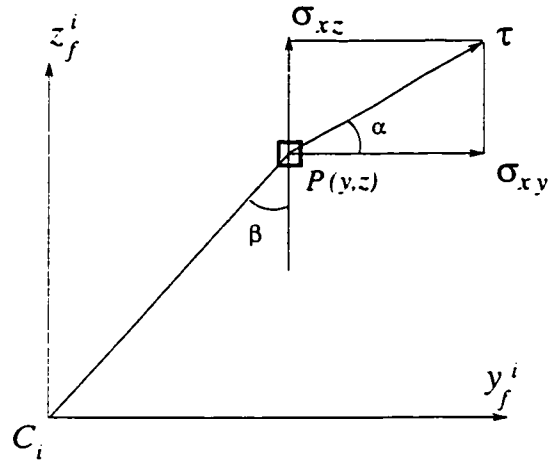



Figure 4.5: The Resultant Shear Stress

shaft combined with the use of its second-order strains. (For a shaft with a circular cross section, $v''w' - v'w'' = 0$, due to its symmetry.) Its direction is obtained by evaluating $\tan(\alpha + \beta)$ as follows

$$\tan(\alpha+\beta) = \frac{\tan\alpha + \tan\beta}{1 - \tan\alpha\tan\beta} \tag{4.92}$$

$$= \frac{-\frac{y}{z} + \frac{y}{z}}{1 + \frac{y}{z}\frac{y}{z}} \tag{4.93}$$

$$= 0 \tag{4.94}$$

Thus, one has

$$\alpha + \beta = 180^\circ \tag{4.95}$$

since $\beta \geq 0$ and $\alpha \geq 0$. Equation (4.95) means that the resultant stress $\tau$ at point $P$ is acting in the same direction as predicted by the torsion formula for a circular shaft.

This brief analysis also serves to show that the assumed deformation field of equation (4.64) predicts the correct deformation for the beam.

Now let us return to equations (4.77), (4.83) and (4.87). It is seen that $\varepsilon_{xx}$, $\varepsilon_{xy}$, and

$\varepsilon_{xz}$ are functions of the elastic deformation variables $u$, $v$, $w$, and $\phi$, and it is possible to vary $\varepsilon_{xx}$, $\varepsilon_{xy}$, $\varepsilon_{xz}$ with respect to them, but first take the variation with respect to the previously defined $H_{ij}$, as follows,

$$\begin{cases} \delta\varepsilon_{xx} &= \delta H_{11} + y\delta H_{12} + z\delta H_{13} + \delta H_{14} \\ \delta\varepsilon_{xy} &= z\delta H_{23} \\ \delta\varepsilon_{xz} &= y\delta H_{32} \end{cases} \tag{4.96}$$

This set of equations, along with others, will be used to find the virtual work of the elastic forces in the next section.

### Virtual Work of Elastic Forces

Now that the strains of the beam have been found, one is ready to compute the virtual work of the elastic forces, using $\int_V \delta\varepsilon^T \sigma dV$ over the volume of the beam, where

$$\delta\varepsilon = \begin{Bmatrix} \delta\varepsilon_{xx} \\ 2\delta\varepsilon_{xy} \\ 2\delta\varepsilon_{xz} \end{Bmatrix}, \quad \sigma = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \end{Bmatrix} = \begin{bmatrix} E & 0 & 0 \\ 0 & 2G & 0 \\ 0 & 0 & 2G \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \end{Bmatrix} \tag{4.97}$$

assuming linear elastic properties of material with $E$ and $G$ as the Young's modulus and shear modulus respectively. Note that $\varepsilon_{yy} = \varepsilon_{zz} = \varepsilon_{yz} = 0$.

To simplify the development for practical purposes, let us first assume that the beam's cross-sectional area is such that

$$\int_A y^n z^m dA = 0 \qquad \text{for } n+m = 1, 3 \text{ and for } n = m = 1 \tag{4.98}$$

where $A$ is the cross sectional area of the beam. This means that the cross section has

two axes of symmetry which coincide with the cross sectional coordinate axes of $y_f^i$ and $z_f^i$, e.g., circular and rectangular cross sections.

To start, one first writes, using equation (4.97)

$$\begin{aligned}\int_V \delta\varepsilon^T \sigma dV &= \int_V (\delta\varepsilon_{xx} E \varepsilon_{xx} + 4\delta\varepsilon_{xy} G \varepsilon_{xy} + 4\delta\varepsilon_{xz} G \varepsilon_{xz}) dV \\ &= E\int_V \delta\varepsilon_{xx}\varepsilon_{xx} dV + 4G\int_V \delta\varepsilon_{xy}\varepsilon_{xy} dV + 4G\int_V \delta\varepsilon_{xz}\varepsilon_{xz} dV \end{aligned} \quad (4.99)$$

This expression is further developed term by term. For the first term, one has

$$\begin{aligned}E\int_V \delta\varepsilon_{xx}\varepsilon_{xx} dV &= E\int_V (\delta H_{11}H_{11} + \underline{\delta H_{11}H_{12}y} + \underline{\delta H_{11}H_{13}z} + \delta H_{11}H_{14} \\ &\quad + \underline{\delta H_{12}H_{11}y} + \delta H_{12}H_{12}y^2 + \underline{\delta H_{12}H_{13}yz} + \underline{\delta H_{12}H_{14}y} \\ &\quad + \underline{\delta H_{13}H_{11}z} + \underline{\delta H_{13}H_{12}yz} + \delta H_{13}H_{13}z^2 + \underline{\delta H_{13}H_{14}z} \\ &\quad + \delta H_{14}H_{11} + \underline{\delta H_{14}H_{12}y} + \underline{\delta H_{14}H_{13}z} + \delta H_{14}H_{14}) dV \end{aligned} \quad (4.100)$$

Due to the assumption (4.98) made on the cross section of the beam and also noting that $H_{14}$ is a homogeneous quadratic function of $y$ and $z$, the underlined terms vanish on carrying out the integration. Thus,

$$\begin{aligned}E\int_V \delta\varepsilon_{xx}\varepsilon_{xx} dV &= E\int_V (\delta H_{11}H_{11} + \delta H_{12}H_{12}y^2 + \delta H_{13}H_{13}z^2 \\ &\quad + \delta H_{11}H_{14} + \delta H_{14}H_{11} + \delta H_{14}H_{14}) dV \\ &= E\int_0^{l_i} (A\delta H_{11}H_{11} + I_{zz}\delta H_{12}H_{12} + I_{yy}\delta H_{13}H_{13}) dx \\ &\quad + E\left(\int_V \delta H_{11}H_{14} dV + \int_V \delta H_{14}H_{11} dV + \int_V \delta H_{14}H_{14} dV\right) \end{aligned} \quad (4.101)$$

in which the definitions

$$J_{yy} \triangleq \int_A z^2 dA, \quad J_{zz} \triangleq \int_A y^2 dA, \quad A \triangleq \int_A dA$$

have been used. They are the second moments of area and the area of the beam's cross-section, respectively.

Now, let us look further in detail at the last three integrals in equation (4.101). Using equations (4.81) and (4.98) the first of them gives

$$\begin{aligned}\int_V \delta H_{11} H_{14} dV &= \frac{1}{2}\int_V \delta H_{11}(a_1 y^2 + a_2 z^2 + a_3 yz) dV \\ &= \frac{1}{2}\int_0^{l_i} \delta H_{11}(a_1 J_{zz} + a_2 J_{yy}) dx \end{aligned} \tag{4.102}$$

Also it can be written from equation (4.81)

$$\delta H_{14} = \frac{1}{2}(\delta a_1 y^2 + \delta a_2 z^2 + \delta a_3 yz)$$

and hence the second term becomes

$$\int_V \delta H_{14} H_{11} dV = \frac{1}{2}\int_0^{l_i} (\delta a_1 J_{zz} + \delta a_2 J_{yy}) H_{11} dx \tag{4.103}$$

Finally, for the last term in equation (4.101) one gets

$$\begin{aligned}\int_V \delta H_{14} H_{14} dV &= \frac{1}{4}\int_V (\delta a_1 y^2 + \delta a_2 z^2 + \delta a_3 yz)(a_1 y^2 + a_2 z^2 + a_3 yz) dV \\ &= \frac{1}{4}\int_V [\delta a_1 a_1 y^4 + (\delta a_1 a_3 + \delta a_3 a_1) y^3 z + (\delta a_1 a_2 + \delta a_2 a_1 + \delta a_3 a_3 y^2 z^2) y^2 z^2 \\ &\quad + (\delta a_2 a_3 + \delta a_3 a_2) y z^3 + \delta a_3 a_3 z^4] dV \\ &= \frac{1}{4}\int_0^{l_i} [\delta(\frac{a_1^2}{2})\int_A y^4 dA + \delta(a_1 a_3)\int_A y^3 z dA + \delta(a_1 a_2 + \frac{a_3^2}{2})\int_A y^2 z^2 dA \\ &\quad + \delta(a_2 a_3)\int_A y z^3 dA + \delta(\frac{a_2^2}{2})\int_A z^4 dA] dx \end{aligned} \tag{4.104}$$

Let

$$JJ_1 \triangleq \int_A y^4 dA \qquad JJ_2 \triangleq \int_A y^3 z dA \tag{4.105}$$

$$JJ_3 \triangleq \int_A y^2 z^2 dA \qquad JJ_4 \triangleq \int_A y z^3 dA \tag{4.106}$$

$$JJ_5 \triangleq \int_A z^4 dA \tag{4.107}$$

which are the fourth moments of the beam's cross-sectional area. For notational brevity, also let

$$b_1 = \frac{a_1^2}{2}, \;\; b_2 = a_1 a_2, \;\; b_3 = a_1 a_2 + \frac{a_3^2}{2}, \;\; b_4 = a_2 a_3, \;\; b_5 = \frac{a_2^2}{2}$$

Thus, equation (4.104) is rewritten as

$$\int_V \delta H_{14} H_{14} dV = \frac{1}{4} \int_0^{l_i} (\delta b_1 JJ_1 + \delta b_2 JJ_2 + \delta b_3 JJ_3 + \delta b_4 JJ_4 + \delta b_5 JJ_5) dx \tag{4.108}$$

Substituting equations (4.102), (4.103) and (4.108) into equation (4.101) yields

$$\begin{aligned} E \int_V \delta \varepsilon_{xx} \varepsilon_{xx} dV &= E \int_0^{l_i} [A \delta H_{11} H_{11} + J_{zz} \delta H_{12} H_{12} + J_{yy} \delta H_{13} H_{13} \\ &\quad + \frac{1}{2} \delta H_{11} (a_1 J_{zz} + a_2 J_{yy}) + \frac{1}{2} (\delta a_1 J_{zz} + \delta a_2 J_{yy}) H_{11} \\ &\quad + \frac{1}{4} (\delta b_1 JJ_1 + \delta b_2 JJ_2 + \delta b_3 JJ_3 + \delta b_4 JJ_4 + \delta b_5 JJ_5)] dx \end{aligned} \tag{4.109}$$

The second integral in equation (4.99) is now developed and gives

$$\begin{aligned} 4G \int_V \delta \varepsilon_{xy} \, \varepsilon_{xy} &= 4G \int_V z \, \delta H_{23} \, z H_{23} \, dV \\ &= 4G \int_0^{l_i} I_{yy} \, \delta H_{23} H_{23} \, dx \end{aligned} \tag{4.110}$$

In much the same way, the last term in equation (4.99) is evaluated and produces

$$\begin{aligned} 4G\int_V \delta\varepsilon_{xz}\varepsilon_{xz}\,dV &= 4G\int_V y\delta H_{32}\,yH_{32}\,dV \\ &= 4G\int_0^{l_i} I_{zz}\delta H_{32}H_{32}\,dx \end{aligned} \tag{4.111}$$

Using equations (4.109), (4.110) and (4.111), equation (4.99) recast as

$$\int_V \delta\varepsilon^T\sigma dV = \delta W_{e1} + \delta W_{e2} + \delta W_{e3} + \delta W_{e4} \tag{4.112}$$

in which

$$\delta W_{e1} = E\int_0^{l_i} (A\,\delta H_{11}H_{11} + J_{zz}\delta H_{12}H_{12} + J_{yy}\delta H_{13}H_{13})\,dx \tag{4.113}$$

$$\delta W_{e2} = \frac{E}{2}\int_0^{l_i} [\delta H_{11}(a_1J_{zz} + a_2J_{yy}) + (\delta a_1J_{zz} + \delta a_2J_{yy})H_{11}]\,dx \tag{4.114}$$

$$\delta W_{e3} = \frac{E}{4}\int_0^{l_i} (\delta b_1JJ_1 + \delta b_2JJ_2 + \delta b_3JJ_3 + \delta b_4JJ_4 + \delta b_5JJ_5)\,dx \tag{4.115}$$

$$\delta W_{e4} = 4G\int_0^{l_i} (J_{yy}\,\delta H_{23}H_{23} + J_{zz}\delta H_{32}H_{32})\,dx \tag{4.116}$$

These four virtual work expressions will produce in the final dynamic equations different terms of various nature regarding their order in and dependency on the generalized coordinates. Before proceeding further, let us have a look at each of them in some detail.

$\delta W_{e1}$ will generate elastic forces associated with the axial deformation and the bendings, and these forces will be of up to third order in the elastic coordinates. In particular, the first term $A\delta H_{11}H_{11}$ produces the elastic force associated solely with axial deformation, including the linear elastic force commonly used in structural dynamics analysis. Considering the equation for $H_{12}$, in addition to producing the linear elastic force in the bending variable $v$ in the dynamic equation associated with the variable, the second term

$I_{zz}\delta H_{12}H_{12}$ will produce the second-order couplings between $u$ and $v$, and among $v$, $w$ and $\phi$ in addition to other second- and third-order terms. Note that while the second-order coupling terms between $u$ and $v$, and $w$ and $\phi$ appear in the dynamic equation corresponding to the bending coordinate $v$, the second-order coupling between $v$ and $\phi$ will appear in the dynamic equation associated with $w$. Observe also that there will be a second-order coupling term between $v$ and $w$ appearing in the dynamic equation of $\phi$. one can examine the third term $I_{yy}\delta H_{13}H_{13}$ in a similar way. There one gets, among other terms, the second-order coupling, between $u$ and $w$, and $v$ and $\phi$ in the dynamic equation corresponding to $w$, between $w$ and $\phi$ in the dynamic equation of $v$, and between $v$ and $w$ in the dynamic equation of $\phi$. Yet, considering the fact that the coupling term resulting from $I_{yy}\delta H_{12}H_{12}$ in the dynamic equation of $\phi$ has an opposite sign to the coupling term from $I_{zz}\delta H_{13}H_{13}$, they will cancel out when $I_{yy}$ equals $I_{zz}$.

The integral $\delta W_{e2}$ exists solely due to the employment of the non-linear strain-displacement relation in evaluating the second-order $\varepsilon_{xx}$, as is clear from the presence of $a_1$ and $a_2$ in it. This integral produces both second- and third-order elastic forces, some in coupled forms. It can be seen from the expressions for $a_1$, $a_2$ and $H_{11}$ that the first term generates second-order forces respectively in $v$, $w$ and $\phi$ in the dynamic equation for axial deformation, along with third-order coupling terms between $v$, $w$, $\phi$ and $u$ scattered among all the dynamic equations. The second term in $\delta W_{e2}$ generates both second- and third-order coupling terms between $v$, $w$, $\phi$ and $u$ respectively. It is observed that the two terms in $\delta W_{e2}$ differ only by the variation. In fact, they can be collected into one single term and this latter form is more amenable to symbolic manipulation due to its compactness.

The third term $\delta W_{e3}$ exists also entirely due to the use of the non-linear strain-displacement relation in the evaluation of $\varepsilon_{xx}$, but unlike the $\delta W_{e2}$ term, it generates mere third-order terms that are multiplied by the fourth moments of area of the beam's

cross section. Since third-order terms are already small compared to lower order ones, their multiplication of the fourth moments of area, which are usually small for slender beams, makes them even smaller. It is therefore conceivable that these terms might be ignored in practice without incurring much error. It is interesting to note that among these negligible terms, there are terms that couple flexural bending deformations $v$ and $w$ with the twist angle $\phi$, and the axial deformation $u$ is not involved in them. Collecting the integrand into one term will also facilitate symbolic manipulation of the integral $\delta W_{e3}$.

The $\delta W_{e4}$ term is easy to interpret. Referring to the definition expressions for $H_{23}$ and $H_{32}$, one sees clearly that it produces the linear elastic forces associated with the torsion of the beam. It also generates second-order couplings amongst $v$, $w$ and $\phi$.

It is interesting and important to point out that in the elastic forces there exist second-order couplings between $u$, $v$, $w$ and $\phi$. The phenomenon of having torsion coupled with stretch and bending has probably never been explicitly identified before, perhaps because they are elusive to physical interpretations.

The complete non-linear formulation of the elastic forces also allows us to gain some useful insight into some of the hotly debated issues in the context of multibody dynamics. The proper form of the approximation for the elastic rotation is the first one. For this purpose, let us look at the fully non-linear expression (4.75) for $\varepsilon_{xx}$, where the second-order approximation for the elastic rotation has been employed, and the second-order $\varepsilon_{xx}$ of equations (4.76) and (4.77) derived from it. Note there are eight terms in equation (4.76) that emanate from the use of the second-order rotation matrix. They are underlined there. The underlined terms in the coefficient of $y$ are examined first. Although two of the four terms are cancelled out there, this does not mask the significance of using the second-order rotation matrix. If the second-order rotation matrix had not been used, the $-u'v''$ and $w'\phi'$ terms would have remained in $H_{12}$ and made their way through the term $\delta H_{12}H_{12}$ in $\delta W_{e1}$ into the final dynamic equations, producing some second-order

coupling terms between $u$ and $v$, and among $v$, $w$ and $\phi$. On the other hand, using the second-order elastic rotation matrix eliminates these terms from the elastic forces. Furthermore, the presence of $u''v' - w''\phi$ in the coefficient of $y$ of equation (4.77), which survived the cancellation in equation (4.76), will produce through $\delta H_{12} H_{12}$ at least some second-order couplings between $u$ and $v$, and among $v$, $w$ and $\phi$, as mentioned earlier in the discussion of $\delta W_{e1}$. Therefore, it follows that the employment of the second-order elastic rotation matrix helps recover second- and higher-order elastic force terms, some of which have counter-balancing effects on the elastic forces otherwise obtained from the commonly used first-order deformation field.

Since the discussion made so far about the coefficient of $y$ translates easily to that of $z$ in the same equations, one now comes to the simple conclusion regarding the proper form of elastic rotation matrix to be used to describe the deformation field of a flexible beam: a second-order elastic rotation matrix is needed, otherwise some second- and higher-order terms in elastic forces that couple $v$, $w$ and $u$, and among $v$, $w$ and $\phi$ would be missing from the system dynamic equations. This conclusion is new, and a similar conclusion can be drawn by examining $\varepsilon_{xy}$ and $\varepsilon_{xz}$ as well.

Another issue that needs re-addressing is the inextensibility assumption of the axial deformation of a beam. This assumption finds its root in the study of dynamics of single flexible beams, like helicopter blades, and has been adopted recently by Piedboeuf [87] in his study of flexible multibody dynamics. As pointed out earlier, there are second-order couplings between $u$ and $\phi$ appearing in the system dynamic equations. Their appearance is the result of not adopting the inextensibility assumption. If the assumption is used, these terms will be eliminated from the dynamic equations. For a system where only one beam is involved, the absence of these eliminated terms might not have a large effect on the system simulation results. For a flexible multibody system, however, one can expect that one body's torsional deformation will influence the position and orientation

of other bodies in the system even when the stretch $u$ is small, and should be preserved in the dynamic equations as much as possible for simulation fidelity. Therefore, it may be important to avoid the inextensibility assumption, in the study of some flexible multibody systems, unless torsion is negligible, if accurate prediction of system behavior is to be obtained. This is also a new observation.

The virtual work done by the elastic forces is automatically calculated symbolically in DynaFlex using equation (4.112) with the deformation variables $u$, $v$, $w$, and $\phi$ discretized according to equation (4.48).

### Discretized Elastic Forces

From equation (4.112), a continuous model of elastic forces could be obtained (for simple problems), but here it is chosen to discretize the deformation variables of the beam using the Rayleigh-Ritz method and to derive the elastic forces corresponding to generalized elastic coordinates of the beam. Although DynaFlex does not make use of the material presented in this subsection, it is discussed for completeness when a finite element method is adopted.

As stated before, the discretization of the deformation variables of a Bernoulli-Euler beam takes the form of equation (4.48). Individually, one has

$$u(x,t) = \mathbf{q}_f^T \begin{Bmatrix} N_u(x) \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{4.117}$$

$$v(x,t) = \mathbf{q}_f^T \begin{Bmatrix} 0 \\ N_v(x) \\ 0 \\ 0 \end{Bmatrix} \tag{4.118}$$

$$w(x,t) = \mathbf{q}_f^T \begin{Bmatrix} 0 \\ 0 \\ N_w(x) \\ 0 \end{Bmatrix} \tag{4.119}$$

$$\phi(x,t) = \mathbf{q}_f^T \begin{Bmatrix} 0 \\ 0 \\ 0 \\ N_\phi(x) \end{Bmatrix} \tag{4.120}$$

Considering these equations and the fact that the $H_{ij}$, $a_i$ and $bi$ defined earlier are functions of $u$, $v$, $w$ and $\phi$, all the variations in equation (4.112) can be expressed as a linear combination of the variations of the generalized elastic coordinates $\mathbf{q}_f$, that is,

$$\begin{cases} \delta H_{rs} = \delta \mathbf{q}_f^T \mathbf{H}_{rs}^* & r = 1, \cdots, 3, \;\; s = 1, \cdots, 4 \\ \delta a_r = \delta \mathbf{q}_f^T \mathbf{a}_r^* & r = 1, \cdots, 3 \\ \delta b_r = \delta \mathbf{q}_f^T \mathbf{b}_r^* & r = 1, \cdots, 5 \end{cases} \tag{4.121}$$

where $\mathbf{H}_{rs}^* = \left[\frac{\partial H_{rs}}{\partial q_{f1}}, \frac{\partial H_{rs}}{\partial q_{f2}}, \cdots, \frac{\partial H_{rs}}{\partial q_{fN_r}}\right]^T$, $\mathbf{a}_r^* = \left[\frac{\partial a_r}{\partial q_{f1}}, \frac{\partial a_r}{\partial q_{f2}}, \cdots, \frac{\partial a_r}{\partial q_{fN_r}}\right]^T$, and so on. Note that the starred quantities can still be functions of $\mathbf{q}_f$.

Using equation (4.121) in equation (4.112) yields

$$\int_V \delta\varepsilon^T \sigma = \delta \mathbf{q}_f^T \int_0^{l_i} \{E[A\mathbf{H}_{11}^* H_{11} + I_{zz}\mathbf{H}_{12}^* H_{12} + I_{yy}\mathbf{H}_{13}^* H_{13}$$

$$
\begin{aligned}
&+\frac{\mathbf{H}_{11}^{\bullet}}{2}(a_1 I_{zz} + a_2 I_{yy}) + \frac{H_{11}}{2}(\mathbf{a}_1^{\bullet} I_{zz} + \mathbf{a}_2^{\bullet} I_{yy}) \\
&+\frac{1}{4}(\mathbf{b}_1^{\bullet} JJ_1 + \mathbf{b}_2^{\bullet} JJ_2 + \mathbf{b}_3^{\bullet} JJ_3 + \mathbf{b}_4^{\bullet} JJ_4 + \mathbf{b}_5^{\bullet} JJ_5)] \\
&+4G(I_{yy}\mathbf{H}_{23}^{\bullet} H_{23} + I_{zz}\mathbf{H}_{32}^{\bullet} H_{32})\} \, dx \\
\triangleq\ & \delta \mathbf{q}_f^T \mathbf{K}^{\bullet} \qquad (4.122)
\end{aligned}
$$

where $\mathbf{K}^{\bullet}$ is a column matrix of the generalized elastic forces corresponding to the generalized elastic coordinates $\mathbf{q}_f$ of the current body. Entries in $\mathbf{K}^{\bullet}$ contain terms up to the third order in $\mathbf{q}_f$, but it is complete only to the second order. As mentioned earlier, there are second-order terms in all possible combinations of the generalized elastic coordinates $\mathbf{q}_f$.

## 4.4 Concluding Remarks

The traditional first-order deformation displacement approximation has been investigated in detail and it has been found that it is not able to provide the correct second-order elastic rotation matrix needed for generating complete first order inertial forces in the system kinetic equations when the Principle of Virtual Work is evoked. In fact, in order to generate inertial forces complete up to the first order in elastic coordinates using any variational dynamics laws, a complete second-order deformation displacement approximation is needed.

To address this problem, a new complete second-order deformation displacement field of the beam has been put forward using a complete second-order elastic rotation matrix resulting from the two-Euler-angle approach. A method to obtain a complete deformation displacement field of any desired order has also been provided.

Graph-theoretic terminal equations based on the proposed $O(2)$ deformation field for a Bernoulli-Euler beam have been discussed. They consist of a set of kinematic equations

and the sum of the virtual work expressions for the inertial forces, the gravitational force, and the internal elastic forces. When other types of body forces are present, one needs to calculate their virtual work for them to be included in the virtual work terminal equation of the beam.

It is interesting to note that although the traditional kinematics of Bernoulli-Euler beams has been around for hundreds of years, mainly related to structural mechanics, this does not mean that it is easy to translate into the realm of multibody dynamics. The new findings that to generate complete first-order inertial forces of a Bernoulli-Euler beam in the context of multibody dynamics requires a complete second-order deformation approximation, and that using the traditional deformation field approximation would have some second-order coupling terms missing in the elastic forces is an example that nothing can be taken for granted when dealing with flexible multibody systems. The importance of using the complete second-order deformation approximation will be substantiated numerically and discussed further in Chapter 7 with a three-dimensional spin-up beam with an off-set mass at its tip.

Another point is in order too. The inclusion of the foreshortening in the deformation displacement field gives rise to the geometric stiffening terms appearing in the inertial forces in the kinetic equations in the form of first-order terms (Vigneron [8], Kaza and Kvaternik [9], Sharf [58]). This effect has to be incorporated through the second-order elastic force terms which couple $u$ with $v$, $w$ if the foreshortening is not included in the deformation displacement field (Kaza and Kvaternik [9]). While both approaches are capable of modelling the stiffening effect correctly, the system equations they lead to have different structures, which may have quite different numerical properties.

Due to the inclusion of the foreshortening in equation (4.64), the recovered second-order terms in the elastic forces, as discussed in Section 4.3.2, do not have as big an impact on the system behavior as they do when the foreshortening is excluded from the

deformation field (Sharf [58]). Thus they can be neglected for some problems without affecting the fidelity of the model appreciably, resulting in linear elastic force terms only. This statement is further substantiated by numerical examples in Chapter 7. Even so, the importance of their presence in system equations may not be neglected for other problems.

# Chapter 5

# Formulation Procedure

It is well known that the main and outstanding advantages of the graph-theoretic approach, when traditional complementary variables are used, are the clear concepts embedded, the versatility promised to deal with complex systems, and the systematic procedure given to formulate system equations. With virtual work used as the through variable, these features are still well preserved.

In this chapter the systematic procedure for generating system equations is discussed. Specifically, it is shown how to construct a graph for a given system, how to select a formulation tree and its implications on the structure of system equations to be obtained, and what steps to follow to get the desired form of the system equations. Three symbolic examples are presented in the end to further demonstrate the systematic nature of the proposed approach.

## 5.1 Construction of a System Graph

To analyze a dynamic system using the graph-theoretic method, the first thing an analyst has to do is construct a linear graph for the system under study. The system

graph is constructed on the basis of the information about what components comprise the system, i.e. individual component properties, and in which order they are connected with each other, i.e. the topology of the system. In other words, we construct the system graph based on information at both the component level and the system level.

For a mechanical system, with components' graph-theoretic models given in Section 3.4, it is a simple matter to construct the system graph by following the two steps:

1. Draw the graph for each component: mentally dissect the system into a number of components, and then replace each component with its terminal graph discussed in Section 3.4.

2. Draw the system graph: unite the component graphs at the appropriate nodes in the same way as the components themselves are connected in the real system. All datum nodes appearing in the component graphs are joined to form a common datum node.

Following is an example of a slider-crank mechanism to illustrate these steps. The mechanism consists of a rigid crank driven by a torque, a mass block and a flexible link, as shown in Figure 5.1. Frame $X^2AY^2$ is the body reference frame for the flexible link. It is located at one end and remains tangent to the beam.

The system is mentally dissolved into individual components as shown in Fig.5.2, and each of these components is replaced with their corresponding graph.

It is seen that there are four joints, three pin joints and one welded joint (employed to model the sliding joint, as mentioned before), in the system. Each of them is replaced by the standard three-edge component graph, as indicated in Fig.5.2. The inertially-fixed arm $e25$ locates pin joint $O$ on the ground. Edge $e26$, defined as a sliding arm on the ground, locates the slider relative to the inertial frame. Together with $e75$, $e76$, and $e65$ (defined as a weld joint), it models the sliding joint at C.

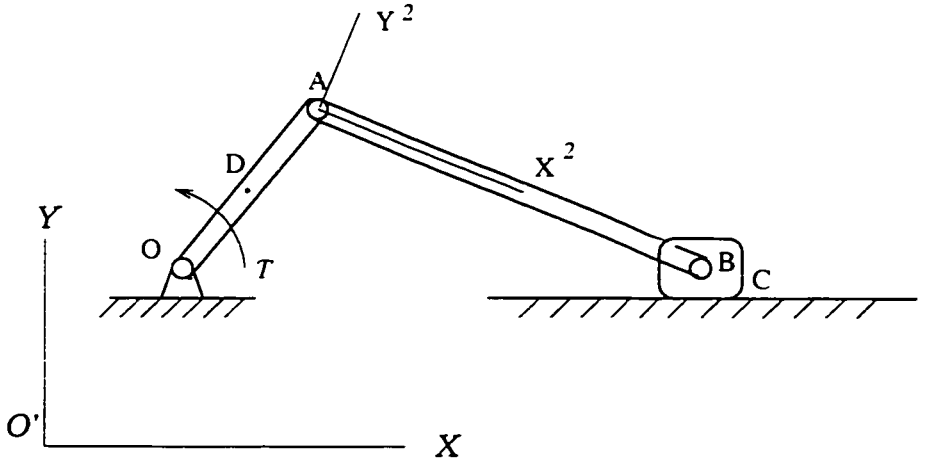



Figure 5.1: The Slider-Crank Mechanism with Flexible Link

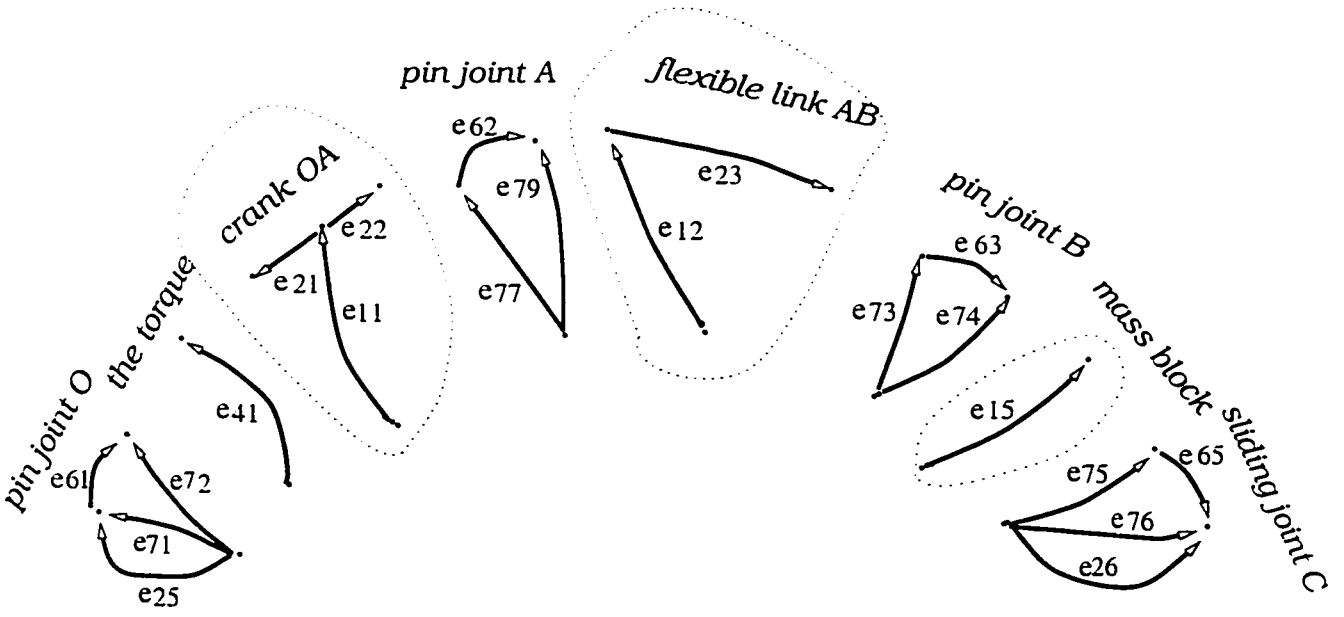



Figure 5.2: The Graph for Components of the Slider-Crank Mechanism

For the rigid crank, because it is acted upon by the two pin joints $O$ and $A$, three edges are needed: edge $e6$, which starts from an inertial reference point and terminates at the mass center of the crank, represents the body itself; edges $e4$ and $e5$ are two rigid arms, used to locate the imparted actions from the pin joints. It is true that there is also an applied torque acting on the crank, and an additional rigid arm could be used to locate its point of application. Yet since the crank is rigid, the point of action of the torque could be specified anywhere on the crank without affecting the results. Therefore the already specified three points on the crank associated with edges e4, e5 and e6 can be used for this purpose.

The externally-applied torque driver itself is represented by a single edge that starts from an inertial reference point and ends at its point of application on the body it acts.

Two edges are used for the flexible link under action. Edge $e10$, which is a flexible body edge, starts from an inertial reference point and terminates at the origin of the body reference for the link. Edge $e11$, which is a flexible arm and starts from point $A$ and ends at point $B$, locates the action transmitted through the joint $B$ when the body reference originates at point $A$. If the body reference originates at point $B$, the flexible arm will start at $B$ and end at $A$. (A body reference originating at an intermediate point is not recommended. It unnecessarily complicates the modelling of a beam.)

It is easier to understand the graph for the mass block if the block is regarded as a point mass with rigid arms of zero length to locate the actions applied on it.

Having obtained the graphs for each individual component, one goes to the second step, i.e., uniting them at the appropriate nodes in the same order as the components are connected in the real system and joining the datum nodes in the component graphs to form a common inertial reference point $O'$. The result is shown in Figure 5.3.

Noteworthy is the torque driver. Its graph, a single edge, starts from the system inertial reference point and terminates at the mass center of the crank. It could also have

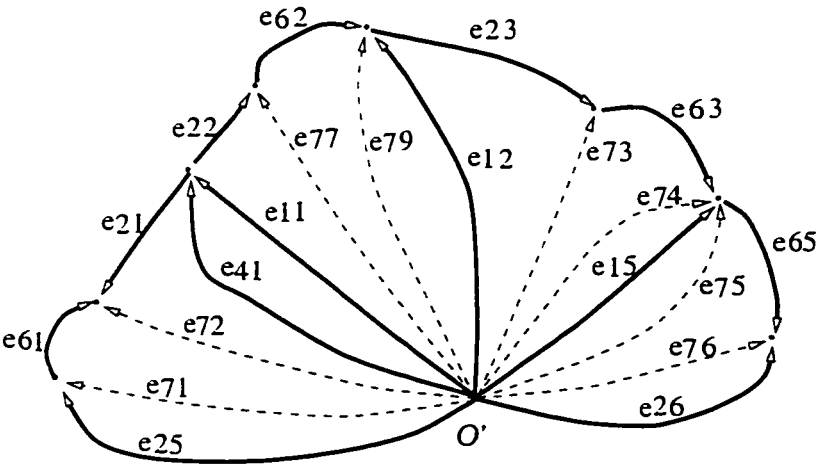



Figure 5.3: The System Graph for the Slider-Crank Mechanism

ended at the head of edge e4 or e5 because of the aforementioned reason. The dependent virtual work elements are drawn in dashed lines here just to make the graph look a little clearer.

Note that in the current VW graph-theoretic approach, only one graph is needed for a system, as in the traditional vector-network approach by Andrews [63].

## 5.2 Selection of the Formulation Tree

The linear graph of a system, created as described in the previous section, contains sufficient information about the mechanical system to permit the generation of the equations of motion by a simple, methodical substitution procedure. The procedure is the same regardless of the complexity of the system, and is basically a branch-chord formulation from the graph-theoretic point of view. This nature of the generation process renders it most suitable for the basis of a computer algorithm that generates a set of system equations as output, given only the system description as input.

Although it is true that from the system graph, one is able to write cutset and circuit

equations to generate correct system equations for any given tree, the proper selection of a formulation tree is very important in that the chosen tree combined with the formulation procedure has a crucial bearing on the structure and complexity of the final system equations generated, as has been demonstrated in [64].

In graph-theoretic terminology, the tree across and cotree through variables are known as "primary variables", while the tree through and cotree across variables as "secondary variables". This is because the secondary variables can be expressed as functions of the primary variables using the chord and branch transformation equations [74]. By using these linear equations, the secondary variables can be eliminated from the set of governing equations. The result is a smaller number of equations solely expressed in terms of across variables for tree elements and through variables for cotree elements. Once these primary variables have been determined, the secondary variables can be obtained through back substitution into the branch and chord transformation equations. This general statement about the graph-theoretic formulation clearly tells something one should heed about the tree selection.

Therefore, in selecting a formulation tree, we first stipulate that body arms associated with all body elements be in the tree. Body arms associated with the inertial base may be combined with other elements. Beside body arms, either body elements or joint edges can be allocated to complete the tree selection.

Three scenarios develop regarding the form and nature of the final set of system equations. By selecting all body elements into the tree, termed "body tree", the final system equations of motion will contain absolute coordinates of the bodies as unknowns, but joint reactions may or may not appear in the final system kinetic equations, depending on the equation generation procedure adopted (see Section 5.3). On the other hand, selecting joint edges alone, termed "joint tree", results in system equations that have joint coordinates as unknown variables. For open-loop systems, no joint reactions will be

present in the final equations with a joint tree. For closed loop systems, joint reactions may or may not appear, depending again on the equation generation procedure used. The third possibility is to select both body and joint elements to complete the tree, which is called a "hybrid tree". Despite the different selection of trees, the mathematical model of the system remains the same, and the generation process of system equations to be introduced next remains the same too when the tree is selected according to the above rules. Only the forms of final system equations are different.

## 5.3 Generation of System Equations

Once the formulation tree has been picked based upon the basic rules outlined in the preceding subsection, one is ready to the generate system equations by following the steps discussed below.

**Step 1: Eliminate Secondary Across Variables**

This step uses branch transformation equations to express cotree across variables (except those already known for motion drivers and joints), which are excess variables, as functions of branch coordinates $\mathbf{q}$, for later use with VW terminal equations of various elements, among others.

It is in this step that the kinematic quantities of a dependent VW element and a cotree body element are calculated in terms of branch coordinates. These kinematic quantities may include displacements (rotations), velocities, virtual displacements (rotations) and accelerations, depending on the imparted action the dependent VW element is modelling.

**Step 2: Generate Kinematic Constraint Equations**

Since the number of branch coordinates $\mathbf{q}$ may not be equal to the degrees of freedom of the system under study, this step is required to produce the equation(s) constraining them. These equations, called circuit equations in graph theory, are generated from the fundamental circuits for any cotree joints and evaluated by substituting kinematic termi-

nal equations of relevant elements involved in the circuits to yield constraint equations that can be at the displacement, velocity and acceleration level. Circuit equations defined by cotree motion drivers are also evaluated in a similar fashion. Whenever a body tree is employed, equations will result from this step even if an open loop system is being considered. One can avoid generating these equations for such a system by employing a joint tree; in this case, the branch coordinates are independent variables.

It is seen that in fact the first and second steps are all branch transformations from a graph-theoretic point of view. Only the equations from the first step will be substituted back into VW terminal equations of elements which are needed to generate system VW equations in the next step. The traditional chord transformation is not needed here in the current VW graph-theoretic approach.

**Step 3: Generate System Virtual Work Equation**

Virtual work equations for the system are obtained by writing a cutset equation for each tree border joint (joint that is both in the tree and connects the system to the inertial frame) and tree body element. These equations may either be summed to zero or remain as is. Final system equations of different nature will result, depending on whether kinematic constraint equations are considered. Both of these issues are further explained in the next step.

**Step 4: Extract Kinetic Equations**

In this step, the system VW equations generated in the last step are manipulated, with or without using the constraint equations, to produce the system equations. Basically there are two approaches.

**Approach 4.1**

In this first approach, the virtual work equations from each cutset are not summed to zero. Kinetic equations are extracted from each of them by setting to zero the coefficients of the virtual branch coordinates involved. There will be as many kinetic equations as

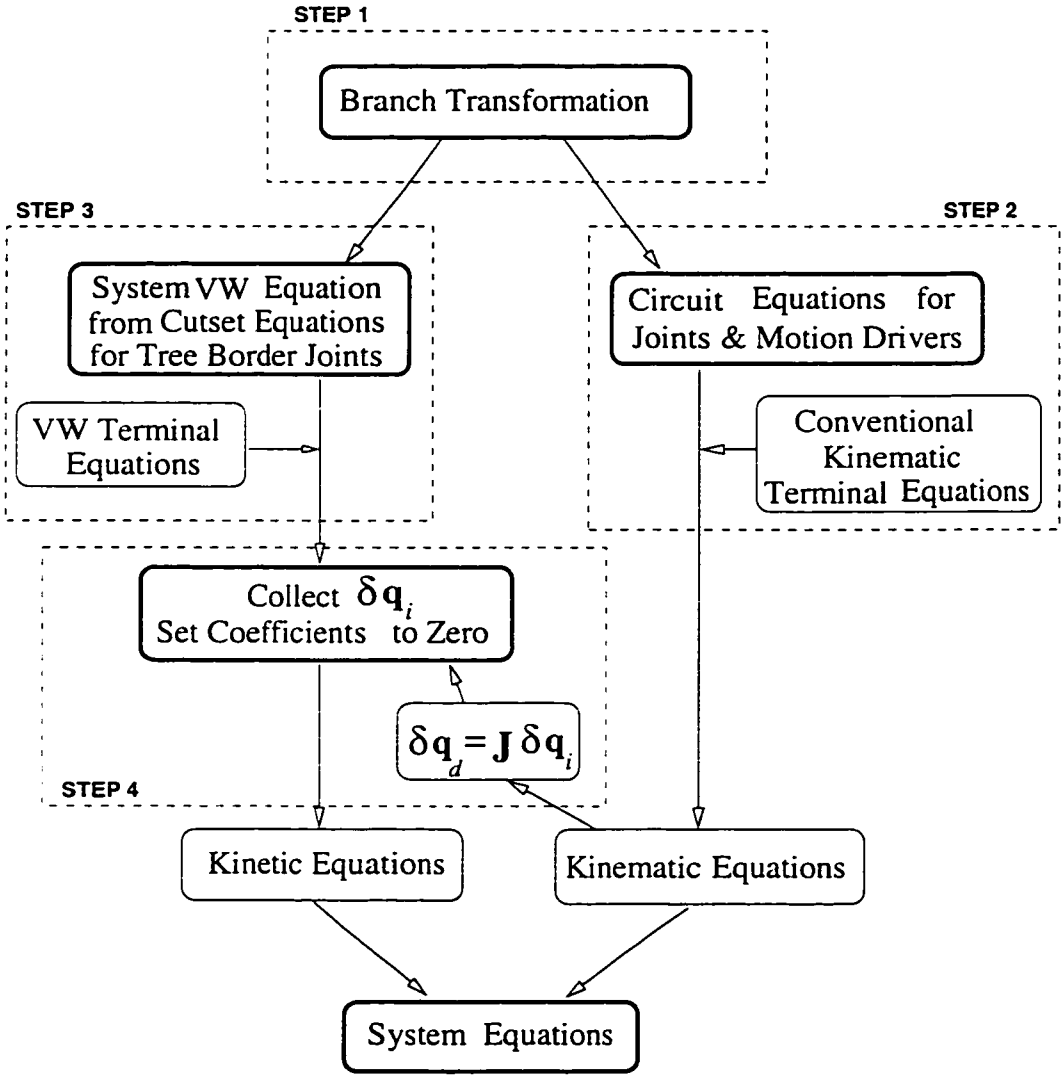



Figure 5.4: The Virtual Work GT Approach Flowchart

tree degrees of freedom, i.e., the number of branch coordinates, and cotree joint forces will appear in them. The final system equations of motion are the union of these kinetic equations with the kinematic constraint equations obtained in Step 1. The general form of the system equations thus generated is

$$\begin{cases} \mathbf{M}^{*}\ddot{\mathbf{q}} + \mathbf{C}_{\mathbf{q}}^{T}\lambda = \mathbf{F}^{*} \\ \mathbf{C}(\mathbf{q}, t) = 0 \end{cases} \tag{5.1}$$

where $\mathbf{M}^{*}$ is the $T_d \times T_d$ mass matrix, with $T_d$ being the tree degrees of freedom; $\mathbf{C}$ is the $(T_d - S_d)\times 1$ constraint equations, with $S_d$ being the system degrees of freedom; $\mathbf{C_q}$ is the Jacobian matrix of $\mathbf{C}$; $\lambda$ is a column matrix of the $T_d - S_d$ Lagrange multipliers associated with the constraint equations; column matrix $\mathbf{F}^{*}$ of size $T_d \times 1$ collects the quadratic velocity terms, the elastic forces and other forces. Equation (5.1) constitutes a set of $2T_d - S_d$ DAEs in terms of $\mathbf{q}$ and $\lambda$.

When a body tree is selected, this approach generates the same system equations as the absolute coordinate approach [111] does.

**Approach 4.2**

In this second approach, the virtual work equations from each cutset are summed to zero to give only one single virtual work expression for the system, as would normally be done in a hand derivation. The virtual work of dependent elements associated with a joint is excluded from this expression since they will eventually cancel out or evaluate to zero. The varied branch coordinates are no longer independent of one another, and one can not set their coefficients to zero to get the kinetic equations. This is because the exclusion of joint dependent VW elements is equivalent to setting the sum of virtual work done by joint forces to zero, which in turn implies the imposition of kinematic conditions for the two bodies to join at the joint.

In this case, the kinetic equations are extracted in this way: first obtain the interdependence of the varied tree across variables $\delta\mathbf{q}$ from the kinematic constraint equations generated in Step 2. These equations, which are linear in $\delta\mathbf{q}$, are then solved for those $\delta\mathbf{q}$ which are selected as dependent functions, denoted $\delta\mathbf{q}_d$, of the remaining varied (independent ones), denoted $\delta\mathbf{q}_i$, which are equal in number to the system degrees of freedom, i.e.,

$$\delta\mathbf{q}_d = \mathbf{J}\delta\mathbf{q}_i \tag{5.2}$$

The matrix $\mathbf{J}$ is derived from $\mathbf{C}(\mathbf{q}, t)$ as shown later in Chapter 6. Afterwards, this set of variational equations is substituted into the system virtual work equation to eliminate $\delta\mathbf{q}_d$. The result is a linear equation in the independent varied branch coordinates $\delta\mathbf{q}_i$. Finally, setting the coefficients of $\delta\mathbf{q}_i$ to zero yields the kinetic equations of the system.

It is helpful to emphasize that with Approach 4.1, there are only as many kinetic equations (ordinary differential equations) as there are system degrees of freedom, and no joint forces will appear in them. The system equations of motion are the union of the kinetic equations and the kinematic constraint equations generated in Step 2. Thus, the general form of the system equation is now

$$\begin{cases} \mathbf{M}\ddot{\mathbf{q}} = \mathbf{F} \\ \mathbf{C}(\mathbf{q}, t) = 0 \end{cases} \tag{5.3}$$

where $\mathbf{M}$ is now a $S_d \times T_d$ mass matrix; $\mathbf{C}$ is the same $(T_d - S_d)\times 1$ constraint equations; column matrix $\mathbf{F}$ still collects the quadratic velocity terms, the elastic forces and other forces, but it now has a reduced size of $S_d \times 1$ accordingly. Equation (5.3) constitute a set of only $T_d$ DAEs, in terms of only $\mathbf{q}$.

When a joint tree is selected, this second formulation generates system equations that have neither joint reactions appearing nor joint constraint equations attached for open

loop systems. For systems with closed loops, it generates kinetic equations that do not contain joint reactions no matter what type of tree is selected.

Other combinations of manipulating the virtual work and kinematic constraint equations are possible: for example, summing some VW cutset equations to zero while leaving others as they are. They will, however, in the end produce equations in either of the two basic forms discussed here.

As is clear from equations (5.1) and (5.3), the number of equations involved in each of these two sets is the same as that of the unknown variables in the same set, i.e., the branch coordinates $\mathbf{q}$ and the Lagrange multipliers $\lambda$. Therefore, they are a necessary and sufficient set for solution.

A flowchart that illustrates the four steps explained above is shown in Figure 6.3.

To adequately demonstrate these steps of generating system equations, a few simple examples are presented next that make use of different variations of the procedure outlined just above. The first example is a rigid double pendulum, where the body tree is selected and system equations are extracted in Step 4 with the first approach, resulting in kinetic equations containing constraint reaction forces. The second example is a rigid slider crank. For this problem, a joint tree is chosen. System equations are extracted in Step 4 with the second approach, thus resulting in kinetic equations that are free from any joint reaction forces (or Lagrange multiplier). The third example, which is a planar spin-up beam, serves to show that the proposed VW approach works equally well for systems involving flexible bodies, by following the same steps.

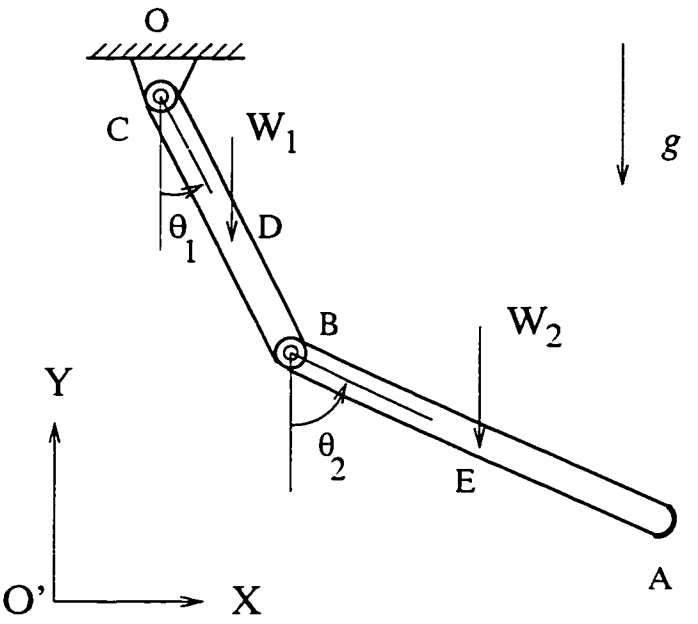



Figure 5.5: The Double Pendulum

## 5.4 Illustrative Examples

### 5.4.1 The Double Pendulum

In this example, equations of motion are derived for the rigid double pendulum shown in Figure 5.5. This system, subjected to a uniform gravitational field $g$, consists of two rigid bodies connected in series to the ground by revolute joints at $B$ and $C$.

The system graph is depicted in Figure 5.6. The bodies are represented by edges $e11$ and $e12$. The two force drivers $\underset{\sim}{W}_1$ and $\underset{\sim}{W}_2$ are modelled with edges $e51$ and $e52$ respectively. Edge $e61$ together with the edges $e71$ and $e74$ represents the revolute joint at $C$, whose two points of connection are located by the rigid body arm $e21$ and inertially fixed body arm $e24$. Edges $e62$, $e72$ and $e73$ together represent the revolute joint at point $B$, whose two points of connection are specified by the two rigid body arms $e22$ and $e23$.

A body tree is selected with the branches shown in bold. Thus, one expects that the

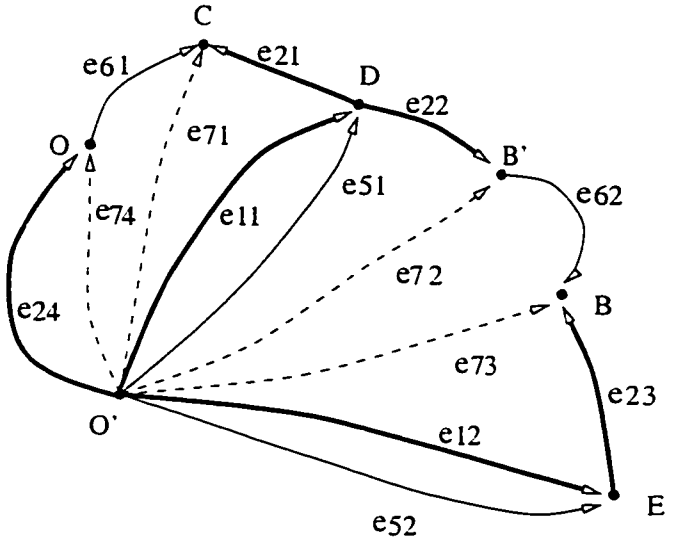



Figure 5.6: The Linear Graph for the Double Pendulum

system equations will be expressed in terms of the branch coordinates $\mathbf{q} = [\underline{r}_1,\ \theta_1,\ \underline{r}_2,\ \theta_2]$, where $\underline{r}_1$ and $\underline{r}_2$ are the position vectors for the mass center of the two links.

**Step 1: Eliminate Secondary Across Variables**

This step uses the branch transformation equations to express the cotree across variables in terms of branch coordinates, for use in later steps. Here for the pendulum, one has

$$\delta\underline{r}_{71} = \delta\underline{r}_{11} + \delta\underline{r}_{21} = \delta\underline{r}_1 + \delta\theta_1\hat{k} \times \underline{r}_{21} \tag{5.4}$$

$$\delta\theta_{71} = \delta\theta_{11} + \delta\theta_{21} = \delta\theta_{11} \triangleq \delta\theta_1 \tag{5.5}$$

$$\delta\underline{r}_{51} = \delta\underline{r}_{11} \triangleq \delta\underline{r}_1 \tag{5.6}$$

$$\delta\underline{r}_{72} = \delta\underline{r}_{11} + \delta\underline{r}_{22} = \delta\underline{r}_1 + \delta\theta_1\hat{k} \times \underline{r}_{22} \tag{5.7}$$

$$\delta\theta_{72} = \delta\theta_{11} + \delta\theta_{22} = \delta\theta_{11} \triangleq \delta\theta_1 \tag{5.8}$$

$$\delta\underline{r}_{73} = \delta\underline{r}_{12} + \delta\underline{r}_{23} = \delta\underline{r}_2 + \delta\theta_2\hat{k} \times \underline{r}_{23} \tag{5.9}$$

$$\delta\theta_{73} = \delta\theta_{12} + \delta\theta_{23} = \delta\theta_{12} \triangleq \delta\theta_2 \tag{5.10}$$

$$\delta \underline{r}_{52} \;=\; \delta \underline{r}_{12} \triangleq \delta \underline{r}_2 \tag{5.11}$$

**Step 2: Generate Kinematic Constraint Equations**

As explained earlier, kinematic constraint equations are derived from the circuit equations for cotree joint edges $e61$ and $e62$, giving at the displacement level:

$$\begin{cases} -\underline{r}_{11} - \underline{r}_{21} + \underline{r}_{24} + \underline{r}_{61} = 0 \\ \underline{r}_{11} - \underline{r}_{12} + \underline{r}_{22} - \underline{r}_{23} + \underline{r}_{62} = 0 \end{cases} \tag{5.12}$$

When the traditional kinematic terminal equations are substituted, they give the constraint equations $\mathbf{C}(\mathbf{q}, t) = 0$, as explained earlier. These equations can be differentiated twice with respect to time to give

$$\begin{cases} -\ddot{\underline{r}}_{11} - \ddot{\underline{r}}_{21} + \ddot{\underline{r}}_{24} + \ddot{\underline{r}}_{61} = 0 \\ \ddot{\underline{r}}_{11} - \ddot{\underline{r}}_{12} + \ddot{\underline{r}}_{22} - \ddot{\underline{r}}_{23} + \ddot{\underline{r}}_{62} = 0 \end{cases} \tag{5.13}$$

which are the constraint equations at the acceleration level. On substitution of corresponding terminal equations for the elements, equations (5.13) become

$$\begin{cases} -\underline{a}_1 - (\ddot{\theta}_1 \hat{k} \times \underline{r}_{21} - \dot{\theta}_1^2 \underline{r}_{21}) = 0 \\ \underline{a}_1 - \underline{a}_2 + (\ddot{\theta}_1 \hat{k} \times \underline{r}_{22} - \dot{\theta}_1^2 \underline{r}_{22}) - (\ddot{\theta}_2 \hat{k} \times \underline{r}_{23} - \dot{\theta}_2^2 \underline{r}_{23}) = 0 \end{cases} \tag{5.14}$$

where $\underline{a}_1 = \ddot{\underline{r}}_1$ and $\underline{a}_2 = \ddot{\underline{r}}_2$ are the center-of-mass accelerations of the upper and lower link, respectively. This set of equations is more expressive of the fact that they constrain the branch coordinates $\mathbf{q} = [\underline{r}_1,\ \theta_1,\ \underline{r}_2,\ \theta_2]$.

**Step 3: Generate System Virtual Work Equation**

System VW equations are obtained by writing VW cutset equations for tree border

joints and tree body edges, as stated above. Since there are only body edges in the tree (a body tree) and no tree border joints in the present case, one needs only to write the VW cutset equations for these edges and have for the system virtual work equations

$$\delta W_{11} + \delta W_{51} + \delta W_{72} - \delta W_{62} + \delta W_{61} + \delta W_{72} = 0 \tag{5.15}$$

for the body edge $e11$ and

$$\delta W_{12} + \delta W_{52} + \delta W_{62} + \delta W_{73} = 0 \tag{5.16}$$

for the body edge $e12$.

On the other hand, from the VW terminal equations defined for the edges, one has

$$\delta W_{11} = -m_1 \underline{a}_{11} \cdot \delta \underline{r}_{11} - I_1 \ddot{\theta}_{11} \delta\theta_{11} \triangleq -m_1 \underline{a}_1 \cdot \delta \underline{r}_1 - I_1 \ddot{\theta}_1 \delta\theta_1 \tag{5.17}$$

$$\delta W_{51} = \underline{W}_1 \cdot \delta \underline{r}_{51} \tag{5.18}$$

$$\delta W_{61} = 0 \tag{5.19}$$

$$\delta W_{62} = 0 \tag{5.20}$$

$$\delta W_{71} = \underline{F}_{61} \cdot \delta \underline{r}_{71} \tag{5.21}$$

$$\delta W_{72} = -\underline{F}_{62} \cdot \delta \underline{r}_{72} \tag{5.22}$$

$$\delta W_{12} = -m_2 \underline{a}_{12} \cdot \delta \underline{r}_{12} - I_2 \ddot{\theta}_{12} \delta\theta_{12} \triangleq -m_2 \underline{a}_2 \cdot \delta \underline{r}_2 - I_2 \ddot{\theta}_2 \delta\theta_2 \tag{5.23}$$

$$\delta W_{52} = \underline{W}_2 \cdot \delta \underline{r}_{52} \tag{5.24}$$

$$\delta W_{73} = \underline{F}_{62} \cdot \delta \underline{r}_{73} \tag{5.25}$$

where $\underline{a}_1$ and $\underline{a}_2$ again stand for the accelerations of the mass centers of the two rigid bodies, and they are substituted into the above two equations, followed by the further

substitution of equations (5.4)-(5.11) to yield

$$\delta \underline{r}_1 \cdot (-m_1 \underline{a}_1 + \underline{W}_1 - \underline{F}_{61} - \underline{F}_{62}) + \delta\theta_1 \left[ -I_1 \ddot{\theta}_1 + \hat{k} \cdot (\underline{r}_{21} \times \underline{F}_{61} - \underline{r}_{22} \times \underline{F}_{62}) \right] = 0 \quad (5.26)$$

$$\delta \underline{r}_2 \cdot (-m_2 \underline{a}_2 + \underline{W}_2 + \underline{F}_{62}) + \delta\theta_2 \left[ -I_2 \ddot{\theta}_2 - \hat{k} \cdot (\underline{r}_{23} \times \underline{F}_{62}) \right] = 0 \quad (5.27)$$

Note, these are system VW equations expressed in the branch coordinates.

**Step 4: Extract Kinetic Equations**

To get the system kinetic equations, one simply sets the coefficients of the varied branch coordinates in equations (5.26) and (5.27) to zero and obtain

$$\begin{cases} m_1 \underline{a}_1 &= \underline{W}_1 + \underline{F}_{61} - \underline{F}_{62} \\ I_1 \ddot{\theta}_1 &= \hat{k} \cdot (\underline{r}_{21} \times \underline{F}_{61} - \underline{r}_{22} \times \underline{F}_{62}) \\ m_2 \underline{a}_2 &= \underline{W}_2 + \underline{F}_{62} \\ I_2 \ddot{\theta}_2 &= \hat{k} \cdot (\underline{r}_{23} \times \underline{F}_{62}) \end{cases} \quad (5.28)$$

This is the set of kinetic equations for the double pendulum, and together with equations (5.12) or (5.14) they form the system equations. It is seen that there are ten system equations which contain ten scalar unknowns. Thus, these equations constitute a necessary and sufficient set for the double pendulum problem.

Note, one could have combined equations (5.15) and (5.16) into one single system VW equation, and carried out the remaining steps. Even so, one still obtains the same system equations as (5.12) (or (5.14)) and (5.28), as pointed out earlier.

As expected, the final equations are expressed in terms of the across variables associated with the bodies, i.e., $\underline{r}_1$, $\theta_1$, $\underline{r}_2$ and $\theta_2$. Besides, the joint reaction forces appear in the system kinetic equations (5.28) due to the extraction of kinetic equations with all branch coordinates held as independent of one another. The kinetic equations obtained

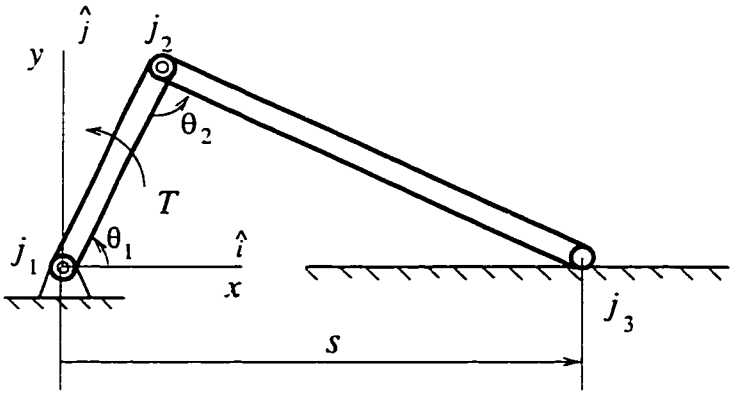



Figure 5.7: The Slider-Crank

above are exactly the same as those obtained by applying the Newton-Euler equations to individual bodies.

### 5.4.2 Planar Slider-Crank Mechanism

In this example, system equations of motion are derived for the rigid slider-crank mechanism shown in Figure 5.7. It is subjected to a torque driver at pin joint $j1$.

The system graph is depicted in Figure 5.8. Edges $e11$ and $e12$ represent respectively bodies $j1j2$ and $j2j3$. Joint $j1$ is represented by edges $e61$ and $e71$, joint $j2$ by $e62$, $e72$ and $e73$, and joint $j3$ by $e63$, $e74$ and $e75$. The joint locations on the bodies are specified by $e21$, $e22$, $e23$ and $e24$. The locations of the two border joints are designated by the datum node and edge $e25$, a sliding arm. The driving torque is modelled by the force driver edge $T$. From this graph, with the help of terminal equations, the system equations of motion are systematically generated as described earlier.

A joint tree is selected this time with the branches shown in bold. This means that the system equations will be expressed in terms of $\mathbf{q} = [\theta_{61},\ \theta_{62},\ s_{25}] \triangleq [\theta_1,\ \theta_2,\ s]$.

**Step 1: Eliminate Secondary Across Variables**

Using circuit equations for chords, cotree across variables (except those already known

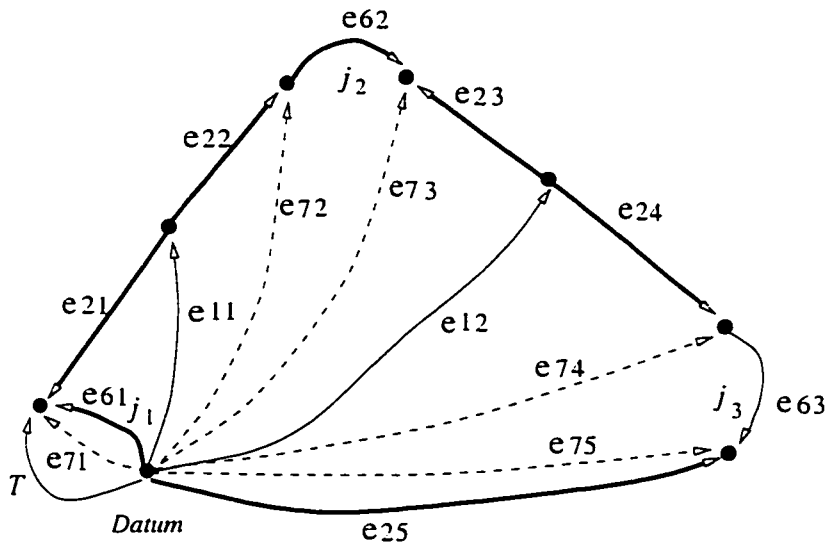



Figure 5.8: System Graph of The Slider-Crank

for motion drivers and joints) are expressed now as functions of tree across variables for later use.

For edge $T$,

$$\dot{\theta}_T = \dot{\theta}_{61} = \dot{\theta}_1 \tag{5.29}$$

For edge $e11$,

$$\begin{cases} \theta_{11} &= \theta_{61} \triangleq \theta_1 \\ \dot{\theta}_{11} &= \dot{\theta}_{61} - \dot{\theta}_{21} = \dot{\theta}_{61} = \dot{\theta}_1 \\ \ddot{\theta}_{11} &= \ddot{\theta}_{61} - \ddot{\theta}_{21} = \ddot{\theta}_{61} = \ddot{\theta}_1 \\ \underset{\sim}{\dot{r}}_{11} &= \underset{\sim}{\dot{r}}_{61} - \underset{\sim}{\dot{r}}_{21} = -\dot{\theta}_1 \hat{k} \times \underset{\sim}{r}_{21} \\ \underset{\sim}{\ddot{r}}_{11} &= \underset{\sim}{\ddot{r}}_{61} - \underset{\sim}{\ddot{r}}_{21} = \dot{\theta}_1^2 \underset{\sim}{r}_{21} - \ddot{\theta}_1 \hat{k} \times \underset{\sim}{r}_{21} \end{cases} \tag{5.30}$$

For edge $e12$,

$$\begin{cases} \theta_{12} &= \theta_{61} + \theta_{62} \triangleq \theta_1 + \theta_2 \\ \dot{\theta}_{12} &= \dot{\theta}_1 + \dot{\theta}_2 \\ \ddot{\theta}_{12} &= \ddot{\theta}_1 + \ddot{\theta}_2 \\ \underline{\dot{r}}_{12} &= \dot{\theta}_1 \hat{k} \times (\underline{r}_{22} - \underline{r}_{21}) - \dot{\theta}_2 \hat{k} \times \underline{r}_{23} \\ \underline{\ddot{r}}_{12} &= \dot{\theta}_1^2 (\underline{r}_{21} - \underline{r}_{22}) - \ddot{\theta}_1 \hat{k} \times (\underline{r}_{21} - \underline{r}_{22}) \\ &+ (\dot{\theta}_1 + \dot{\theta}_2)^2 \underline{r}_{23} - (\ddot{\theta}_1 + \ddot{\theta}_2) \hat{k} \times \underline{r}_{23} \end{cases} \tag{5.31}$$

**Step 2: Generate Kinematic Constraint Equations**

Constraint equations for branch coordinates are derived from the circuit equations corresponding to cotree joints and cotree motion drivers. For edge $e63$, one has from its circuit equation, after relevant terminal equations at the displacement level have been substituted

$$-\underline{r}_{21} + \underline{r}_{22} - \underline{r}_{23} + \underline{r}_{24} - \underline{r}_{25} = 0 \tag{5.32}$$

which, after the relevant terminal equations and branch transformations are substituted, gives the two scalar constraint equations by joint $j3$ in terms of $\theta_1$, $\theta_2$, and $s$, i.e., $C(q,t) = 0$. From these two equations, corresponding scalar velocity level equations are then obtained. They are solved for the dependent velocities of $\dot{s}$ associated with edge $e25$ and $\dot{\theta}_2$ associated with $e62$ in terms of the chosen independent across variable $\dot{\theta}_1$ associated with $e61$. Since the relation between $\dot{s}$ and $\dot{\theta}_1$ will not be needed here, only the other equation is listed, after relevant terminal equations at the velocity level are substituted, as

$$\dot{\theta}_2 = \frac{(\underline{r}_{22} - \underline{r}_{21} - \underline{r}_{23} + \underline{r}_{24}) \cdot \hat{i}}{(\underline{r}_{23} - \underline{r}_{24}) \cdot \hat{i}} \dot{\theta}_1 = J_1 \dot{\theta}_1 \tag{5.33}$$

where $J_1$ is the first row of the matrix **J** in equation (5.2). Thus,

$$\delta\theta_2 = \frac{(\underset{\sim}{r}_{22} - \underset{\sim}{r}_{21} - \underset{\sim}{r}_{23} + \underset{\sim}{r}_{24}) \cdot \hat{\imath}}{(\underset{\sim}{r}_{23} - \underset{\sim}{r}_{24}) \cdot \hat{\imath}} \delta\theta_1 \tag{5.34}$$

**Step 3: Generate System Virtual Work Equations**

Using virtual work as a through variable and considering the fact that the virtual work of the pair of dependent elements corresponding to the same internal joint sums to zero, one has from the cutset equation of the tree border joint edge $e62$

$$\delta W_T + \delta W_{71} + \delta W_{61} + \delta W_{11} + \delta W_{12} + \delta W_{74} - \delta W_{63} = 0 \tag{5.35}$$

in which it is defined that

$$\begin{cases} \delta W_T &= T\delta\theta_T \\ \delta W_{11} &= -m_1\ddot{\underset{\sim}{r}}_{11}\cdot\delta\underset{\sim}{r}_{11} - I_1\ddot{\theta}_{11}\delta\theta_{11} \\ \delta W_{12} &= -m_2\ddot{\underset{\sim}{r}}_{12}\cdot\delta\underset{\sim}{r}_{12} - I_2\ddot{\theta}_{12}\delta\theta_{12} \\ \delta W_{61} &= 0 \\ \delta W_{63} &= 0 \end{cases} \tag{5.36}$$

in addition to

$$\begin{cases} \delta W_{71} &= 0 \\ \delta W_{74} &= 0 \end{cases} \tag{5.37}$$

since they are associated with the two border joints where the virtual work of the joint forces is constantly zero.

Substitution of equations obtained in Step 1 into equations (5.36) through (5.37), and

those in turn into the system virtual work equation (5.35) yields

$$\begin{aligned}\{\, T \;&+\; m_1\ddot{\underset{\sim}{r}}_1 \cdot (\dot{k} \times \underset{\sim}{r}_{21}) - I_1\ddot{\theta}_1 \\ &-\; m_2\ddot{\underset{\sim}{r}}_2 \cdot \left[\dot{k} \times (\underset{\sim}{r}_{22} - \underset{\sim}{r}_{21} - \underset{\sim}{r}_{23})\right] - I_2\ddot{\theta}_{12} \,\}\,\delta\theta_1 \\ &+\; (m_2\ddot{\underset{\sim}{r}}_2 \cdot \dot{k} \times \underset{\sim}{r}_{23} - I_2\ddot{\theta}_{12})\delta\theta_2 = 0 \end{aligned} \tag{5.38}$$

where the symbols $\ddot{\underset{\sim}{r}}_1$, $\ddot{\underset{\sim}{r}}_2$, $\ddot{\theta}_{12} = \ddot{\theta}_1 + \ddot{\theta}_2$ are retained for brevity of the equation.

As expected, equation (5.38) is linear in the varied branch coordinates $\delta\theta_1$, $\delta\theta_2$ and $\delta s$. The coefficient of $\delta s$ is zero, which is why an expression for $\delta s$ was not presented in Step 2.

**Step 4: Extract Kinetic Equations**

Since the second approach has been adopted in the third step, one needs to incorporate the constraint effect in extracting the kinetic equations. That is, $\delta\theta_1$ and $\delta\theta_2$ in equation (5.38) are not independent of each other; their interdependence is expressed by equation (5.34). Substituting this relationship into equation (5.38), and considering the arbitrariness of $\delta\theta_1$, one has

$$\begin{aligned}&\left[m_2\ddot{\underset{\sim}{r}}_2 \cdot (\dot{k} \times \underset{\sim}{r}_{23}) - I_2\ddot{\theta}_{12}\right] J_1 \\ &+T + m_1\ddot{\underset{\sim}{r}}_1 \cdot \dot{k} \times \underset{\sim}{r}_{21} - I_1\ddot{\theta}_1 \\ &-m_2\ddot{\underset{\sim}{r}}_2 \cdot \left[\dot{k} \times (\underset{\sim}{r}_{22} - \underset{\sim}{r}_{21} - \underset{\sim}{r}_{23})\right] - I_2\ddot{\theta}_{12} = 0 \end{aligned} \tag{5.39}$$

as the system kinetic equation. Note, there are no joint reactions appearing in it.

Equations (5.32) and (5.39) together give three scalar equations in DAE form, containing the three unknowns $\theta_1$, $\theta_2$, and $s$. Therefore, they constitute a necessary and sufficient set of system equations for the slider-crank mechanism under consideration. The approach presented here produces only one ordinary differential equation, equal in

number to the system degrees of freedom. The total number of system DAEs obtained is $T_d$, where $T_d$ is the number of branch coordinates. This observation is true of other general systems with or without closed kinematic chains.

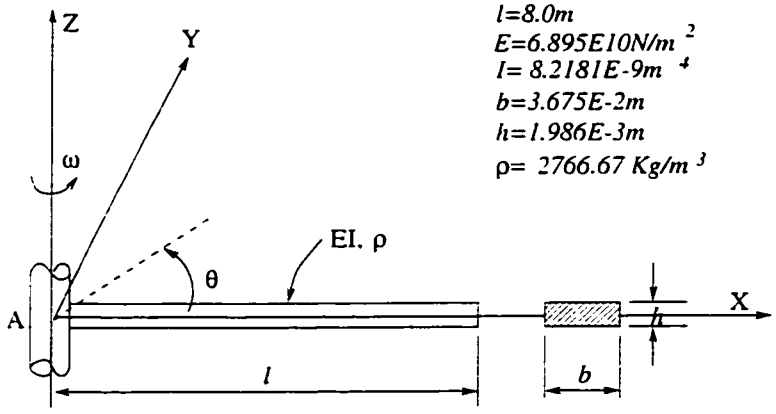
For most systems with closed loops, Approach 4.2 effects a reduction in the number of system equations since it eliminates the reactions associated with the loop closure joints. This issue is further discussed later in Chapter 7.

### 5.4.3 Motion-Driven Rotating Flexible Beam

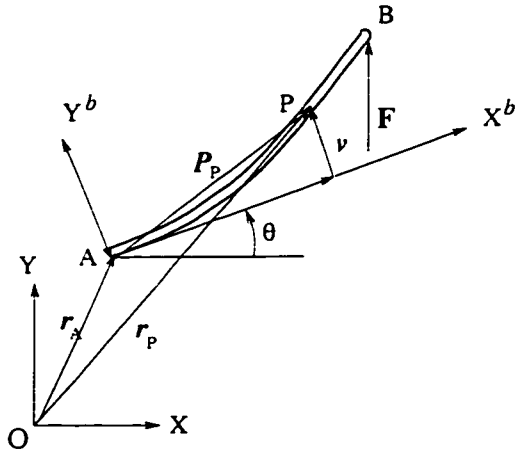
Shown in Figure 5.9 is a Bernoulli-Euler beam built into a rigid shaft which is driven with an angular displacement $\theta(t)$ about the $Z$ axis. The beam is treated the same way as in Chapter 4 except that the geometric stiffening effect is neglected since neither the foreshortening of the beam is included in the its deformation field approximation nor are second-order elastic forces retained in its kinetic equations. Deformation variables are discretized with Taylor monomials. This example serves only to show that the formulation procedure works equally well for systems containing flexible bodies. Some more involved examples are solved with more advanced models in Chapter 7.

The system graph is constructed first, as shown in Figure 7.10. Edge $e31$ models the motion driver. Edge $e11$ represents the flexible beam. Edge $e61$ together with edges $e71$ and $e72$ represents the pin joint. Edge $e72$ also serves to locate the position of the pin joint relative to the origin of the inertial frame. In fact, it now can be seen as a combined edge from a VW dependent edge and an inertially-fixed body arm. Its through variables are the sum of those of its two constituent edges. Its across variables are the same as those for any one of the two edges.

From this graph, equations of motion are systematically obtained using the same formulation procedure as outlined before, with the edges in bold selected into the tree (a body tree). This selection of tree means that the final system equations will be expressed

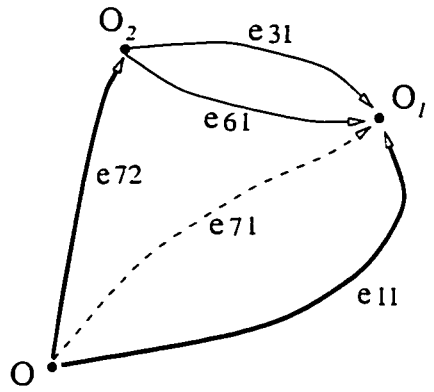



(a)




(b)

Figure 5.9: The Flexible Beam and Reference Frames




*Elements:*
*e11: the beam*
*e31: the motion driver*
*e61: the joint edge*
*e71,e72: the dependent virtual work element*

Figure 5.10: System Graph for the Motion-Driven Beam

in terms of the across variables of edge $e11$ and the elastic coordinates of the beam $\mathbf{q}_f$.

**Step 1: Eliminate Secondary Across Variables**

For simplicity, let

$$\underset{\sim}{r}_{11} \triangleq \underset{\sim}{R} \tag{5.40}$$

and

$$\theta_{11} \triangleq \theta \tag{5.41}$$

This means that the configuration of the body frame $X^bAY^b$ is specified with the symbols $\underset{\sim}{R}$ and $\theta$ now, and that the branch coordinates $\mathbf{q} = [R_x \;\; R_y \;\; \theta \;\; \mathbf{q}_f^T]^T$.

Using circuit equations for chords, cotree across variables except those already known for motion drivers and joints are expressed as functions of branch coordinates, for possible later use. After substituting the relevant terminal equations, one has

$$\theta_{61} = \theta_{11} \overset{(5.41)}{=} \theta \tag{5.42}$$

$$\underset{\sim}{r}_{71} = \underset{\sim}{r}_{11} \overset{(5.40)}{=} \underset{\sim}{R} \tag{5.43}$$

$$\theta_{71} = \theta_{11} \overset{(5.41)}{=} \theta \tag{5.44}$$

$$\delta\theta_{31} = \delta\theta_{11} \overset{(5.41)}{=} \delta\theta \tag{5.45}$$

for edges $e31$, $e61$ and $e71$. Note that $\delta\theta_{31}$ is not zero even though $\theta_{31}$ is a specified function of time; as mentioned earlier in Section 3.4, its varied across variable is calculated as a linear combination of the varied branch coordinates.

**Step 2: Generate Kinematic Constraint Equations**

Constraint equations on branch coordinates are derived from circuit equations corresponding to cotree joints and cotree motion drivers. After substituting in the terminal equations of relevant elements, one has

$$-\underline{r}_{11} + \underline{r}_{72} = 0 \tag{5.46}$$

from the pin joint element. From this and given that $\underline{r}_{72} = 0$ and equation (5.40), one obtains:

$$\begin{cases} R_X &= 0 \\ R_Y &= 0 \end{cases} \tag{5.47}$$

The circuit equation for the motion driver element $e31$ gives:

$$\theta_{11} \overset{(5.41)}{=} \theta = \theta_{31}(t) \tag{5.48}$$

where $\theta_{31}(t)$ is the angular displacement imposed on the shaft by the motion driver element $e31$.

Equations (5.47) and (5.48) are the system constraint equations $\mathbf{C}(\mathbf{q},t) = 0$. Constraint equations at the velocity and acceleration level can be obtained in a similar fashion.

**Step 3: Generate System Virtual Work Equation**

Using virtual work as a through variable, one has from the cutset equation for the body element $e11$

$$\delta W_{11} + \delta W_{31} + \delta W_{61} + \delta W_{71} = 0 \tag{5.49}$$

where $\delta W_{61} = 0$ by definition, and $\delta W_{11}$, $\delta W_{31}$ and $\delta W_{71}$ are evaluated from their terminal equations defined in Section 3.4 as follows.

First the VW of the beam is defined to be

$$\begin{aligned} \delta W_{11} &= -\int_V \delta\boldsymbol{\epsilon}^T \boldsymbol{\sigma}\, dV + \int_V \delta \mathbf{r}_P^T (\mathbf{f}_b - \rho\, \ddot{\mathbf{r}}_P) dV \\ &= \delta W_e + \delta W_b \end{aligned} \tag{5.50}$$

where

$\delta W_e = -\int_V \delta\boldsymbol{\epsilon}^T \boldsymbol{\sigma}\, dV$ is the virtual work of internal elastic forces,

$\delta W_b = \int_V (\mathbf{f}_b - \rho\, \ddot{\mathbf{r}}_P) dV$ is the virtual work of body forces, including inertial forces.

Note that the general definition for $\delta W_{11}$ given in Section 3.4 is utilized.

For the purpose of this example, an approximation of the beam's deformation field in the body frame is taken as

$$\begin{cases} u(x,t) &= 0 \\ v(x,t) &= q_1(t)\, N_1(x) + q_2(t)\, N_2(x) \\ &= q_1(t)\, x^2 + q_2(t)\, x^3 \end{cases} \tag{5.51}$$

where $u(x,t)$ and $v(x,t)$ are respectively the deformations along the $X^b$ and $Y^b$ axes of the body reference frame, $q_i(t)$ are the generalized elastic coordinates that are only functions of time, $N_i(x)$ are the shape functions of the approximation, and $x$ is the coordinate of a typical point on the neutral axis in the undeformed state.

From this deformation field, one has for stress and strain [71]:

$$\begin{cases} \sigma_x &= -E\, y_b\, \mathcal{K} \\ \varepsilon_x &= \dfrac{\sigma_x}{E} = -y_b\, \mathcal{K} \end{cases} \tag{5.52}$$

where $y_b$ is the $y$ coordinate of the point in question on a typical beam cross section, $E$ is the modulus of the beam material, and $\mathcal{K}$ is the curvature of the neutral axis given by

$$\mathcal{K} = \frac{d^2v}{dx^2} = q_1(t)\; N_1''(x) + q_2(t)\; N_2''(x) = 2\, q_1 + 6\, x\, q_2 \tag{5.53}$$

Therefore,

$$\begin{aligned} \delta W_e &= -\int_V \delta\boldsymbol{\varepsilon}^T \boldsymbol{\sigma}\, dV \\ &= -EI[\delta q_1 \;\; \delta q_2] \int_0^l \begin{bmatrix} (N_1'')^2 & N_1'' N_2'' \\ N_2'' N_1'' & (N_2'')^2 \end{bmatrix} dx \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} \\ &= -EI[\delta q_1 \;\; \delta q_2] \begin{bmatrix} 4l & 6l^2 \\ 6l^2 & 12l^3 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} \\ &= -\delta \mathbf{q}_f^T\, \mathbf{K}_f\, \mathbf{q}_f \end{aligned} \tag{5.54}$$

where $I = \int_S y^2 dy\, dz$ is the second moment of the beam's cross-sectional area; $\mathbf{K}_f$ is the stiffness matrix; and

$$\delta \mathbf{q}_f = \begin{Bmatrix} \delta q_1 \\ \delta q_2 \end{Bmatrix}, \quad \mathbf{K}_f = EI \begin{bmatrix} 4l & 6l^2 \\ 6l^2 & 12l^3 \end{bmatrix}, \quad \mathbf{q}_f = \begin{Bmatrix} q_1 \\ q_2 \end{Bmatrix} \tag{5.55}$$

To evaluate $\delta W_b$, for a typical point on the beam, one can write from kinematics (see

Appendices A.1 and A.2)

$$\delta \mathbf{r}_P = \begin{bmatrix} 1 & 0 & -(x^2 q_1 + x^3 q_2)\cos\theta - x\,\sin\theta & -x^2\sin\theta & -x^3\sin\theta \\ 0 & 1 & -(x^2 q_1 + x^3 q_2)\sin\theta + x\,\cos\theta & x^2\cos\theta & x^3\cos\theta \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \delta R_X \\ \delta R_Y \\ \delta\theta \\ \delta q_1 \\ \delta q_2 \end{Bmatrix}$$

$$= \mathbf{L}_1 \delta \mathbf{q} \tag{5.56}$$

and

$$\ddot{\mathbf{r}}_P = \mathbf{L}_1 \ddot{\mathbf{q}} + \mathbf{L}_2 \tag{5.57}$$

in which

$$\mathbf{L}_2 = \begin{Bmatrix} -\dot{\theta}^2[x\cos\theta - (x^2 q_1 + x^3 q_2)\sin\theta] \\ -\dot{\theta}^2[x\sin\theta + (x^2 q_1 + x^3 q_2)\cos\theta] \\ 0 \end{Bmatrix} + 2 \begin{Bmatrix} -\dot{\theta}(x^2\dot{q}_1 + x^3\dot{q}_2)\cos\theta \\ -\dot{\theta}(x^2\dot{q}_1 + x^3\dot{q}_2)\sin\theta \\ 0 \end{Bmatrix} \tag{5.58}$$

is the sum of the Coriolis and the centripetal accelerations.

Therefore,

$$\begin{aligned} \delta W_b &= -\int_V \delta \mathbf{r}_P^T \rho\, \ddot{\mathbf{r}}_P dV \\ &= -\int_V \delta \mathbf{q}^T \mathbf{L}_1^T \rho\, (\mathbf{L}_1 \ddot{\mathbf{q}} + \mathbf{L}_2) dV \\ &= -\delta \mathbf{q}^T (\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Q}_v) \end{aligned} \tag{5.59}$$

in which

$$\mathbf{M} = \int_V \rho \mathbf{L}_1^T \mathbf{L}_1 dV \quad \text{is the symmetric mass matrix,}$$

$$\mathbf{Q}_v = \int_V \rho \mathbf{L}_1^T \mathbf{L}_2 dV \quad \text{is the quadratic velocity term.}$$

Thus, it follows from equations (5.54) and (5.59)

$$\begin{aligned} \delta W_{11} &= -\delta \mathbf{q}_f^T \mathbf{K}_f \, \mathbf{q}_f - \delta \mathbf{q}^T (\mathbf{M}\ddot{\mathbf{q}} + \mathbf{Q}_v) \\ &= -\delta \mathbf{q}^T (\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{Q}_v) \end{aligned} \tag{5.60}$$

where

$$\mathbf{K} = EI \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4l & 6l^2 \\ 0 & 0 & 0 & 6l^2 & 12l^3 \end{bmatrix} \tag{5.61}$$

For $\delta W_{31}$, one has by definition,

$$\delta W_{31} = T_{31} \delta \theta_{31} \overset{(5.45)}{=} T_{31} \, \delta \theta = \delta \mathbf{q}^T \, \mathbf{T} \tag{5.62}$$

where $\mathbf{T} = [0 \;\; 0 \;\; T_{31} \;\; 0 \;\; 0]^T$.

For $\delta W_{71}$, one has again by definition,

$$\delta W_{71} = \delta \underset{\sim}{r}_{71} \cdot \underset{\sim}{F}_{61} + \delta \theta_{71} T_{61} \tag{5.63}$$

where $T_{61} = 0$ due to the pin joint at point O, and from equation (5.43),

$$\delta W_{71} = \delta \underset{\sim}{r}_{11} \cdot \underset{\sim}{F}_{61} = \delta r_{11}^T \mathcal{R}_1^1 \, \mathbf{F}_{61}^b = \delta \mathbf{q}^T \mathbf{S}_0 \mathcal{R}_1^1 \, \mathbf{F}_{61}^b \tag{5.64}$$

where

$$\mathbf{S}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{5.65}$$

$\mathbf{F}_{61}^b$ is the column matrix of the components of the pin joint force expressed in the body reference frame; $\mathcal{R}_1^1$ is the rotation transformation matrix from the body frame to the inertial frame, given by:

$$\mathcal{R}_1^1 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.66}$$

Finally the VW equation of the system expressed in the varied branch coordinates is obtained by summing up equations (5.60), (5.62) and (5.64) to give

$$-\delta\mathbf{q}^T(\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{Q}_v - \mathbf{T} - \mathbf{S}_0\mathcal{R}_1^1\mathbf{F}_{61}^b) = 0 \tag{5.67}$$

**Step 4: Extract Kinetic Equations**

With the arbitrariness of $\delta\mathbf{q}$, equation (5.67) produces

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{S}_0\mathcal{R}_1^1\mathbf{F}_{61}^b + \mathbf{T} - \mathbf{Q}_v \tag{5.68}$$

which are the set of kinetic equations of the motion-driven beam. As pointed out in Step 4 of Section 5.3, joint reactions $\mathbf{F}_{61}$ appear in the kinetic equations since the body tree has been adopted and system kinetic equations were extracted without substituting the joint constraint equation (5.47). The torque associated with the motion driver also appears in the kinetic equation for exactly the same reason.

Equations (5.68), along with equations (5.47) and (5.48), constitute the system equa-

tions that govern the motion of the beam. These equations, a set of DAEs, contain eight independent scalar equations and the eight unknown scalar quantities, $R_X$, $R_Y$, $\theta$, $q_1$, $q_2$, $F^b_{61X}$, $F^b_{61Y}$ and $T$, which specify the dynamic state of the rotating beam system. Therefore, they form a necessary and sufficient set of system equations for the beam.

## 5.5 Concluding Remarks

A solid and feasible procedure for generating dynamic equations with the new VW graph-theoretic approach for multibody systems has been presented. The procedure is able to deal with systems that may contain both rigid and flexible bodies with or without closed loops when a proper choice of the formulation tree is made. Its validity has been proved with three examples.

The VW dependent elements for any ideal joint can be omitted in the system graph when the joint constraint equations at the virtual displacement level are substituted into a single system VW equation to extract the kinetic equations, since the VW of these dependent element sums to zero or is constantly zero.

It helps understand the GT approach better to point out that the first three steps are based solely on the graph and terminal equations of the components involved while the final fourth step is a mathematical manipulation of the equations obtained from past steps. This in fact is the way the graph-theoretic approach has been applied to other general physical systems [74].

The selection of a formulation tree to aid in generating system equations plays a very important role in dictating the form and nature of the final set of system equations of motion [64]. Although it is desired that from a graph-theoretic point of view cutset and circuit equations be written to generate correct system equations for any given tree. Regrettably this is not always the case when it comes to its application to multibody

mechanical systems. For example, picking arm elements into the cotree will mar the established traditional equation-generation procedure, since the across variables of the dependent torque will appear in the final system equations along with the across variable of the body on which the arm resides as primary variables, which is rarely desirable. Picking the VW dependent edge into the tree in the present formulation will cause similar problems too. Thus, it is obvious that not all tree selections allow for a systematic formulation procedure. The root cause lies with the dependent nature of the across variables associated with these dependent elements.

It is worth mentioning too that the present formulation draws on the traditional GT procedure, with some modifications. For example, in the traditional procedure the chord transformation is needed so that cotree through variables, instead of tree through variables, will appear in the final system kinetic equations. This transformation is not used here since there are no tree through variables of whatever type appearing in the system VW equations. The traditional procedure fails to eliminate joint reaction forces from its dynamic equations for closed loop systems (unless married with the projection method [103]) while the present one accomplishes this almost effortlessly.

Finally, the significance of the proposed GT approach is that the adoption of virtual work as a through variable makes it possible to bring the traditional formulation of multi-body dynamics under the framework of graph theory, resulting in a simple, systematic, and above all, a unified modelling procedure that is amenable to computer implementation, no matter which kinematic description scheme is used for modelling the elastic deformation of the flexible bodies in a system. In addition, this new approach provides a general method for formulating the governing equations of systems containing both rigid and flexible bodies, as can be seen from the three examples and more to come later in the thesis.

# Chapter 6

# Computer Implementation and Numerical Solution

Although a new GT approach has been proposed and validated to some extent in previous chapters, and its application is quite procedural, it is still a tedious task to manually apply this formulation to more complex systems, such as three-dimensional ones and systems with flexible beams. Thus, a computer implementation of the procedure is most desirable.

In this chapter, a Maple implementation of the equation generation steps discussed in the preceding chapter is described with details on all the major blocks of Maple scripts used to effect the steps. Specifically, an explanation is given on the procedure by which the Maple part of the package DynaFlex is created, based upon the joint tree formulation, with the loop closing constraint equations substituted in the process of extracting system kinetic equations. Since all arms and as many as possible joints are selected into the tree for this formulation, the system equations are expressed in the relative coordinates of the joints selected into the tree and are free from any joint constraint reactions or Lagrange multipliers, even for systems with closed kinematic chains. In the case of systems with

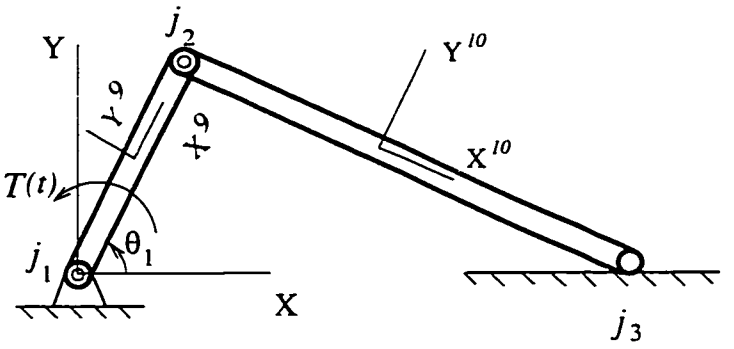



Figure 6.1: The Slider-Crank

flexible beams, these equations are in terms of the joint coordinates, and the beams' generalized elastic coordinates, which can be regarded as the across variables for the flexible arms involved.

The unique structure of these Lagrange multiplier free (LMF) system DAEs is then examined, and some numerical solution methods that are suitable for use with them are proposed.

## 6.1 Symbolic Computer Implementation for Joint Tree Formulation

Throughout this section, use is made of the slider crank mechanism presented in Chapter 5 to help understand the explanation. The mechanism is repeated here for easy reference, as shown in Fig. 6.1. To make the final system equations readily tangible, in what follows it is assumed that the link has zero mass, and a torque $T(t)$ is acting on the crank. The graph for the slider crank is shown in Fig. 6.2. Note that edges are labelled arbitrarily and not according to their type, since the implementation does not require that.

In the graph, $e9$ is a rigid body element representing the $j1j2$ body while edges $e2$ and $e3$ are rigid arms needed to locate the joint positions on it; the same is done with

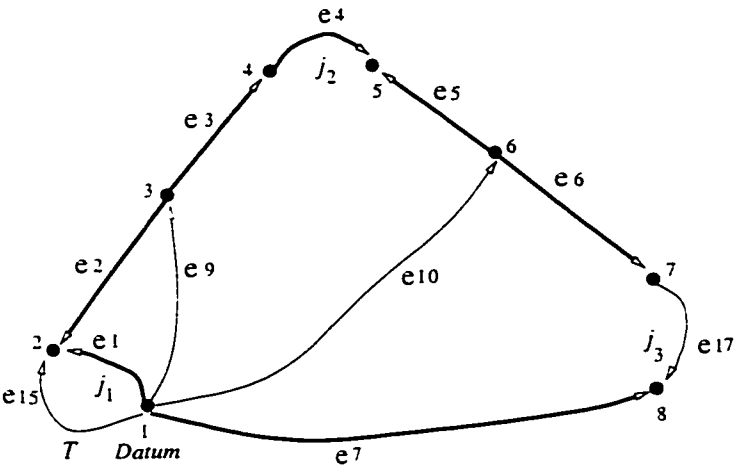



Figure 6.2: Graph of the Slider-Crank

the body $j2j3$. Joint $j1$ is represented by the joint edge $e1$ whereas the torque driver is modelled with a torque driver edge $e15$ parallel to the joint edge. Joint $j2$ is represented by joint edge $e4$ and the sliding joint is modelled with the sliding arm element $e7$ and the pin joint edge $e17$. VW edges have not been shown in the figure, for clarity.

Based upon the procedure outlined in the previous chapter for generating system equations, the implementation of DynaFlex follows the flowchart given in Fig. 6.3. There are many tasks and sub-tasks that can be divided into five fundamental modules: the System Input File Module, the Topological Analysis Module, the Branch Transformation Module, the Kinematic Constraint Module, and the Dynamic Module. Each of the modules corresponds to one heavy box in the chart in the same order. Before delving into other details, the input file for a system is described.

### 6.1.1 System Input File Format

As the individual graph edges are the building blocks of the graph theoretical model of a system, it is natural to describe the physical system in terms of the graph edges it contains. For this reason, DynaFlex requires as input the information about each of the

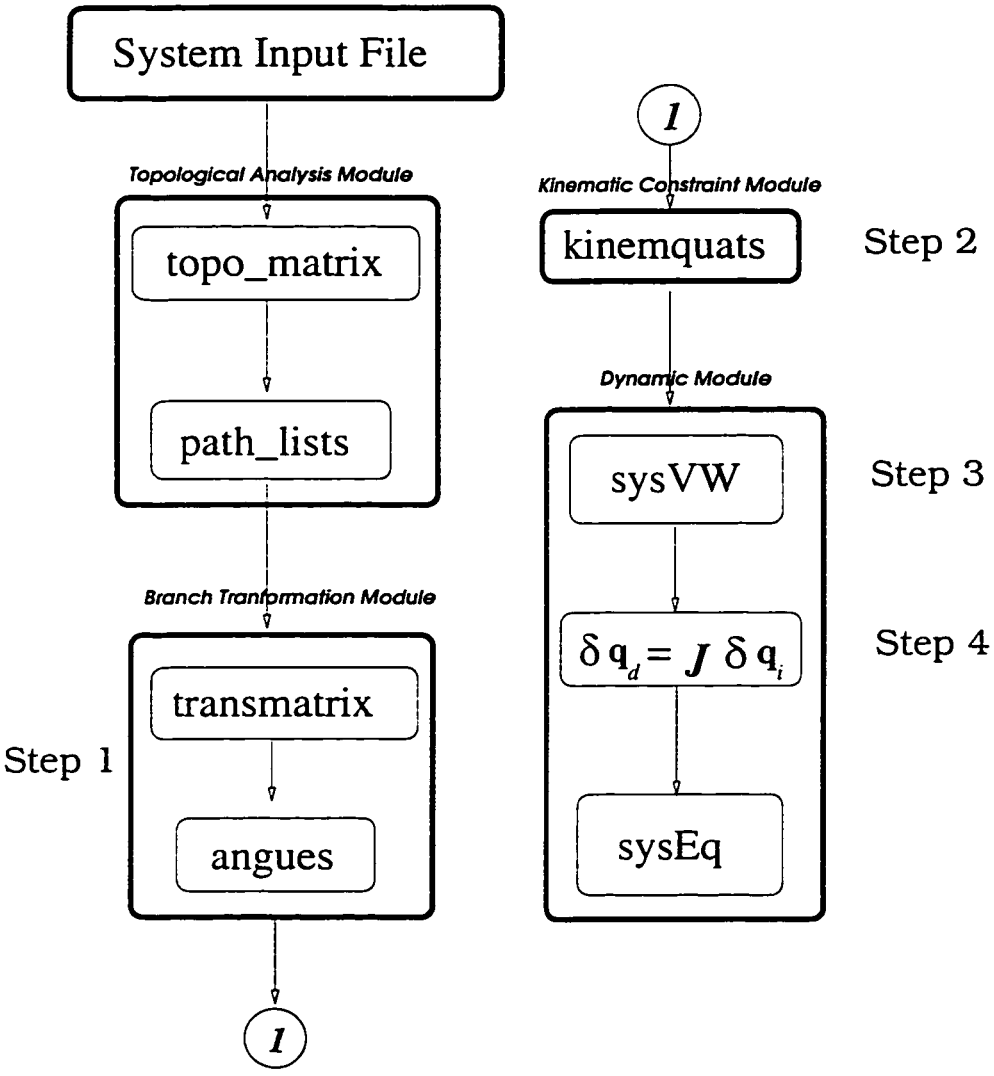



Figure 6.3: The Flowchart of DynaFlex

edges contained in a system graph.

It is my experience that Maple offers an extremely versatile data structure — the table, which is very suitable for implementing the chosen format of the input file. The required information for an edge numbered $i$ is specified using the *table* command with the word *edge* as its name in the form:

$$\begin{aligned} edge[i] \;:=\; \text{table}([\; & 1 = \cdots, \\ & 2 = \cdots, \\ & \vdots \\ & ]) \end{aligned}$$

where $i$ can be any number the user chooses.

The label on the left hand side of the equal sign in the table is called the index of the table, while the label on the other side is the value of the corresponding index. This table is referred to here as the *edge table* for easy reference. An edge table may contain a number of indices and other sub-tables as its value (see examples below).

Although the exact content and format of an edge table depends on the type of the edge so that DynaFlex is able to distinguish and perform different operations on them, as required in the formulation, there are some general rules to follow.

Index 1 is always assigned with either the letter $Y$ or $N$, which indicates that the current edge is either in the tree or in the cotree respectively, index 2 is always assigned a two-entry list whose first entry indicates the start node of the current edge while the second the end node, and index 3 is assigned an identifier that indicates the exact type of the edge, such as JE for joint type, FDE for force driver type and MDE for motion driver type. The only thing that requires more explanation is the value assigned to index 4. For a joint, it specifies the type of the joint, such as RV for revolute (pin) joint, SPH for spherical joint, UNIV for universal joint and so on. For a rigid body element, it

takes on the inertia properties about the center of mass. For a flexible body element, it specifies a list of the number of modes to be used for discretization, the name of the discretizing monomials, and the orders to which the beam's inertial and elastic virtual work are evaluated. For a rigid arm, this index signifies the local coordinates of the arm and the relative rotation matrix from a frame at its end node to its start node frame, which is the reference frame of the rigid body on which the arm resides. For a flexible arm, the same information is specified, but in the undeformed state of the beam now. The default for an arm's relative rotation matrix is a $3 \times 3$ unit matrix. For a force or motion driver, it contains the sub-type (such as JD for a joint driver when it acts on a joint or PD for an inertial driver when it acts from the inertial frame), the driven coordinates or forces, and the corresponding functions specified. For easy reference, the build-up of the edge table for an element is specified in Tables 6.1 and 6.2.

For example, for a rigid body edge selected into the cotree and numbered 9 that starts from Datum node numbered 1 and ends at node 3, (see the edge representing the crank in the slider example,) the input edge table assumes the following form:

$$
\begin{aligned}
&edge[9] := \text{table}([\\
&\quad 1 = N,\\
&\quad 2 = [1, 3],\\
&\quad 3 = BE\_R,\\
&\quad 4 = \text{table}([\\
&\quad inert = [[I91, I92, I93], [I94, I95, I96], [I97, I98, I99]],\\
&\quad mass = s \qquad\qquad ]) \qquad\qquad ])
\end{aligned}
$$

An edge can have the same labelling number as a node; DynaFlex will not confuse them. In the above edge table, the inertia properties of the rigid body are represented by index 4, which itself is a table. This sub-table has two indices: *inert* and *mass*. The

| Index<br>Value<br>Element | 1 | 3 | 4 | Comments |
|---|---|---|---|---|
| Rigid Body | N | BE_R | *inert=*<br>*mass=* | Inertia matrix<br>Body mass |
| Flexible Beam | N | BE_F | $[a, b, c, d]$<br>*Scheme=*<br>*Order_Felastic=*<br>*Order_Finert=* | # of monomials for each deformation variable<br>Type of discretization monomials (*Taylor, Cheby, or Legen*)<br>Order of elastic forces<br>Order of inertial forces |
| Rigid Arm | Y | AE_R | *Coords=*three-entry list | Components in body frame |
| Flexible Arm | | AE_F | *rotmat=* 3 three-entry lists | Arm's rotation matrix |
| Sliding Arm | | AE_S | *Coords=* six-entry list<br>*rotmat=* 3 three-entry lists | first three: components of sliding vector in body frame; second three: location of point on sliding axis<br>Rotation matrix from end to start node |
| Motion Driver | N | MDE | *type=JD*<br>*theta=*<br>*alpha=*<br>*beta=* | acts parallel to joint's $\theta$, $\alpha$ or $\beta$ degree |
| | | | *type=DD*<br>*dis=* | fixes distances between points on two bodies<br>specifies the distance |

Table 6.1: Specification for Edge Tables

| Index<br>Value<br>Element | 1 | 3 | 4 | Comments |
|---|---|---|---|---|
| Force Driver | N | FDE | *type=PD* | acts from inertial frame |
| | | | $f_x =$ | Force components |
| | | | $f_y =$ | |
| | | | $f_z =$ | |
| | | | $m_x =$ | Moment components |
| | | | $m_y =$ | |
| | | | $m_z =$ | |
| | | | *force* =*gl* or *lc* | frame in which $f_i$ is specified (use *gl* for global, *lc* for local) |
| | | | *moment* =*gl* or *lc* | frame in which $m_i$ is specifies (use *gl* for global, *lc* for local) |
| | | | *type* = *JD* | SDA acting parallel to joint's $\theta$, $\alpha$ or |
| | | | *theta* = $(k, d, f(t))$ | $\beta$ degree |
| | | | *alpha* = $(k, d, f(t))$ | |
| | | | *beta* = $(k, d, f(t))$ | |
| SDA | N | FDE | *type=LSD*<br>*character=(k,d,f(t))* | Linear SDA acting between any two bodies |
| Revolute (Pin) Joint | Y | JE | *type=RV* | rotates about $Z$ axis by $\theta$ |
| Universal Joint | or | | *type=UNIV* | rotates about $Y$ and $Z$ axes by $\alpha$ and $\beta$ respectively |
| Spherical Joint | N | | *type=SPH* | with 3-1-3 sequence of Euler angles, represented by $\zeta$, $\eta$, and $\xi$ respectively |
| Cylindrical-Revolute | | | *type=CR* | rotates about $X$ and $Z$ axis by $\alpha$ and $\beta$ respectively |
| Weld Joint | | | *type=WELD* | |
| Free Joint | Y | | *type=FREE* | |

Table 6.2: Specification for Edge Tables (Cont'd)

*inert* index specifies the inertia matrix of the body with respect to the body's reference frame that is fixed at its center of mass, with a list of lists. This list will be converted into a Maple matrix by DynaFlex for proper use. The *mass* index specifies the mass of the body.

To build the sub-table, the user does not actually need to specify the symbolic values for each of the entries. He only needs to type in a non-zero alphanumeric value in the proper place if this entry is not zero. DynaFlex will automatically set this entry to a pre-defined symbolic value. For example, for the body numbered 9, its mass will appear in system equations as $m9$ while its inertia components appear as $I91$, $I92$, $\cdots$, $I99$, by which the inertia matrix is defined

$$[I9] = \begin{bmatrix} I91 & I92 & I93 \\ I94 & I95 & I96 \\ I97 & I98 & I99 \end{bmatrix} \tag{6.1}$$

In the case when an entry is zero, the user can replace it with a zero; otherwise it will take on a corresponding symbolic value. Failing to enter a zero in the right place may result in meaningless equations in some cases and affect other DynaFlex operations. Therefore, entering zeros in the right places ensures meaningful results. Besides, it cuts down on the $CPU$ time and simplifies the final equations at the same time.

It is realized that this way of symbolically representing the entries of the inertia tensor is cumbersome, especially for bodies labelled with a multiple-digit number , but a better naming has yet to be found. In fact, how to name quantities in the context of symbolic computation in a way that is simple and as close in form as possible to what humans do by hand is not an easy question to answer, as discovered in my experience with Maple.

The polynomial types that are available for discretizing the deformation variables of a beam include the Taylor, Chebyshev, and Legendre polynomials. The highest order

allowed for both the inertial and elastic virtual work of a beam is 3.

In addition to specifying data for each of the edges in the system graph, DynaFlex requires as input the total number of edges and nodes used in the system graph, the number representing the datum node and a list, named *Iedge*, of the labelling numbers of the joint edges whose relative coordinates are to be used as the system independent coordinates in extracting kinetic equations when closed kinematic loops are involved. DynaFlex uses the numbers of edges and nodes specified by the user only as a check to make sure the user has specified the right number of them. It counts them itself, and when these two sets of counts are not in agreement with each other, it prompts the user to make one more check on them. For a system without closed kinematic loops, the independent coordinate list *Iedge* is assigned a NULL list by the user.

For the slider-crank, *Iedge:=[1]*, meaning that $\theta_1$ is identified as the independent branch coordinate for the one-degree-of-freedom system.

The current version of DynaFlex is capable of handling all the elements discussed in Section 3.4. The sub-class of motion drivers that drives the general rotation of a reference frame (or body), however, is not available for now. Only rotation drivers associated with joints are available, as described in Table 6.1.

### 6.1.2 Topological Analysis Module

Once DynaFlex reads in the system input file created by the user, it sorts out the topological structure of the system. The analysis includes building up the incidence matrix of the system graph, the most essential foundation on which DynaFlex's algorithm is based, from which the cutset matrix and the circuit matrix are obtained for use in generating system kinetic and kinematic equations respectively.

Instead of employing the built-in Maple *new* command to create a Maple graph (which will be exactly the same as the user has created for the system) and employing the

*incidence* command afterwards to create the incidence matrix from it, DynaFlex creates the incidence matrix directly by checking each edge in the input file and making use of their start and end nodes. It is more efficient this way in view of the style used for the input file.

As a general rule, the incidence matrix is built up with the branches corresponding to its first columns, the virtual work edges to the last ones, and with the chords to the middle columns. Note that the user is not required to draw any virtual work edges; DynaFlex will automatically create them when there is a Spring-Damper-Actuator that does not act parallel to any joint (termed a "loose" SDA).

With the incidence matrix in place, DynaFlex generates the cutset matrix by simple row operations, which is accomplished by first deleting the last row of the incidence matrix and then calling the Maple *gauss-jordan* elimination procedure. The circuit matrix is created from the cutset matrix, using their orthogonality, see [74].

For the slider-crank these matrices are obtained after the Maple script is run as

$$INCIDENCE := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

with the corresponding branch, chord and node lists:

$$branch\_list := [1, 2, 3, 4, 5, 6, 7]$$

$$chord\_list := [9, 10, 15, 17]$$

$$node\_list := [1, 2, 3, 4, 5, 6, 7, 8]$$

In view of the importance of these matrices for the procedure, an error check is done on the $CUTSET$ matrix. It looks into every element it has and searches for anything that is neither a $-1$, 1 or 0. Once such an entry is detected, DynaFlex alerts the user to it.

For general three-dimensional problems, where large rotations are no longer commutative, not only does one need to know which edges are involved in which circuit, one also needs to know the sequence in which the edges in a circuit are connected to one another and their directions, which represent the relative orientation of their kinematic quantities. Although the circuit matrix tells the former, it does not tell the latter explicitly. Thus,

DynaFlex proceeds further to extract this information, producing what is called *path lists* and *direction lists.*

In order to understand what DynaFlex does to obtain the sought-after sequence and the accompanying directional information, the term *tree top nodes* or simply *top nodes* in introduced. A tree top node is a node on which there is only one branch incident. In other words, it is the outermost end node of a path comprised of branches only and starting from the Datum. For example, in the graph of the slider, nodes 7 and 8 are the tree top nodes. From a topological point of view, top nodes of a graph can be extracted from the part of the incidence matrix that corresponds to the branches. DynaFlex identities top nodes this way with the following script:

```
> # extract the top vertices of the tree:
> vert1:=NULL:  # to keep the tree vertices that have only
>               # one branch incident on it.
> for i to nops(node_list) do
>   AA:=submatrix(INCIDENCE,i..i,1..nops(branch_list));
>   seq1:=NULL:
>   for j to nops(branch_list) do
>    if AA[1,j]<>0 then
>     seq1:=seq1,AA[1,j];
>    else next;
>    fi;
>   od:
>
>   if nops([seq1])=1 then
>   vert1:=vert1,node_list[i];
>   fi;
> od:
>
> vert:='minus'({vert1},{Datum}): # eliminate Datum node from set of
> tree_top_nodes:=vert;    # one-degree vertices if Datum is among them.
> # Variable vert is going to be taken as global variable!
```

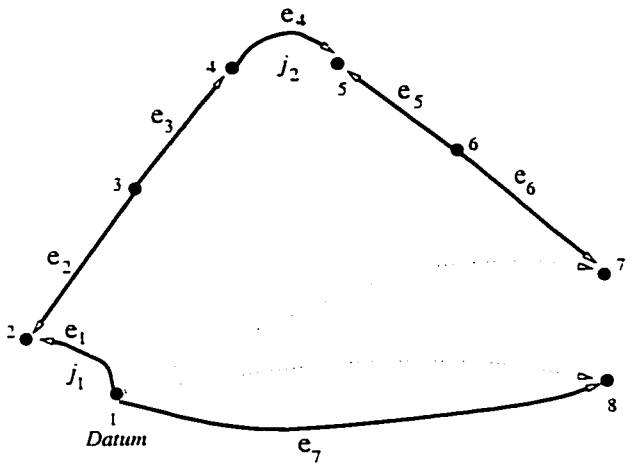



Figure 6.4: Auxiliary Graph of the Slider-Crank

What this portion of Maple code does is look at each of the nodes and see how many branches it has incident on it. The nodes that have only one branch incident on it are the tree top nodes.

Once the top nodes are identified, DynaFlex constructs an auxiliary graph in the form of its incidence matrix to assist determining the connectivity of the branches on a path from the datum to a tree top node. This auxiliary graph retains all the branches of the original system graph and has one chord added from the Datum to each of the tree top nodes. For the slider, the auxiliary graph is shown in Figure 6.4.

The incidence matrix of this graph is then constructed and manipulated in the same way as the original system incidence matrix to give its cutset and circuit matrix. (Note that the auxiliary graph itself is of no interest. It is its incidence matrix that is of interest.) Now there is one circuit per top node and the branches contained in each of the circuits are those that comprise the path from the Datum node to a tree top node. They are put in a list and then sorted for their connectivity along with their directional values through the following block of Maple script, which is run for each tree top node and corresponding branch list:

```
>     # Arrange branches in list list1 from tree
>     # top to datum node and their directional
>     # values in list list3 accordingly.
>     # Initial value of the variable node is the
>     # number of the tree top node.
>     seq1:=NULL;
>     seq3:=NULL;
>     while node<>Datum do  # When the new search node
>      aa:=nops(list1);  # is datum node, stop search.
>      for j to aa do
>       # If current branch is the sought one,
>       # put it into the sequence seq1:
>       if (edge[op(j,list1)][(2)][1]=node) or
>        (edge[op(j,list1)][(2)][2]=node) then
>           seq1:=seq1,op(j,list1);
>           seq3:=seq3,op(j,list3);
>           # Then set the new search node:
>           if    edge[op(j,list1)][(2)][1]=node then
>                 node:=edge[op(j,list1)][(2)][2];
>            else  node:=edge[op(j,list1)][(2)][1];
>           fi;
>           # Updates list1:
>           list1:=subsop(j=NULL,list1);
>           # Updates list3:
>           list3:=subsop(j=NULL,list3);
>           break;  # Stop search in j-loop since the
>       else next; # right branch has been found and
>       fi;        # placed in right sequence.
>      od;
>     od;
```

Afterwards the order of the entries in the sequences *seq1* and *seq2* are reversed through the following script:

```
>     # Reverse seq1 and seq3 so that they  start
```

```
>     # from the branch incident on datum node.
>      seq1:=[seq1];
>      seq3:=[seq3];
>      n:=nops(seq1);
>      seq1:=[seq(seq1[n-j+1],j=1..n)];
>      seq3:=[seq(seq3[n-j+1],j=1..n)];
>      path.(vert[i]):=seq1;
>      direct.(vert[i]):=seq3;
```

The results are lists of paths and the corresponding directional values. For the slider example, one has the following results

$$tree\_top\_nodes := \{7, 8\}$$

$$path7 := [1, 2, 3, 4, 5, 6]$$

$$direct7 := [-1, 1, -1, -1, 1, -1]$$

$$path8 := [7]$$

$$direct8 := [-1]$$

Now that the path lists have been generated, another set of topological information that is very important is readily obtained. These are the paths that lead up to the non-datum node of each body edge and their directional values which are needed in preparation for the next steps.

This is done as follows. For each body element, edges in its circuit and the branches that are incident on its non-datum node are extracted. They are then operated on with an *intersect* command to single out the branch that is on the datum side of the path through the non-datum node of the body edge, and incident on it. This key branch is used afterwards to identify the previously found path list(s) that can be employed to extract the path leading up to the body's non-datum node, which is usually only a part

of the *path* list(s). For the slider, DynaFlex gives

$$Path9 := [1, 2]$$

$$direction9 := [-1, 1]$$

$$Path10 := [1, 2, 3, 4, 5]$$

$$direction10 := [-1, 1, -1, -1, 1]$$

for bodies numbered 9 and 10 respectively. In other words, edges 1 and 2 lead to the end node 3 of the body edge 9, with their directions given by *direction*9.

It is noted that Maple does not have a built-in command that works on a tree and produces the path that connects two given nodes and the directional values at the same time [110]. In fact, the direction of edges does not seem to count in its "networks" package.

### 6.1.3 Branch Transformation Module

Once the topology of the system is in place, DynaFlex goes on to calculate the transformation matrices of all the bodies' reference frames with respect to the inertial frame, their rotational velocities and accelerations, expressed in the inertial frame as well. These quantities are the basic ingredients for generating both kinematic and kinetic equations.

The transformation matrix of a body reference frame is obtained by searching up the *Path* list for the body and post-multiplying the transformation matrix of the elements on the path one after another in the order as they appear in the list *Path*.

In building up the transformation matrix, care must be taken as to the direction of the branches in the list because an edge's rotation transformation matrix is defined with respect to this direction. The transformation is multiplied as is when the branch directs away from the datum, i.e., the directional value is $-1$ in the *direction* list. Otherwise

its transpose is used. For example, in the present slider-crank, DynaFlex calculates the rotation matrix of bodies 9 and 10 as

$$\mathcal{R}_9 = \mathcal{R}_1 \mathcal{R}_2^T \tag{6.2}$$

$$\mathcal{R}_{10} = \mathcal{R}_1 \mathcal{R}_2^T \mathcal{R}_3 \mathcal{R}_4 \mathcal{R}_5^T \tag{6.3}$$

They can be regarded as circuit equations of a multiplicative nature, as mentioned in Chapter 3.

The angular velocity of a body reference frame is obtained in a similar way with a summation of angular velocities of all the branches, i.e., the branch transformation equation at the angular velocity level. To project the angular velocity onto the inertial frame, the transformation matrix of body frames found above is employed since the angular velocity terminal equation of an element is always defined with respect to some local frame.

The angular acceleration of a body reference frame is obtained by taking time derivative of its angular velocity.

In theory, it is possible to find the angular velocity and acceleration of a frame by operating on its rotation transformation matrix. In practise, however, my observation is that this leads to much lengthier and complicated system equations even after extensive and expensive simplification techniques are used. My experience shows, on the other hand, that deriving the acceleration directly from the velocity that has been obtained from built-in velocity terminal equations is justified in this regard, and angular accelerations of reference frame are derived in this manner.

Note also that although angular acceleration terminal equations of elements could have been substituted into the branch transformation equation to build up the angular acceleration of a frame, this would have demanded more programming effort than needed

in the present approach of taking time derivatives of velocities.

Again for the slider example, DynaFlex finds the rotation transformation matrices, the angular velocities and accelerations to be

$$Tr9 := \begin{bmatrix} \cos(\theta 1(t)) & -\sin(\theta 1(t)) & 0 \\ \sin(\theta 1(t)) & \cos(\theta 1(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Tr10 :=$$
$$[\cos(\theta 1(t))\cos(\theta 4(t)) - \sin(\theta 1(t))\sin(\theta 4(t)),$$
$$-\cos(\theta 1(t))\sin(\theta 4(t)) - \sin(\theta 1(t))\cos(\theta 4(t)), 0]$$
$$[\sin(\theta 1(t))\cos(\theta 4(t)) + \cos(\theta 1(t))\sin(\theta 4(t)),$$
$$\cos(\theta 1(t))\cos(\theta 4(t)) - \sin(\theta 1(t))\sin(\theta 4(t)), 0]$$
$$[0, 0, 1]$$

$$V\_ang9 := \left[ 0\ 0\ \frac{\partial}{\partial t}\theta 1(t) \right]$$

$$W\_ang9 := \left[ 0\ 0\ \frac{\partial^2}{\partial t^2}\theta 1(t) \right]$$

$$V\_ang10 := \left[ 0\ 0\ \left(\frac{\partial}{\partial t}\theta 1(t)\right) + \left(\frac{\partial}{\partial t}\theta 4(t)\right) \right]$$

$$W\_ang10 := \left[ 0\ 0\ \left(\frac{\partial^2}{\partial t^2}\theta 1(t)\right) + \left(\frac{\partial^2}{\partial t^2}\theta 4(t)\right) \right]$$

for bodies 9 and 10 as indicated. Rotation transformation matrices $Tr9$ and $Tr10$ are the expanded version of equations (6.2) and (6.3) respectively, and $V\_ang9$, $V\_ang10$, $W\_ang9$ and $W\_ang10$ are the angular velocity and acceleration of the respective bodies. It is seen that they are all as expected.

For a flexible arm on the path, DynaFlex searches for the beam the arm resides on, extracts the entries of the sub-table to index 4 of the beam's edge table, applies them

to construct the complete second-order elastic rotation matrix discussed in Chapter 4 and computes from it all the kinematic terminal equations needed in the present module for later use. Adopting commonly-used symbols, DynaFlex labels the axial generalized elastic coordinate as $u$, bending coordinates as $v$ and $w$, and generalized angle of twist as $\phi$. When the user specifies that the Taylor monomials be used to discretize these variables, DynaFlex automatically construct them in following general form:

$$u = \sum_{i=1}^{a} x^i u_i(t) \tag{6.4}$$

$$v = \sum_{i=1}^{b} x^{i+1} v_i(t) \tag{6.5}$$

$$w = \sum_{i=1}^{c} x^{i+1} w_i(t) \tag{6.6}$$

$$\phi = \sum_{i=1}^{d} x^i \phi_i(t) \tag{6.7}$$

where $0 \leq x \leq L$, with $L$ being the length of the beam, and $a$, $b$, $c$ and $d$ are the number of monomials specified too by the user for the deformation variables $u$, $v$, $w$ and $\phi$ respectively. This set of discretization equations ensure that the local body frame $X^i Y^i Z^i$ is fixed at the $x = 0$ end of the beam and its $X$ axis is kept tangent to the centroidal axis of the beam, as stipulated in Chapter 4. Note that $a + b + c + d$ gives the number of generalized elastic coordinates in $\mathbf{q}_f$ for the beam.

When four monomials are employed for discretizing the bending deformation in the local $Y$ direction of a beam whose graphical edge is numbered 2, these generalized elastic coordinates will appear to be $v2_1(t)$, $v2_2(t)$, $v2_3(t)$ and $v2_4(t)$ in the Maple implementation.

To use other types of monomials, the user also needs only to specify in the system input file the name of the type and the number of the monomials for each of the deformation

variables. DynaFlex automatically constructs the deformation variables, as shown above. In addition to Taylor monomials, DynaFlex accepts Chebyshev and Legendre monomials.

### 6.1.4 Kinematic Constraint Module

Having found the transformation matrices, DynaFlex is now ready to generate system kinematic constraint equations.

DynaFlex first looks at all chords to single out the cotree joint edges, and determines, based upon the type of the cotree joint, what kinematic quantities are needed to build up the constraint equations. It then uses the *path* lists and the attendant directional information, in combination with the terminal equations of the elements involved, to compute the needed kinematic quantities.

To facilitate the process, every joint in the physical system is assigned a pair of reference frames, called joint definition frames (Haug [44]). The initial orientation between them is set in a certain manner, usually parallel to each other. With these joint definition frames, the constraints each type of joint puts on the bodies they connect can be expressed by specifying the displacement between their origins and the relative orientation of their coordinate axes allowed by the joint.

For example, when a ball joint is found in the cotree DynaFlex will compute the displacement of each of its joint definition points using the displacement terminal equations of all the elements that make up the path from the datum to the joint's two nodes (tree top nodes), and set their difference to zero. The operations are more involved and complicated for other types of joints (Haug [44]), but basically DynaFlex proceeds in a similar way.

DynaFlex treats cotree kinematic drivers in the same manner.

When the position vector of the origin of a joint definition frame is needed and there is a flexible arm element on the path leading to the frame, DynaFlex looks into the edge

table of the beam on which the flexible arm resides, constructs, and evaluates accordingly the second-order deformation displacement field for the arm. It then produces the arm's displacement terminal equations expressed in the global frame for use in building up the position vector.

For the slider crank example, DynaFlex produces the following loop closure equations from the circuit equation for cotree joint $j_3$:

$$
\begin{aligned}
&CJ\_eq := \\
&\quad [-\cos(\theta 1(t))r2x + \cos(\theta 1(t))r3x - \mathrm{s}7(t) \\
&\quad + (-\cos(\theta 1(t))\cos(\theta 4(t)) + \sin(\theta 1(t))\sin(\theta 4(t)))r5x \\
&\quad - (-\cos(\theta 1(t))\cos(\theta 4(t)) + \sin(\theta 1(t))\sin(\theta 4(t)))r6x] \\
&\quad [-\sin(\theta 1(t))r2x + \sin(\theta 1(t))r3x \\
&\quad - (\sin(\theta 1(t))\cos(\theta 4(t)) + \cos(\theta 1(t))\sin(\theta 4(t)))r5x \\
&\quad + (\sin(\theta 1(t))\cos(\theta 4(t)) + \cos(\theta 1(t))\sin(\theta 4(t)))r6x]
\end{aligned}
$$

At present, DynaFlex is capable of handling constraint equations directly for revolute (pin), universal, spherical, weld and revolute-cylindrical joint. A sliding joint is modelled with a sliding arm together with a weld joint. A cylindrical joint is modelled with a sliding arm and a revolute joint.

### 6.1.5 Dynamic Module

While kinematic equations are formulated based on the circuit matrix, kinetic equations are formulated based on the cutset matrix. In this module, DynaFlex first singles out all the edges that represent the tree border joints, writes a virtual work expression for each of the cutsets defined by the joints, and sums them to zero to give a single system VW equation.

To find the tree border joints, DynaFlex searches up each of the *path* lists set up in

the topological analysis. The first joint edge it finds is the sought tree border joint edge. (For a free floater system, the user is expected to introduce a free joint element between the system and the inertial frame.) To write the virtual work cutset equation for the joint edge, DynaFlex then extracts the row of the cutset matrix that corresponds to the edge and substitutes in the cutset the virtual work terminal equation for each edge, which has been defined in Chapter 3.4. The substitution process is achieved by calling Maple procedures that evaluate the virtual work of various elements, in terms of the virtual branch coordinates, with the help of the kinematic quantities found previously. Other Maple procedures compute virtual rotations based on angular velocities, the displacement, velocity, acceleration, and virtual displacement of points of interest in the system.

Finally DynaFlex sums up virtual work equations for all tree border joints and sets it to zero. The system virtual work equation is now expressed in varied branch coordinates $\delta\mathbf{q}$, as the Approach 4.2 in Section 5.3 suggested.

At this point, DynaFlex tests the global variable *Iedge* that has been set in the input file and see whether it is assigned a NULL list of edges. If it is, which means the system under consideration is an open loop one, DynaFlex will simply collect its system virtual work equation with respect to the varied branch coordinates $\delta\mathbf{q}$, and set their coefficients to zero to produce the system kinetic equations. In such a case, $\mathbf{q}$ are system independent coordinates.

If *Iedge* is not assigned a NULL list, which means the system has at least one closed kinematic loop, DynaFlex constructs, according to a set of name-creating rules, the coordinates corresponding to the edges specified in the list of *Iedge*, takes them to be the system independent coordinates $\mathbf{q_i}$. It then takes up the loop closure equations at velocity level and partitions them into the following form:

$$\mathbf{C}_d\dot{\mathbf{q}}_d + \mathbf{C}_i\dot{\mathbf{q}}_i = -\mathbf{C}_t \tag{6.8}$$

where $\mathbf{C}_d$, $\mathbf{C}_i$, and $\mathbf{C}_t$ are the partial derivative of $\mathbf{C}$ with respect to $\mathbf{q}_d$ (the system dependent coordinates), $\mathbf{q}_i$, and $t$. This set of equations are then solved to give

$$\dot{\mathbf{q}}_d = -\mathbf{C}_d^{-1}(\mathbf{C}_i\dot{\mathbf{q}}_i + \mathbf{C}_t) \tag{6.9}$$

These equations are finally turned into a set of corresponding variational equations. This is done first by discarding the $\mathbf{C}_t$ terms that are not functions of branch coordinates and replacing the velocities with their corresponding variation to give

$$\delta\mathbf{q}_d = \mathbf{J}\delta\mathbf{q}_i \tag{6.10}$$

where $\mathbf{J} = -\mathbf{C}_d^{-1}\mathbf{C}_i$. Note that $\mathbf{J}$ is non-singular as long as there are no redundant constraints.

For the slider-crank, one has

$$\begin{aligned}
&\delta s7(t) = \sin(\theta 4(t)) \\
&\left(\sin(\theta 1(t))^2 r2x - \sin(\theta 1(t))^2 r3x - \cos(\theta 1(t))^2 r3x + \cos(\theta 1(t))^2 r2x\right)\delta\theta 1(t) \Big/ ( \\
&-\cos(\theta 1(t))\cos(\theta 4(t)) + \sin(\theta 1(t))\sin(\theta 4(t))) \\
&\delta\theta 4(t) = (-\cos(\theta 1(t))r3x \\
&+ r5x\cos(\theta 1(t))\cos(\theta 4(t)) - r5x\sin(\theta 1(t))\sin(\theta 4(t)) + \cos(\theta 1(t))r2x \\
&- r6x\cos(\theta 1(t))\cos(\theta 4(t)) + r6x\sin(\theta 1(t))\sin(\theta 4(t)))\delta\theta 1(t) \Big/ ( \\
&r6x\cos(\theta 1(t))\cos(\theta 4(t)) - r6x\sin(\theta 1(t))\sin(\theta 4(t)) \\
&- r5x\cos(\theta 1(t))\cos(\theta 4(t)) + r5x\sin(\theta 1(t))\sin(\theta 4(t)))
\end{aligned}$$

the second of which is the expanded form of equation (5.34).

These equations are then substituted back into the system virtual work equation, which is also collected afterwards, but now with respect to the varied system independent coordinates $\delta\mathbf{q}_i$. Setting their coefficients to zero produces the kinetic equations for the

closed loop system.

Note, when there are beams involved in the kinematic loop, equation (6.10) contains terms of very high order in the generalized elastic coordinates, and they are truncated to whatever order the user prefers, usually to second order only. The truncation idea also is applied to the system VW after equation (6.10) has been substituted, usually to second order as well.

The symbolic inversion operation on equation (6.8) to obtain equation (6.9) is feasible here since the employment of a joint tree combined with an intelligent choice of cotree joints (see the Rigid Spatial Slider-Crank example in the next chapter) reduces the number of constraint equations that need to be inverted to a minimum; also, in practice, systems that have many closed loops are rare. In other words, $C_d$ is usually a small matrix.

To bring the kinetic equations into their simplest form, DynaFlex can perform some surgical operations on them. It first writes the kinetic equations in the form

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{F}_1(\dot{\mathbf{q}}) + \mathbf{F}_2(\mathbf{q}, t) = 0 \tag{6.11}$$

so that the three terms are separate from each other. It then carries out simplifications on the entries of the mass matrix $\mathbf{M}$, one by one, using commands of both *simplify* and *combine*. It collects the force terms $\mathbf{F}_1(\dot{\mathbf{q}})$ using the command *coeffs* with respect to the velocities $\dot{\mathbf{q}}$ and their products, and simplifies the resulting coefficients with *simplify*. The last term in the kinetic equation is simplified also using the *simplify* and *combine* commands and collected with respect to system branch coordinates $\mathbf{q}$ to further simplify for some cases.

Note that the simplification operations are very costly especially for large problems. In fact, in such cases, more than two-thirds of the Maple *CPU* time is spent on them. Even so, one can never be sure that the results are indeed simplified.

For the slider crank, the system virtual work equation reads

$$VW := \left( - m9\sin(\theta 1(t))^2 r2x^2 \left( \frac{\partial^2}{\partial t^2}\theta 1(t) \right) + \mathrm{T}(t) \right.$$
$$\left. - m9\cos(\theta 1(t))^2 r2x^2 \left( \frac{\partial^2}{\partial t^2}\theta 1(t) \right) - J99 \left( \frac{\partial^2}{\partial t^2}\theta 1(t) \right) \right) \chi$$

where $\chi$ stands for $\delta\theta 1(t)$. From this equation, DynaFlex extracts the system kinetic equation to be

$$dyn\_eq2 := \left[ \left( m9r2x^2 + J99 \right) \left( \frac{\partial^2}{\partial t^2}\theta 1(t) \right) - \mathrm{T}(t) \right]$$

as expected, since the system is equivalent to a single body rotating about joint $j_1$.

To facilitate subsequent numerical solution of system equations, including kinematic and kinetic equations, the Dynamic Module presents them in both the second-order state space form (above) and the DAE form (see the following section). The user is also presented with the mass and the force matrices involved in both of the two forms.

## 6.2 Numerical Solution Strategies

The unique structure of the symbolic system equations generated by DynaFlex is examined. Since these equations are not easy to solve directly, either symbolically or numerically, they are generally converted into forms for which there exist good numerical solution strategies. The flowcharts for the numerical methods that have been used to solve the converted forms of equations are illustrated.

### 6.2.1 Structure of System Equations

As mentioned previously, the joint tree formulation of the VW graph-theoretic approach (as currently implemented) generates system equations that do not have unknown joint reaction forces in them, even for systems with closed loops, due to the advantage taken

of the zero-VW property of ideal joints in the system. Generally, these system equations can be written compactly as

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{f}(\dot{\mathbf{q}}, \mathbf{q}, t) \tag{6.12}$$

$$\mathbf{C}(\mathbf{q}, t) = 0 \tag{6.13}$$

where $\mathbf{M}$ is the mass matrix and has the dimension of $S_d \times T_d$, with $S_d$ being the system degrees of freedom, which is also equal in number to $\mathbf{q}_i$, and $T_d$ the tree degrees of freedom, equal in number to $\mathbf{q} = [\mathbf{q}_d \;\; \mathbf{q}_i]^T$, the branch coordinates; $\mathbf{f}(\dot{\mathbf{q}}, \mathbf{q}, t)$ is the force column matrix with $S_d$ rows that collects the remaining force terms that are not functions of the system accelerations; $\mathbf{C}(\mathbf{q}, t)$ are the $T_d - S_d$ kinematic constraint equations. The first set of second-order ordinary differential equations are system kinetic equations while the second set are the algebraic constraint equations. It is worth noting that the mass matrix $\mathbf{M}$ is non-square, as a consequence of eliminating the joint reactions.

It is clear that there are $T_d$ equations in (6.12) and (6.13) containing $T_d$ unknowns, the branch coordinates $\mathbf{q}$. Thus, they constitute a necessary and sufficient set of system equations.

From the view point of differential equation theory, equations (6.12) and (6.13) are a set of index two differential-algebraic equations (DAEs), one index lower than those usually obtained with joint reactions or Lagrange Multipliers appearing (Brenan et al. [3]). This means that these Lagrange Multiplier Free (LMF) equations are easier to solve than those that contain the multipliers, if DAE methods are applied. Even so, direct numerical solution of index 2 DAEs is still generally not attainable. Thus, to solve equations (6.12) and (6.13), they are converted into two other major forms, namely, the state space representation and the index 1 form, for which numerical techniques are much more mature, although not perfect.

To get the state-space form for equations (6.12) and (6.13), the constraint equations

are differentiated twice to give

$$\mathbf{C_q}\ddot{\mathbf{q}} = \gamma \tag{6.14}$$

These equation are then joined with the second-order ordinary differential kinetic equations to give

$$\mathbf{sysM}\ddot{\mathbf{q}} = \mathbf{sysf}(\dot{\mathbf{q}}, \mathbf{q}, t) \tag{6.15}$$

where $\mathbf{sysM} = [\mathbf{M}\ \ \mathbf{C_q}]^{\mathbf{T}}$ is the system mass matrix and has the dimension of $T_d \times T_d$, $\mathbf{sysf}(\dot{\mathbf{q}}, \mathbf{q}, t) = [\mathbf{F}(\dot{\mathbf{q}}, \mathbf{q}, t)\ \ \gamma]^T$ is the system force column matrix with $T_d$ rows.

These equations are exported from Maple in Fortran so that their first-order ordinary differential equations can be constructed and solved more efficiently.

To get the index 1 form, let

$$\dot{\mathbf{q}} = \mathbf{r} \tag{6.16}$$

Thus, equations (6.12) and (6.13) along with (6.16) become

$$\begin{cases} \mathbf{M}\dot{\mathbf{r}} = \mathbf{f}(\mathbf{r}, \mathbf{q}, t) \\ \dot{\mathbf{q}} = \mathbf{r} \\ \mathbf{C}(\mathbf{q}, t) = 0 \end{cases} \tag{6.17}$$

which are the sought index 1 form DAEs of the system equations (6.12) and (6.13). Note there are now $2 \times T_d$ equations, for the same number of unknowns, $\mathbf{q}$ and $\mathbf{r}$.

The conversion to index 1 forms is done automatically in the Dynamic Module of DynaFlex.

A third approach that can be used to solve the index 2 DAEs is the so called embedding technique (Shabana [6]). Instead of joining the constraint equations to the first ODEs as in the state space approach, this technique eliminates the dependent variables at the velocity and acceleration level in the kinetic equations of (6.12) via the constraint equations (6.13)

in much the same way as the varied dependent tree variables were eliminated in the system VW equation with equation (6.10). In doing so, the number of unknown velocities and accelerations in the ODEs are reduced to a minimum, i.e., equal to $S_d$. Since dependent variables at the displacement level still appear in the ODEs, however, the integration of the ODEs can not be advanced without using again the constraint equation at the displacement level to calculate these dependent displacements. Thus, as compared with the state space approach, this technique reduces the number of ODEs that need to be solved and makes the constraint satisfied at the velocity level but not at the displacement level. The displacement constraints are satisfied after the integration of the minimum set of ODEs when the displacements of independent variables are obtained, by solving these constraint equations for the dependent displacements with a numerical approach for non-linear equations, e.g., the Newton-Raphson method.

The purely numerical implementation of this approach was found by Nikravesh [113] to be not as efficient as the state space approach in terms of computer $CPU$ time. It is also much more complicated to implement than the other two approaches mentioned above. In view of this, the embedding technique is not implemented as a built-in numerical solution option in DynaFlex. With the current symbolic generation of system equations, however, the user can implement it with relative ease.

### 6.2.2 Implemented Numerical Methods

All numerical methods have been implemented in Fortran–77 and built largely on the scientific numerical package NAG.

#### ODE Methods

The numerical methods that have been implemented for the solution of the state space form of ODEs include the Runge-Kutta-Merson with the NAG routine *D02BAF*, Runge-

Kutta-Adams with the NAG routine *D02CAF*, a backward differentiation formula (BDF) with the NAG routine *D02NBF*, and the HHT-$\alpha$ [94] method. This last was implemented using the NAG linear algebra routine *F04ATF* [85], which calculates the accurate solution of a set of real linear equations with a single right-hand side, using an $LU$ factorization with partial pivoting and iterative refinement.

For non-stiff systems of equations, the first two methods can be used, between which the *D02BAF* is preferred for its fast and fairly accurate performance. For stiff equations, the last two should be used.

Figure 6.5 shows the steps involved in using the NAG ODE solver routines. Figure 6.6 outlines the steps involved in using the HHT-$\alpha$ method. All the expressions that are needed in the Fortran code are exported from Maple in optimized Fortran code.

When the NAG routines are used, the system ODEs given by equations (6.15) are first converted into a total of $2 \times T_d$ first-order differential equations, and solved afterwards while the $\alpha$ method discretizes the system equations (6.15) directly.

In the $\alpha$ method code, the Jacobian for the Newton iterations is also computed symbolically in Maple. It is not updated for each iteration in a stepping. It is updated only when convergence has failed after several trials with the same Jacobian. The number of trials before the Jacobian is updated can be adjusted by the user as well. The criterion for ending the iteration is based on the relative tolerance (RTOL) and absolute tolerance (ATOL) chosen by the user, and has the following form

$$RTOL * q(i) + ATOL \tag{6.18}$$

Once all the increments in q are smaller than their tolerance specified by this expression, the iteration is stopped and convergence is regarded as achieved. Otherwise, iteration continues until its number reaches 2000. Then the program stops, giving the error message

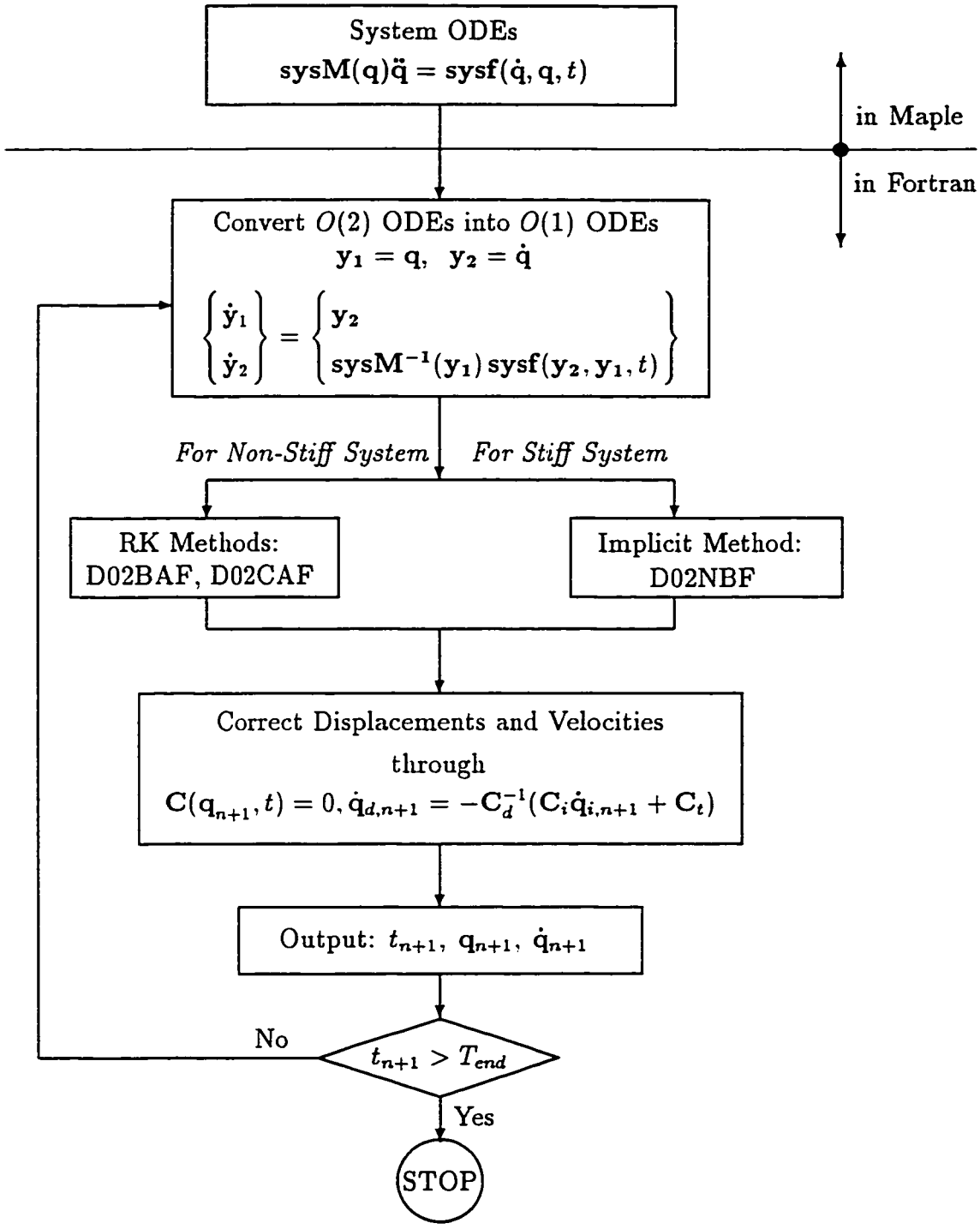



Figure 6.5: Flowchart of Using NAG Routines for ODEs

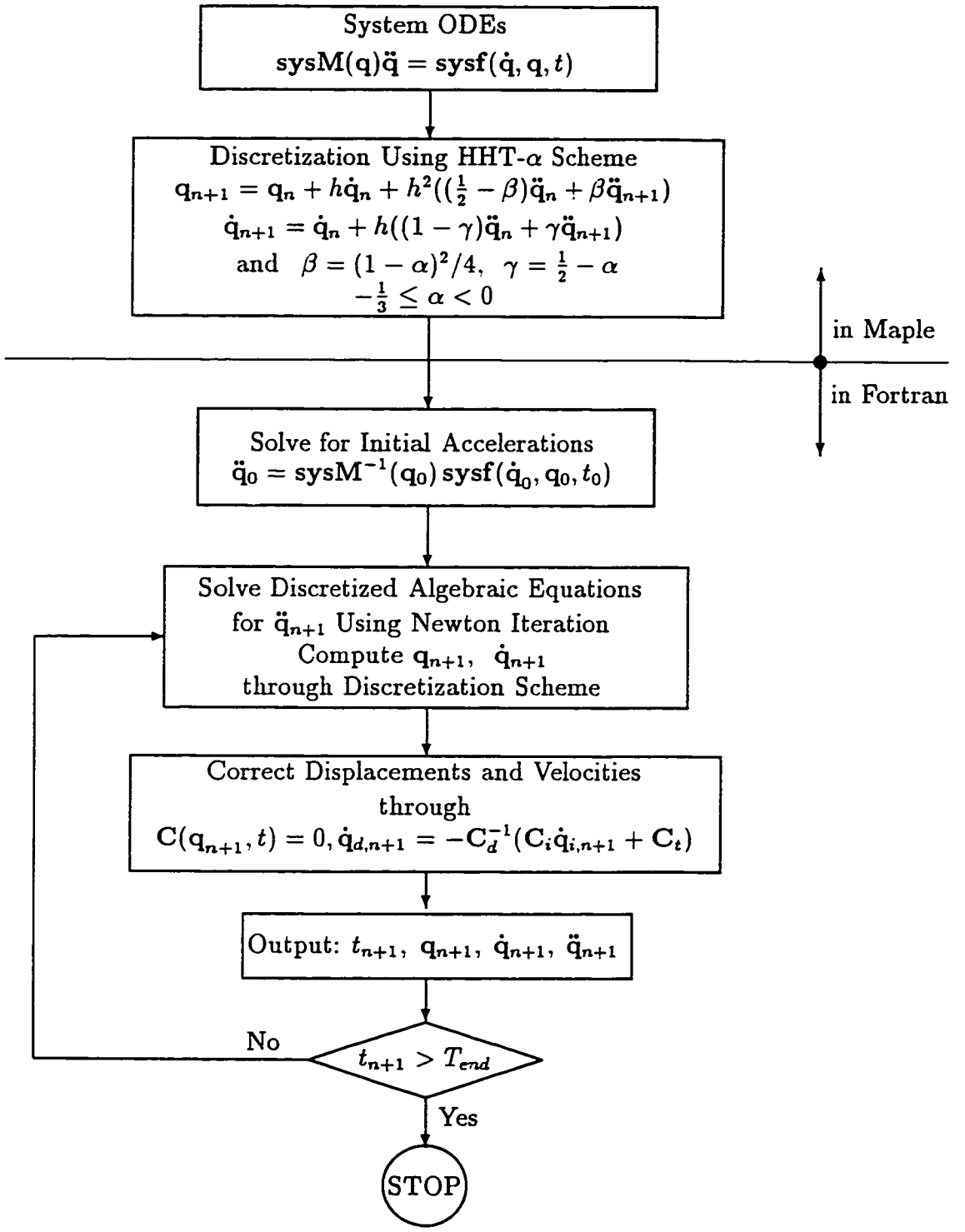



Figure 6.6: Flowchart of Using HHT-$\alpha$ Method for ODEs

'too many iterations'.

With the state space approach, regardless of which integration method is used, the solved velocities and displacements may not satisfy the constraint equations imposed on them. Thus, measures must be taken to address this. The popular way to achieve this is to hold results for some of the coordinates and correct the rest through the constraint equations. The coordinates whose results are held as accurate are termed independent coordinates, and others dependent ones. In the current implementation, the same independent coordinates that have been chosen in the Dynamic Module of DynaFlex are used here so that the symbolic generation and the numerical solver coordinate well with each other.

The correction on velocities is simple. It evaluates the dependent velocities in terms of the independent ones that have been obtained through the integration, using the Fortran output from Maple of equation (6.9). The correction on displacements proceeds more or less the same. The only difference is that now there is the need to solve a set of non-linear algebraic equations (6.13) to find the dependent coordinates given the independent and the approximate dependent ones. This, however, is basically a repetition of the Newton iteration process used for the $\alpha$ method.

### DAE Methods

The numerical methods that have been implemented for the solution of the index 1 form of system DAEs include the BDF Euler method and the Trapezoidal rule. The use of DASSL is easily accommodated by exporting from DynaFlex the residual of the Index 1 equations when it is used with the simplest set of control parameters (Brenan, Campbell and Petzold [3]).

Figure 6.7 shows the steps of the BDF Euler method for solving index 1 system DAEs. The Newton iteration is coded exactly the same as with the above-mentioned $\alpha$ method.

The Jacobian matrix is obtained symbolically in Maple.

It is seen that the code is considerably simpler than those used for the state space form since the numerical solutions for the displacements now satisfy the kinematic constraint equations automatically and thus there is no need to correct them. In addition, unlike other methods discussed here, the Euler method requires only the initial displacements and velocities to start. It does not need accelerations to start or advance the integration.

Although solutions for velocities and accelerations may not satisfy the constraint equations exactly for the index 1 form, their corrections are easy and straightforward as mentioned above; in practice they are not corrected at all since they stray very little due to the satisfaction of constraints at the displacement level.

The steps involved in the trapezoidal rule are shown in Figure 6.8. They are very much similar to those of the BDF Euler because they all belong to the one-step BDF family. The only difference is that the trapezoidal rule needs initial accelerations, in addition to initial velocities and displacements, to start, as do the NAG routine solvers.

## 6.3 Concluding Remarks

The joint tree formulation of the proposed VW graph-theoretic approach has been implemented with Maple, a symbolic computation package. The resulting Maple code, called DynaFlex, takes the specification of the individual edges of a system graph as input, along with other control parameters selected by the user, and automatically generates the system equations, including kinetic and kinematic constraint equations. These equations are then manipulated symbolically and exported for numerical solution.

Numerical solutions of system equations are implemented with different methods in Fortran, giving the user a variety of choices. These Fortran codes are suitable for use with either the state space or the index 1 form of the systems' index 2 Lagrange Multiplier

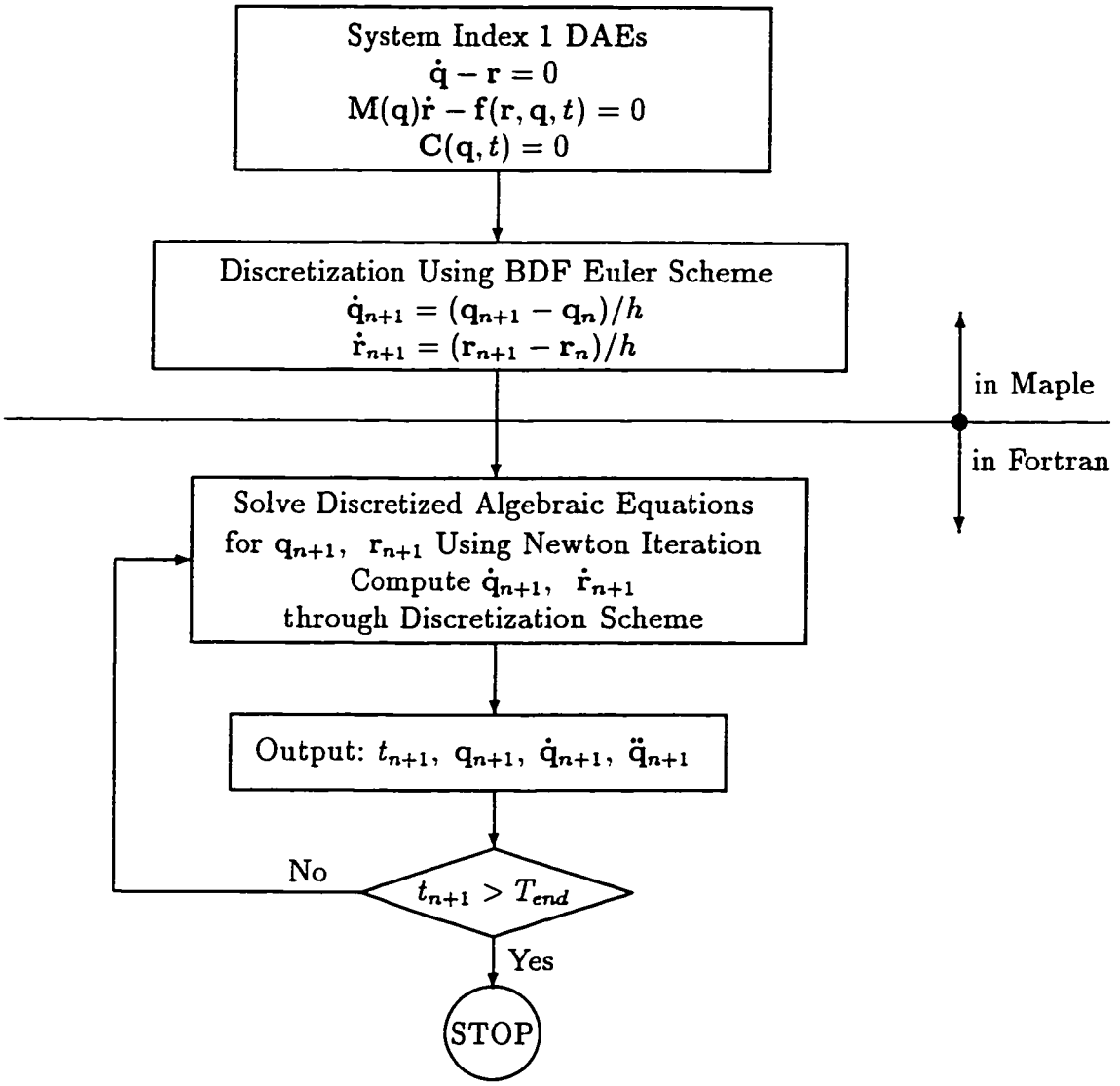



Figure 6.7: Flowchart of Using BDF Euler for Index 1 DAEs

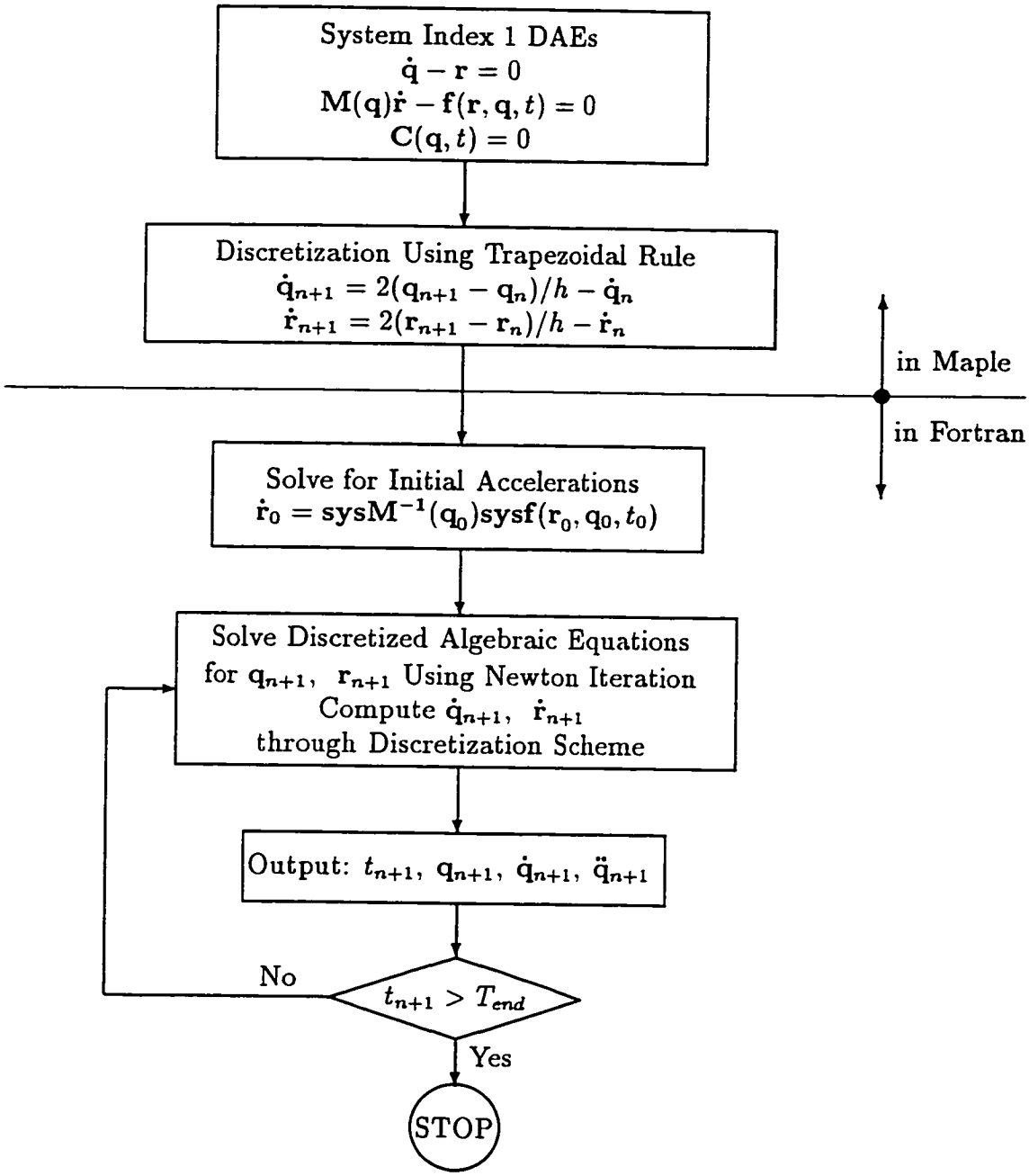



Figure 6.8: Flowchart of Using Trapezoidal Rule for Index 1 DAEs

Free equations.

Experience with Maple so far shows that it is powerful and fairly straightforward to use, although there are some tricks and anomalies. For example, for large objects, the *multiply* command of the linear algebra package does not work as fast as when *do* loops are used instead to compute the product. Furthermore, when the objects are very large, this *multiply* command fails to work while the *do* loops work perfectly. Symbolic integration of polynomials is executed tens or even hundreds of times faster, depending on the size of the integrand, when the integrand is collected with respect to the integration variable and its coefficients substituted with some temporary variables first. This speed-up is especially prominent for systems containing beams. Collection of a polynomial expression is also at least tens of times faster than that when the monomials are replaced with a sequence of variables like $x1$, $x2$, $x3$ and so on, and collected afterwards.

# Chapter 7

# Example Problems

In this chapter a number of problems solved by DynaFlex are presented to further validate the proposed VW graph-theoretic approach, as well as the implemented computer code. These examples include both planar and spatial systems, some containing flexible beams. The DynaFlex results, either in symbolic form or in numerical form, are all compared with results from the literature or with other codes.

## 7.1 Spinning Disk

This spinning disk, see Figure 7.1, was taken from Ginsberg [115]. It consists of two massless shafts and a uniform disk connected in series with pin joints to the ground. The ground joint at $D$ and the joint between the disk and the shaft at $B$ are driven by internal joint torques so that these joint coordinates have constant velocities of $\dot{\psi}$ and $\dot{\phi}$, respectively. One wishes to find the required joint torques and the differential equation governing the nutation angle $\theta$.

Various reference frames are first set up, among which the joint definition frames are arranged according to the conventions employed in formulating their constraint equations (see Table 6.2). The $Z^1$ axis of the top shaft's body frame is arranged along the longitu-

dinal axis of the shaft and the $X^1$ along the the joint axis at point $A$. This frame is also taken to be the joint definition frame on the shaft for the ground pin joint. For clarity, the joint definition frames at point $A$ are shown in Figure 7.1, where axes $Z_A^1$ and $Z_A^2$ are coincident with each other and with the pin joint axis at point $A$ as well, and when $\theta = 0$ the two frames coincide. The $Y^2$ axis of the intermediate shaft's body frame is arranged parallel to the axis of the pin joint at $A$ too. Joint definition frames at $B$ are parallel to those of the two bodies it connects.

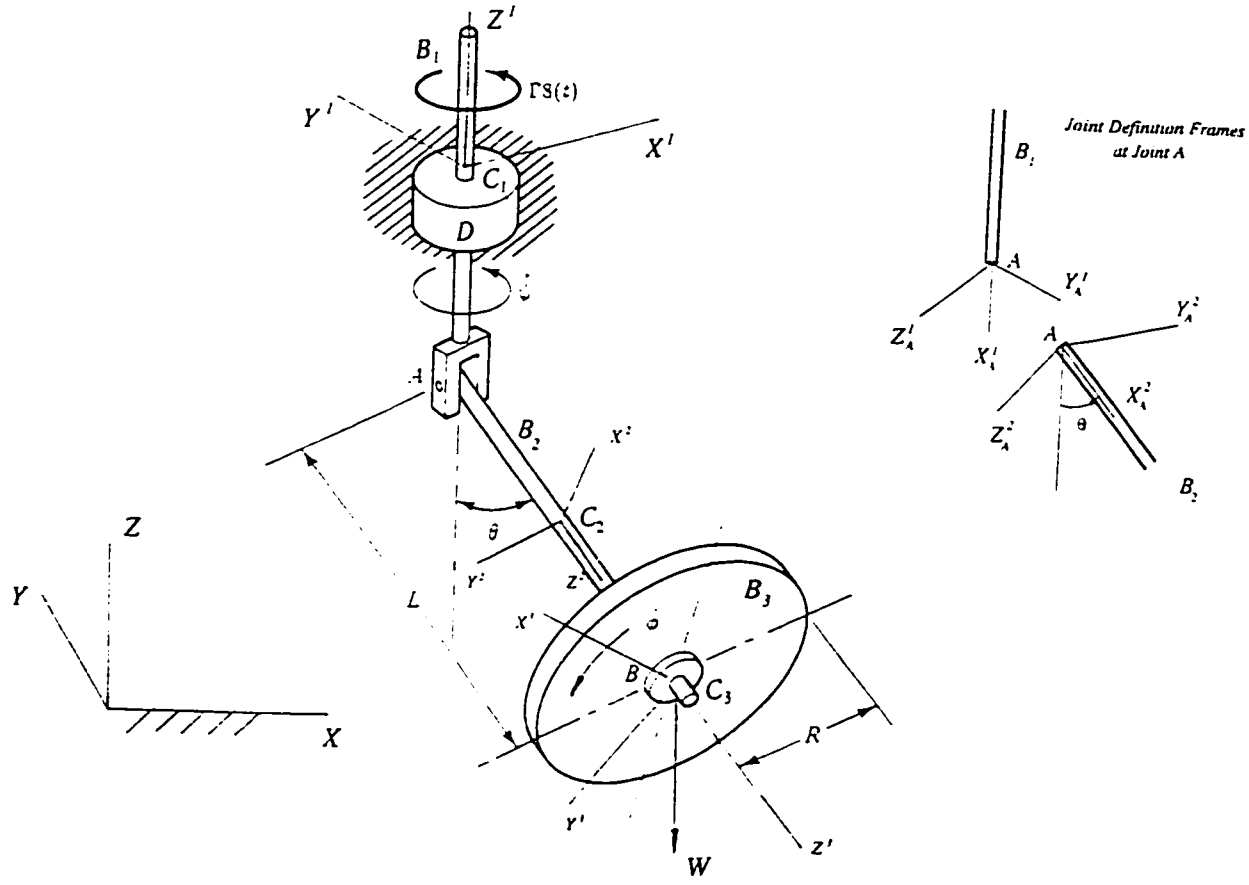



Figure 7.1: The Spinning Disk (Ginsberg [115])

The system graph is then constructed based on the elements involved in the problem,

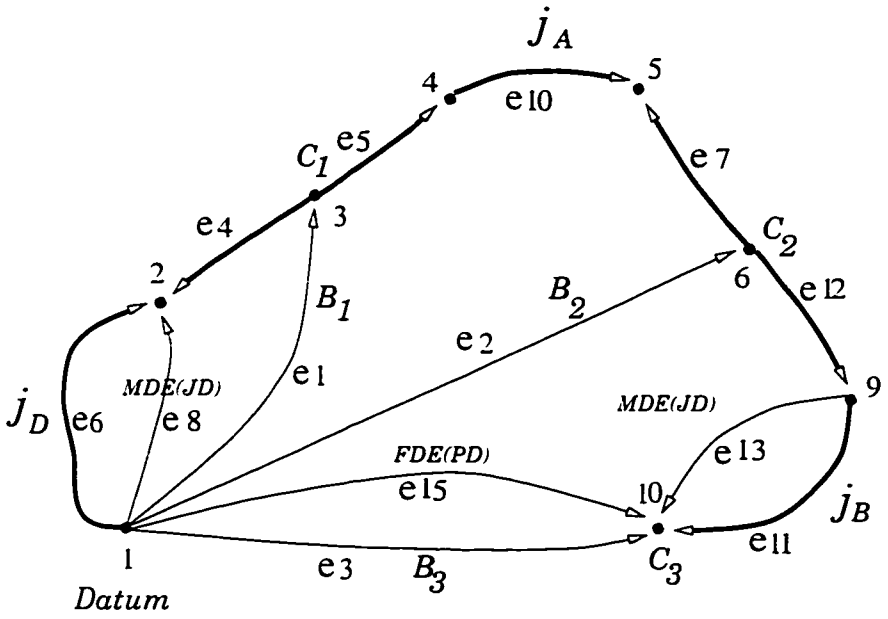



Figure 7.2: Graph of the Spinning Disk

as shown in Fig. 7.2, and the bold edges form the joint formulation tree. The three bodies are represented by edges $e1$, $e2$ and $e3$. Joints at $D$, $A$ and $B$ are represented by edges $e6$, $e10$ and $e11$ respectively, and their locations on the bodies by edges $e4$, $e5$, $e7$ and $e12$. The motion driver at the ground joint $D$ is modelled with edge $e8$, a motion driver element, the motion driver at joint $B$ by edge $e13$, and the weight of the disk by edge $e15$, a force driver that is applied from the inertial frame (termed a position type). From the choice of the formulation tree, it is expected that the system equations will be in terms of the branch coordinates $\mathbf{q} = [\psi \ \theta \ \phi]$.

The Maple input file that describes each of the edges is included in Appendix A.3, which also contains part of the output. DynaFlex is thus run, after which the following data file for the problem is read in:

```
> r7z:=-L/2; r12z:=L/2; J31:=(m3*R^2)/4; J35:=J31; J39:=(m3*R^2)/2;
> theta.6(t):=psi(t); theta.10(t):=theta(t); theta.11(t):=phi(t);
```

$$r7z := -\frac{1}{2}L$$

$$r12z := \frac{1}{2}L$$

$$J31 := \frac{1}{4}m3\,R^2$$

$$J35 := \frac{1}{4}m3\,R^2$$

$$J39 := \frac{1}{2}m3\,R^2$$

$$\theta 6(t) := \psi(t)$$

$$\theta 10(t) := \theta(t)$$

$$\theta 11(t) := \phi(t)$$

resulting in system equations in the form:

$$\begin{aligned}
dyn := \\
&\left[\left(\frac{1}{4}m3\,R^2 + \frac{1}{4}m3\,R^2\cos(\theta(t))^2 - m3\,L^2\cos(\theta(t))^2 + m3\,L^2\right)\right. \\
&\left(\frac{\partial^2}{\partial t^2}\psi(t)\right) - \frac{1}{2}m3\,R^2\cos(\theta(t))\left(\frac{\partial^2}{\partial t^2}\phi(t)\right) \\
&+ \frac{1}{2}m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\phi(t)\right)\left(\frac{\partial}{\partial t}\theta(t)\right) + \\
&\left(-\frac{1}{2}\sin(\theta(t))m3\,R^2\cos(\theta(t)) + 2m3\cos(\theta(t))L^2\sin(\theta(t))\right) \\
&\left.\left(\frac{\partial}{\partial t}\psi(t)\right)\left(\frac{\partial}{\partial t}\theta(t)\right) - \Gamma 8(t)\right] \\
&\left[\left(\frac{1}{4}m3\,R^2 + m3\,L^2\right)\left(\frac{\partial^2}{\partial t^2}\theta(t)\right) + \right. \\
&\left.\left(-m3\cos(\theta(t))L^2\sin(\theta(t)) + \frac{1}{4}\sin(\theta(t))m3\,R^2\cos(\theta(t))\right)\right.
\end{aligned}$$

$$\left(\frac{\partial}{\partial t}\psi(t)\right)^2 - \frac{1}{2}m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\psi(t)\right)\left(\frac{\partial}{\partial t}\phi(t)\right)$$
$$- \mathrm{W}\sin(\theta(t))\left(-\sin(\psi(t))^2 L - \cos(\psi(t))^2 L\right)\Big]$$

$$\Big[-\frac{1}{2}m3\,R^2\cos(\theta(t))\left(\frac{\partial^2}{\partial t^2}\psi(t)\right) + \frac{1}{2}m3\,R^2\left(\frac{\partial^2}{\partial t^2}\phi(t)\right)$$
$$+ \frac{1}{2}m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\theta(t)\right)\left(\frac{\partial}{\partial t}\psi(t)\right) - \Gamma 13(t)\Big]$$

Setting them to zero gives the system equations. As expected, these equations are in terms of the branch coordinates $\psi$, $\theta$, and $\phi$.

Note, in its most general sense, this example is a three-degree-of-freedom open-loop system. DynaFlex thus generates three ordinary equations, as expected, which can be solved for any three unknowns given sufficient information. In fact, DynaFlex always produces the set of system equations in this general sense unless the user specifies some input data, (such as the driving torques in the present case, either in symbolic or numerical form) before or during its execution.

To get the required solution, $\ddot{\psi}(t)$ and $\ddot{\phi}(t)$ in the above equations are set to zero by executing the following Maple command to produce

```
> dyn1:=map(eval,subs(diff(psi(t),t$2)=0,diff(phi(t),t$2)=0, eval(dyn)));
```

$$dyn1 :=$$
$$\Big[\frac{1}{2}m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\phi(t)\right)\left(\frac{\partial}{\partial t}\theta(t)\right) +$$
$$\left(-\frac{1}{2}\sin(\theta(t))m3\,R^2\cos(\theta(t)) + 2m3\cos(\theta(t))L^2\sin(\theta(t))\right)$$
$$\left(\frac{\partial}{\partial t}\psi(t)\right)\left(\frac{\partial}{\partial t}\theta(t)\right) - \Gamma 8(t)\Big]$$

$$\Big[\left(\frac{1}{4}m3\,R^2 + m3\,L^2\right)\left(\frac{\partial^2}{\partial t^2}\theta(t)\right) +$$
$$\left(-m3\cos(\theta(t))L^2\sin(\theta(t)) + \frac{1}{4}\sin(\theta(t))m3\,R^2\cos(\theta(t))\right)$$

$$\left(\frac{\partial}{\partial t}\psi(t)\right)^2 - \frac{1}{2}m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\psi(t)\right)\left(\frac{\partial}{\partial t}\phi(t)\right)$$
$$- \mathrm{W}\sin(\theta(t))\left(-\sin(\psi(t))^2 L - \cos(\psi(t))^2 L\right)\Big]$$

$$\left[\frac{1}{2}m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\theta(t)\right)\left(\frac{\partial}{\partial t}\psi(t)\right) - \Gamma 13(t)\right]$$

This set of equations agrees exactly with those given by Ginsberg [115]. They now contain $\Gamma 8(t)$, the motion driver torque at joint $D$, $\Gamma 13(t)$, the motion driver torque at joint $B$, and $\theta(t)$, the nutation angle. Given constant $\dot{\psi}$ and $\dot{\phi}$, the second equation can be solved first for $\theta(t)$, which is equivalent to solving a forward dynamics problem. The $\theta(t)$ can be substituted afterwards into the other two equations to give solutions for $\Gamma 8(t)$ and $\Gamma 13(t)$.

It is interesting to point out that although the moment of inertia and the mass for the two shafts are set to zero in the Maple input file, this does not cause any singularity difficulty, as it usually does with a purely numerical approach. This certainly is one of the advantages of a symbolic approach over a numerical one.

## 7.2 Rigid Spatial Slider-Crank

This example is taken from Haug [111] and is shown in Figure 7.3. The driving torque on the crank is such that the crank rotates with a constant speed of $\omega_o = 2\pi$ *rad/s*. This is a closed loop system and requires an inverse dynamic analysis to obtain the driving torque.

The body frame $X^3Y^3Z^3$ of the slider is set up parallel to the global inertial frame $OXYZ$. The link's body reference frame is parallel to $X^3Y^3Z^3$ if link is in its horizontal position parallel to the global $Z$ axis. The crank's body frame is at its center of mass, with $X^1$ pointing longitudinally and $Z^1$ perpendicular to the plane of its motion. The

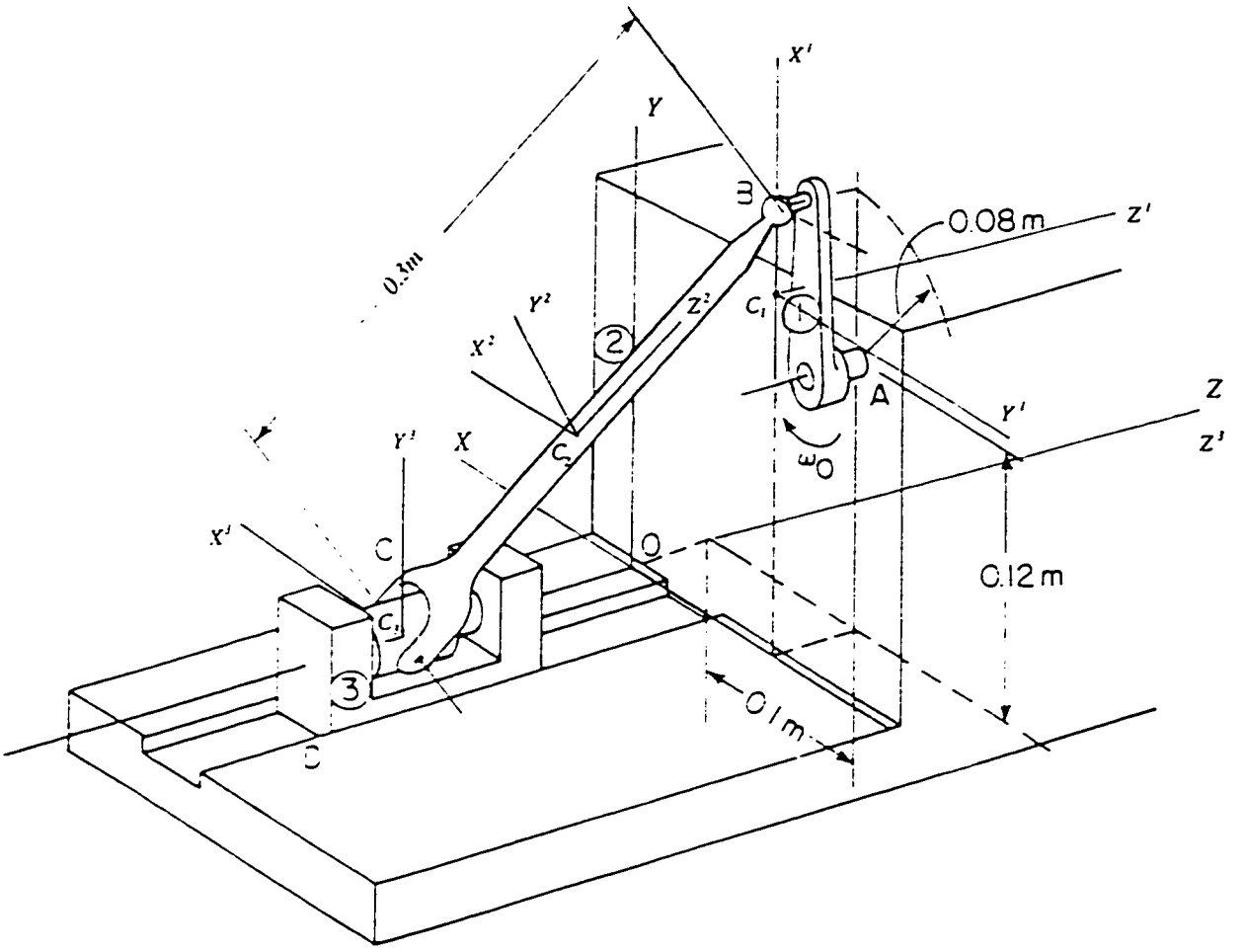



Figure 7.3: The Rigid Spatial Slider-Crank (Haug [111])

joint definition frames for the joint at $C$ are so established that they initially coincide with Frame $X^3Y^3Z^3$, i.e., the joint definition frame on the link is always parallel to the link's body frame while the slider's body frame also serves as the joint definition on the slider. Joint definition frames at point $A$ are arranged parallel to the crank's body frame at the moment shown.

The system graph is again constructed by assembling component graphs, as shown in Figure 7.4. Node 5 is the Datum node, representing the origin of the inertial frame, whereas edge $e16$ represents the location of the joint $A$ relative to the origin. While

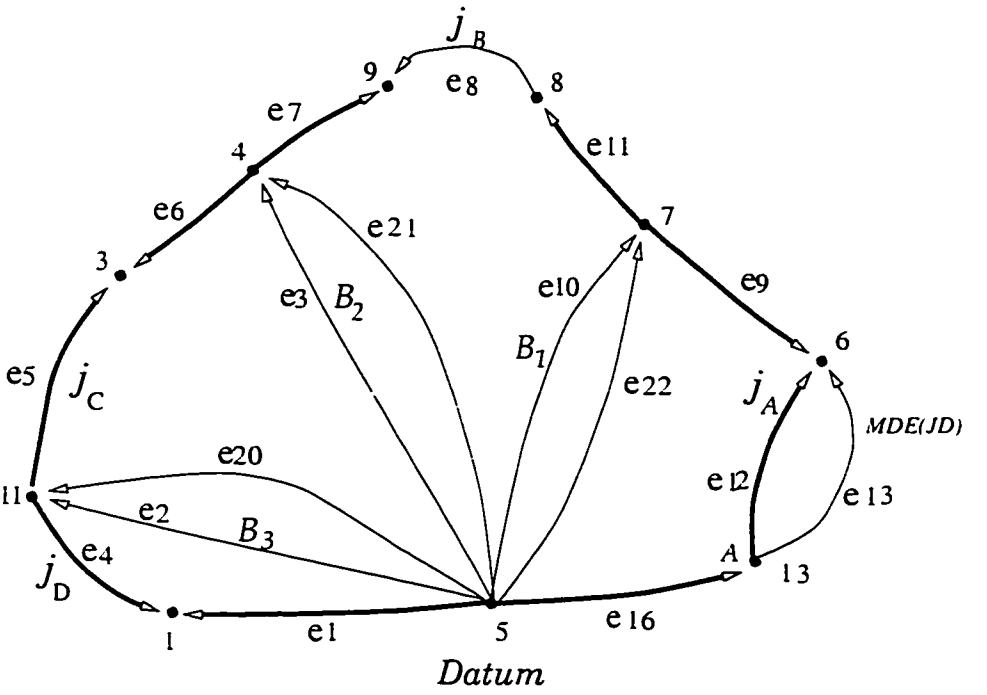



Figure 7.4: Graph of the Rigid Spatial Slider-Crank

edge $e12$ represents the pin joint at $A$, the motion driver at the same joint is represented separately by a motion driver edge $e13$ parallel to it. The crank is modelled with edge $e10$, the weight of the crank by edge $e22$, a force driver edge, and edges $e11$ and $e9$ are body arms used to locate the origins of the two joints on the crank. The link and the slider are represented in a similar way except that, since the joints on the slider can all be regarded as being located at its center of mass and their definition frames are parallel to $X^3Y^3Z^3$, the body arms required to locate these joint locations degenerate to zero length, which can be and are omitted in this graph so that a simpler graph results. Edge $e8$ represents the spherical joint at $B$. Edge $e5$ models the joint at C as a cylindrical-revolute joint between the link and the slider, with the attendant translational coordinate set to zero. Edge $e4$, a weld joint edge, together with edge $e1$ of a sliding body arm, models the sliding joint between the slider and the inertial frame.

To use DynaFlex, a joint tree is formed with the bold edges and the spherical joint is selected into the cotree. This means that a cut to open the kinematic chain is made at the spherical joint, and that there will be three joint constraint equations in the system DAEs. Any other choice of a cotree joint will result in more constraint equations (Haug [111]).

From the formulation tree, one expects to have four branch coordinates ($s1(t)$ for the sliding arm, $\alpha 5(t)$ and $\beta 5(t)$ for the cylindrical-revolute joint, and $\theta 12(t)$ for the pin joint) appearing in system DAEs. With three system constraint equations, the system's degree of freedom is computed to be $4 - 3 = 1$, as expected.

A Maple input file that specifies the edges and nodes in the graph is created, as shown in Appendix A.4. In this file, edge $e12$ (corresponding to pin joint at $A$) is selected as the system independent edge, i.e., all other virtual displacements is expressed in terms of the virtual coordinate of this joint to extract the system kinetic equation.

With this input file, DynaFlex is run. After these system parameters

$$
\begin{aligned}
parameter := \Big[ & r16x = \frac{-1}{10}, r16y = \frac{3}{25}, r9x = \frac{-1}{25}, r11x = \frac{1}{25}, r6z = \frac{-3}{25}, \\
& r7z = \frac{3}{25}, r5y = 0, m10 = \frac{3}{25}, J101 = \frac{1}{100000}, J105 = \frac{1}{10000}, \\
& J109 = \frac{1}{10000}, m3 = \frac{1}{2}, J31 = \frac{1}{250}, J35 = \frac{1}{250}, J39 = \frac{1}{2500}, m2 = 2, \\
& J21 = \frac{1}{10000}, J25 = \frac{1}{10000}, J29 = \frac{1}{10000}, grav = \frac{981}{100}, \\
& \theta 12(t) = 2\pi t \Big]
\end{aligned}
$$

are substituted, the system DAEs become

$$
\begin{aligned}
& dyn1 := \\
& \Bigg[ \frac{25}{6} \frac{\left( \frac{27}{625}\%2 - \frac{27}{625}\%1 \right) \sin(\beta 5(t)) \left( \frac{\partial^2}{\partial t^2} s1(t) \right)}{\cos(\beta 5(t))} + \frac{25}{6} \left( - \frac{14}{15625} \sin(2\pi t) \sin(\alpha 5(t)) \right.
\end{aligned}
$$

$$+\frac{27}{31250}\sin(2\pi t)\sin(\alpha 5(t))\cos(\beta 5(t))^2-\frac{14}{15625}\cos(\alpha 5(t))\cos(2\pi t)$$
$$+\frac{27}{31250}\cos(\alpha 5(t))\cos(2\pi t)\cos(\beta 5(t))^2\Big)\left(\frac{\partial^2}{\partial t^2}\alpha 5(t)\right)/(\sin(\beta 5(t)))+\frac{25}{6}\Big($$
$$-\frac{18}{15625}\sin(\alpha 5(t))\cos(2\pi t)\cos(\beta 5(t))^2+\frac{4}{15625}\%2-\frac{4}{15625}\%1$$
$$+\frac{18}{15625}\cos(\alpha 5(t))\sin(2\pi t)\cos(\beta 5(t))^2\Big)\left(\frac{\partial^2}{\partial t^2}\beta 5(t)\right)/(\cos(\beta 5(t)))$$
$$+\frac{73}{250000}\left(\frac{\partial^2}{\partial t^2}(2\pi t)\right)+\frac{25}{3}$$
$$\left(-\frac{27}{31250}\sin(\beta 5(t))\sin(2\pi t)\sin(\alpha 5(t))-\frac{27}{31250}\sin(\beta 5(t))\cos(\alpha 5(t))\cos(2\pi t)\right)$$
$$\cos(\beta 5(t))\left(\frac{\partial}{\partial t}\alpha 5(t)\right)\left(\frac{\partial}{\partial t}\beta 5(t)\right)/(\sin(\beta 5(t)))$$
$$-\frac{3}{50}\left(\frac{2}{25}\%1-\frac{2}{25}\%2\right)\sin(\beta 5(t))\left(\frac{\partial}{\partial t}\beta 5(t)\right)^2-\frac{25}{6}$$
$$\left(-\frac{27}{31250}\sin(\alpha 5(t))\cos(2\pi t)\sin(\beta 5(t))+\frac{27}{31250}\cos(\alpha 5(t))\sin(2\pi t)\sin(\beta 5(t))\right)$$
$$\left(\frac{\partial}{\partial t}\alpha 5(t)\right)^2-\Gamma 13(t)-\frac{2943}{62500}\sin(2\pi t)$$
$$-\frac{2943}{62500}\frac{\sin(\alpha 5(t))\sin(\beta 5(t))(\cos(\alpha 5(t))\cos(2\pi t)+\sin(2\pi t)\sin(\alpha 5(t)))}{\frac{6}{25}\cos(\alpha 5(t))^2\sin(\beta 5(t))+\frac{6}{25}\sin(\alpha 5(t))^2\sin(\beta 5(t))}$$
$$+\frac{2943}{62500}\frac{\cos(\alpha 5(t))(\%2-\%1)}{\frac{6}{25}\cos(\alpha 5(t))^2+\frac{6}{25}\sin(\alpha 5(t))^2}\Bigg]$$

$$\left[-\frac{6}{25}\sin(\alpha 5(t))\sin(\beta 5(t))-\frac{2}{25}\sin(2\pi t)-\frac{1}{10}\right]$$

$$\left[\frac{6}{25}\cos(\alpha 5(t))\sin(\beta 5(t))+\frac{2}{25}\cos(2\pi t)+\frac{3}{25}\right]$$

$$\left[s1(t)-\frac{6}{25}\cos(\beta 5(t))\right]$$

$$\%1:=\cos(\alpha 5(t))\sin(2\pi t)$$

$$\%2:=\sin(\alpha 5(t))\cos(2\pi t)$$

Due to the relative simplicity of the bottom three joint constraint equations, they can be solved for $s1(t)$, $\alpha 5(t)$ and $\beta 5(t)$ first, which amounts to a kinematic analysis of the system. Then these three are substituted into the first equation to produce the driving torque $\Gamma 13(t)$, which amounts to an inverse dynamic analysis. The torque profile thus obtained is plotted in Figure 7.5 in solid line, in which the same driving torque generated by MOBILE [46] and DADS [44] are presented too, in dotted and dashed line respectively.

It is clear that, with the same set of geometric dimensions and inertia properties of the bodies involved, the DynaFlex results agree completely with that of MOBILE (they superimpose each other), but is different from those of DADS. Particularly, DADS results have an obvious blip at about $t = 0.6$ second, whereas the MOBILE and DynaFlex results have only a barely noticeable bend.

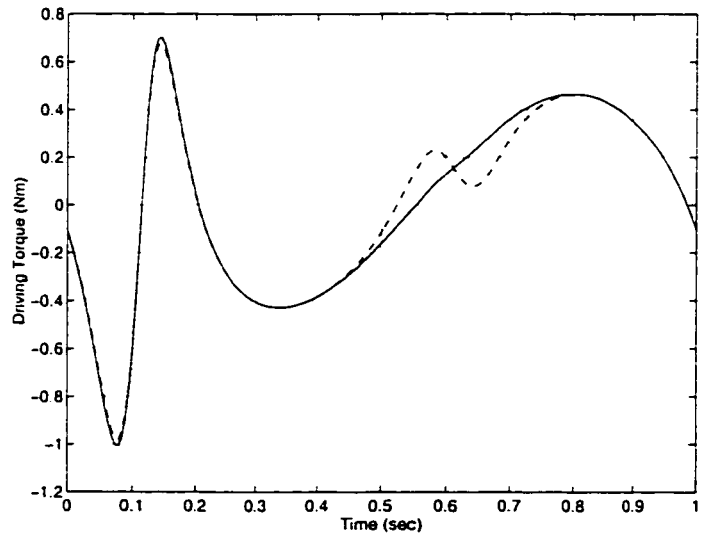



Figure 7.5: The Driving Torques: solid – DynaFlex results; dotted – MOBILE results; dashed – DADS results

Also note the reduction in the number of system DAEs generated by DynaFlex. For this problem, DADS generates fifty-five DAEs consisting of twenty-eight ODEs for the four bodies involved (including the ground) with seven for each of them, since it em-

ploys absolute coordinates and Euler parameters to model rotation, and twenty-seven algebraic equations. Four of the algebraic equations are constraints on the Euler parameters employed and the rest are for the various joints in the system. The conventional joint formulation method [112] for this problem with the same cut joint would result in seven DAEs made up of four ODEs, one for each joint coordinate in the tree, and three algebraic constraint equations for the cut ball joint. In contrast to these conventional approaches, my approach of using virtual work as the through variable combined with the graph-theoretic method reduces the number of system DAEs to merely four, one ODE and three algebraic constraint equations for the ball joint in the cotree.

## 7.3 Planar Spin-Up Beam

The planar rotating flexible beam, discussed previously in Section 5.4.3, has become a benchmark for testing dynamic analysis packages that deal with flexible bodies [40, 11, 88, 57]. This beam, see Figure 7.6, is built into a rigid shaft which is driven with an angular displacement $\theta(t)$ about the $z$ axis. Assuming zero gravity, the beam's tip deformation in its body frame $X^1Y^1Z^1$ is computed using DynaFlex.

The geometric and material properties used for the present flexible beam are: length $l = 10\ m$, cross-sectional area $Area = 0.0004\ m^2$, mass density $\rho = 3000\ kg/m^3$, area moment of inertia $I_z = 2 \times 10^{-7}\ m^4$ and modulus of elasticity $E = 7 \times 10^{10}\ N/m^2$, which are the same as in [40, 11, 88]. The prescribed motion of the joint is also the same and in this form:

$$\theta(t) = \begin{cases} \frac{\omega_s}{T_s}\left[\frac{t^2}{2} + (\frac{T_s}{2\pi})^2(\cos(\frac{2\pi t}{T_s}) - 1)\right] & t < T_s \\ \omega_s(t - \frac{T_s}{2}) & t \geq T_s \end{cases} \tag{7.1}$$

with $\omega_s$ set to 6.0 $rad/s$, and $T_s$ to 15 $s$. This shaft reaches steady motion from rest after $T_s$ $s$, and then rotates at the constant angular velocity $\omega_s$ $rad/s$.

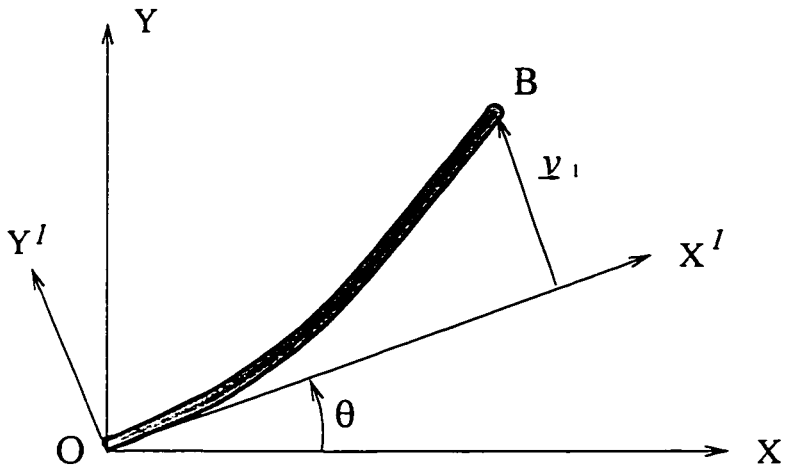



Figure 7.6: The Motion-Driven Spin-Up Beam

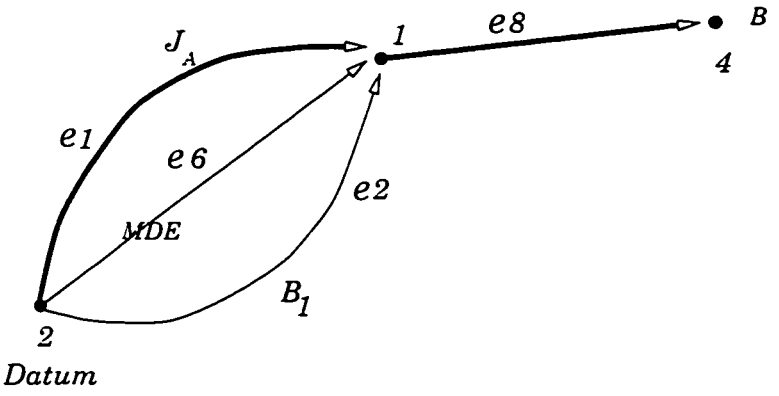



Figure 7.7: Graph for the Motion-Driven Spin-Up Beam

The system graph is constructed first, as shown in Figure 7.7. The flexible beam is represented by edge $e2$, which starts from the origin of the inertial frame and ends at the origin of the body frame. The pin joint is represented by joint edge $e1$ and the motion driver by edge $e6$ parallel to the joint edge.

From this graph, the bold joint edge is selected into the tree (a joint tree), and $\theta 1(t)$ and $\mathbf{q}_f$ (the generalized elastic coordinates for the beam) are thus the branch coordinates $\mathbf{q}$.

A Maple input file (see Appendix **A**.4 along with partial output) is created and then

run to generate system equations of order one in the generalized elastic coordinates.

Two lowest possible Taylor monomials for the axial deformation and two for the lateral deformation are used to discretize the deformation variables. They are

$$u_1 = xul_1 + x^2 ul_2 \tag{7.2}$$

$$v_1 = x^2 vl_1 + x^3 vl_2 \tag{7.3}$$

Since this is an open-loop system, the system equations are second-order ordinary differential equations. Numerical solution is carried out with the NAG implicit $D02NBF$ for these ODEs.

The tip deflection profile of the beam predicted by DynaFlex is shown in Fig.7.8, whose peak value is 0.573 m, which is almost exactly the same (0.570 m) as presented in [40, 11, 88].

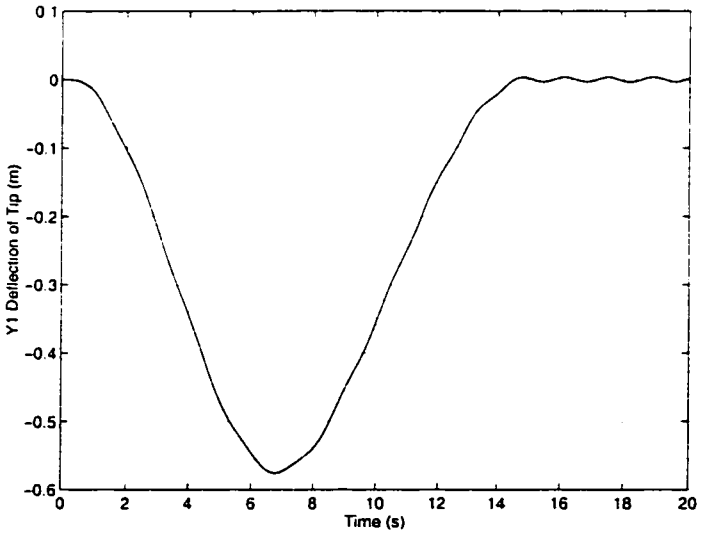



Figure 7.8: Tip Deflection of the Motion-Driven Spin-Up Beam

Experiments have also been carried out with different numbers and types of monomials used for discretizing the axial and lateral deformation variables, and different numerical

solvers as well. To clearly define the number of monomials of the same type used to discretize the deformation variables of a beam, the notation *uabcd* is used to denote the discretization scheme that, as mentioned in Chapter 4, uses $a$ monomials for the axial deformation, $b$ monomials for the deflection in the $Y$ direction of the beam's body frame, $c$ monomials for the deflection in the $Z$ direction and $d$ for the angle of twist. The $u$ in the denotation serves as an identifier for a discretization scheme. Note that $a+b+c+d$ is the number of generalized elastic coordinates $\mathbf{q}_f$ for the beam.

A tip deflection of well within two percent of 0.57 m has been predicted when two to five Taylor monomials are used for the bending variable $v_1$, between zero and two Taylor monomials are used for the axial deflection variable $u_1$, and when the NAG implicit $D02NBF$ solver is called. This solver becomes unstable, however, with the discretization scheme $u0600$ in Taylor monomials, which means that, except for the $Y$ deflection discretized with six monomials, other deformation (stretch) variables are set to zero. It predicts the correct tip deflection, on the other hand, with the same deformation discretization scheme when both the Chebyshev and Legendre polynomials are selected.

For the discretization scheme $u2600$ using the Taylor polynomial, the solver regains stability, but produces inaccurate results. It predicts a larger tip deflection when the complete second-order elastic rotation matrix is used than when the first-order elastic rotation matrix is used, as expected.

It is encouraging to point out that although the $D02NBF$ failed for scheme $u0600$ with Taylor monomials, the HHT-$\alpha$ method succeeded. In fact, it succeeds even for $u6700$ with Taylor monomials, i.e., when six Taylor monomials are used for the axial deformation and seven for the $Y$ direction deflection.

Experiments also show that, for this particular problem, the employment of the traditional first-order deformation field and the new complete second-order one does not make any difference in the tip deflection when a solver is able to give the correct tip deflection.

The $CPU$ time taken on a Silicon Graphics Indigo 2 XZ workstation for the numerical solution of equations obtained using the scheme $u0500$ with Taylor discretization of deformation variables is around 2.81 seconds with the $D02NBF$ solver, and 0.47 seconds with the HHT-$\alpha$ method solver. It is worthy to note that although the NAG solver is considerably slower than the HHT-$\alpha$ method, both are only a fraction of the real time (20 $s$), and both solutions give an accurate tip deflection.

To compare the consumption of computer resources by the DynaFlex generation of system symbolic equations with different types of discretization monomials, the $u0500$ scheme is taken again. With other factors unchanged, DynaFlex spends $2047K$ in Maple memory and 26.2 second in $CPU$ time for a Taylor monomial discretization, $3199K$ and 91.1 second for a Chebyshev discretization, and $3135K$ and 90.1 second for Legendre monomials. Thus it is seen that Chebyshev and Legendre polynomials are more expensive to use than the Taylor ones, while they each perform almost the same in this regard. Experiments with other schemes also support this statement.

To finish this example, it seems one can conclude that he can start solving a flexible problem by employing the Taylor discretization scheme with two monomials for each deformation variable in the symbolic generation, followed by the HHT-$\alpha$ method for the numerical solution of system equations to quickly get rough numerical results. He can then use more monomials to refine the results. To verify these results, we can select Legendre or Chebyshev monomials in place of the Taylor ones. For this particular problem, accurate results can be predicted with the first-order elastic rotation matrix. This is not always the case, as will be demonstrated in the following example.

## 7.4 Three-Dimensional Spin-Up Beam with Off-Set Tip Mass

This example is also taken from Valembois et al. [88]. The system contains a beam that is subjected to simultaneous torsion, axial loading, and bending about two axes. To my knowledge, no other authors have analyzed this very complex problem. In [88], the geometrically nonlinear beam approach was adopted and beam equations of motion were obtained through the Newton-Euler equations. Thus, all first-order terms in the inertial forces of the system are included even though only a first-order elastic rotation matrix has been utilized.

The physical make-up of the beam is shown in Figure 7.9. The system consists of a 5 $m$ long beam $B_1$ which has the same cross-sectional geometry and material properties as those of the previous spin-up beam, in addition to $G$, the shear modulus, equal to $2.7 \times 10^{10}\ N/m^2$, and a point mass $B_2$ of 5 $kg$ rigidly attached to a massless bar of 1 $m$ length that is welded perpendicular to the beam in the $X^1CY^1$ plane when the beam is undeformed. With gravity acting along the negative of the $Z^1$ coordinate and the motion driver specified as in equation (7.1), the tip deflection of the beam and the deflection of the point mass are to be found when the system starts from its static equilibrium position. As usual, the system graph is first constructed based on the components involved, as shown in Figure 7.10. The flexible beam is modelled with edge $e2$, a flexible body element, the joint at $C$ with edge $e1$, a revolute joint edge, the motion driver with edge $e6$, a motion driver of joint type. The weld joint at $A$ is represented by joint edge $e5$, whose two points of connection are located with two arm elements: edges $e4$ and $e8$. Edge $e4$ is a flexible arm element since it resides on the beam while edge $e8$ is a rigid arm that starts from the off-set mass and ends at the other end of the massless connecting rod, i.e., the connection point of the weld joint on the massless rod. The mass is modelled with edge $e3$, a rigid body element with all of its moments of inertia set to zero. The

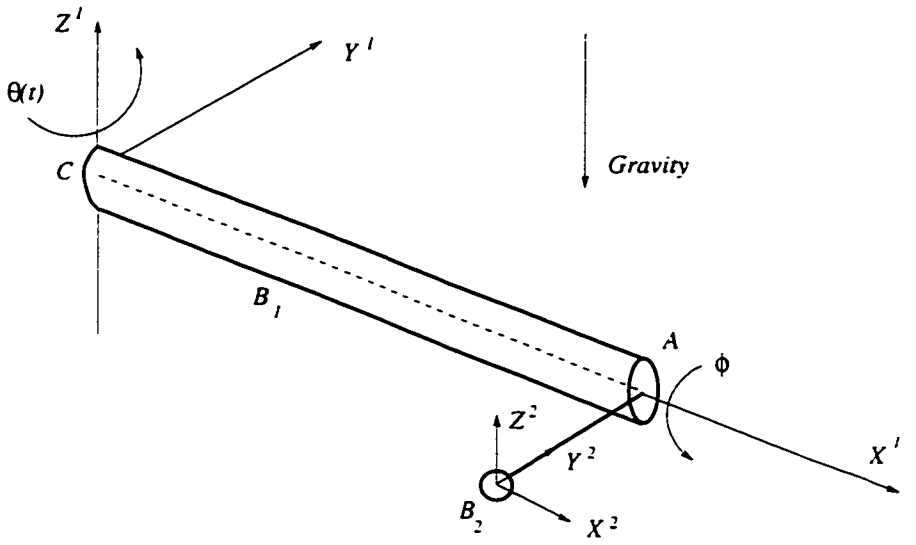



Figure 7.9: The Three-Dimensional Spin-Up Beam with Off-Set Tip Mass

weight of the mass is represented by edge $e7$, a force driver of position type. The weight of the beam is included in its virtual work terminal equation, as discussed in Chapter 4.

With the bold edges selected into the tree, a joint tree, the Maple input file (see Appendix A.5) is created and run to get the system symbolic equations truncated to $O(1)$ in the generalized elastic coordinates. After system geometric and material properties are substituted, these equations are output in Fortran so that they are then solved with the implemented HHT-$\alpha$ method.

Numerical results for the deformations of the tip of the beam in its local frame, from two runs of DynaFlex using two Taylor monomials for each of the four deformation variables of the beam (i.e., u2222), are presented in Figures 7.11–7.13, along with the results obtained in [88]. The first run is executed with the traditional $O(1)$ elastic rotation matrix, i.e., the $O(1)$ deformation approximation for the beam. Results are shown in dashed lines. The second run is with the complete $O(2)$ elastic rotation matrix proposed earlier in this thesis, thus the complete $O(2)$ deformation approximation, and results are

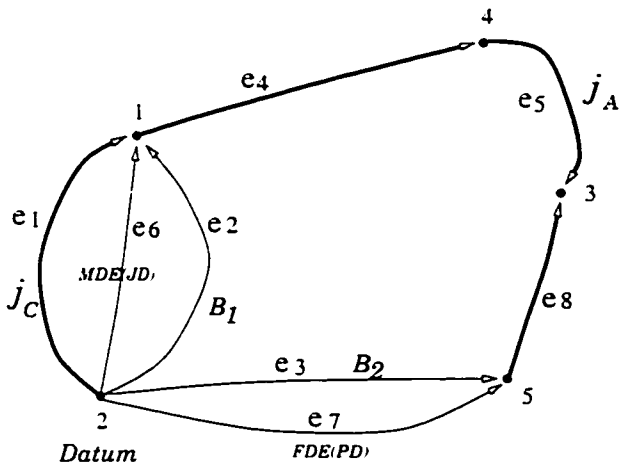



Figure 7.10: Graph for Three-Dimensional Spin-Up Beam with Off-Set Tip Mass

shown in solid line. Lines of the dot-dash style are results from [88]. In [88], the twist angle of the beam's tip is not reported. It is observed that, except for the twist angle at the beam's tip and the $Z$ deflection of the mass, the three sets of results are well within 5% of one another. The twist angles, and thus the $Z$ deflection of the tip mass, are too far apart from each other to ignore. The angle predicted with the $O(1)$ deformation field is 0.0212 *rad* (about 1.2°) while that with the complete $O(2)$ deformation field is 0.0089 *rad* (about 0.5°). Thus there is a sizable difference between them. The results also implies that the beam that utilizes the $O(2)$ deformation field becomes softer than the one that uses the $O(1)$ deformation field when twisted, which makes sense since adding more terms in the deformation field generally means that the beam is allowed more latitude to deform.

Now look at the subplot of the $Z$ deflection of the mass. With the same initial value, its minimum deflection predicted by DynaFlex with the $O(1)$ deformation field is $-0.139$ $m$, and that with the $O(2)$ is $-0.124$ $m$. Thus the difference between them is quite appreciable—well over 10%. While the $O(2)$ deflection agrees well with results in [88] (in spite of a small percentage of difference in their initial values), the $O(1)$ results do not.

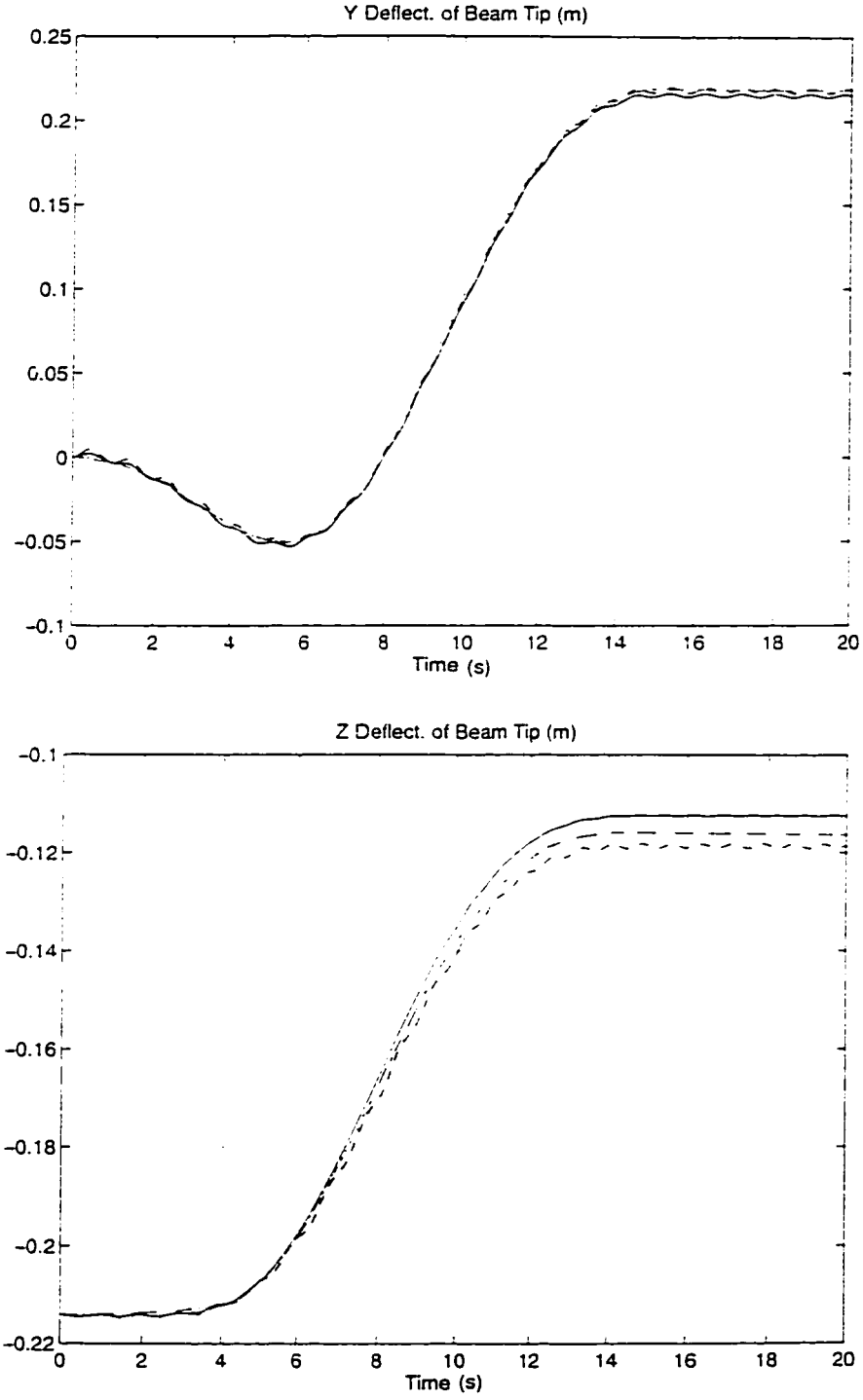



Figure 7.11: Deformations of Spin-Up Beam with Off-Set Tip Mass (a)

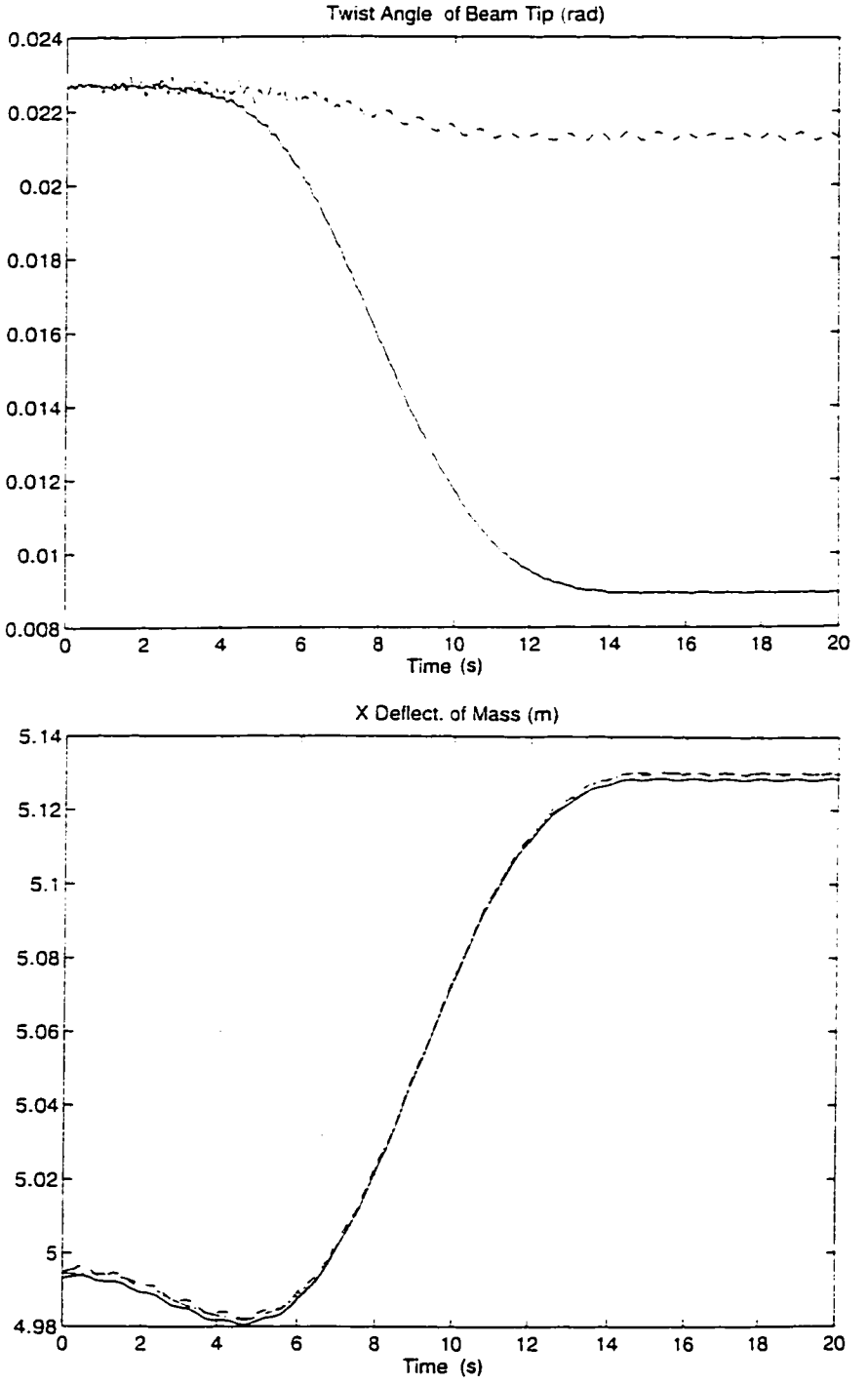



Figure 7.12: Deformations of Spin-Up Beam with Off-Set Tip Mass (b)

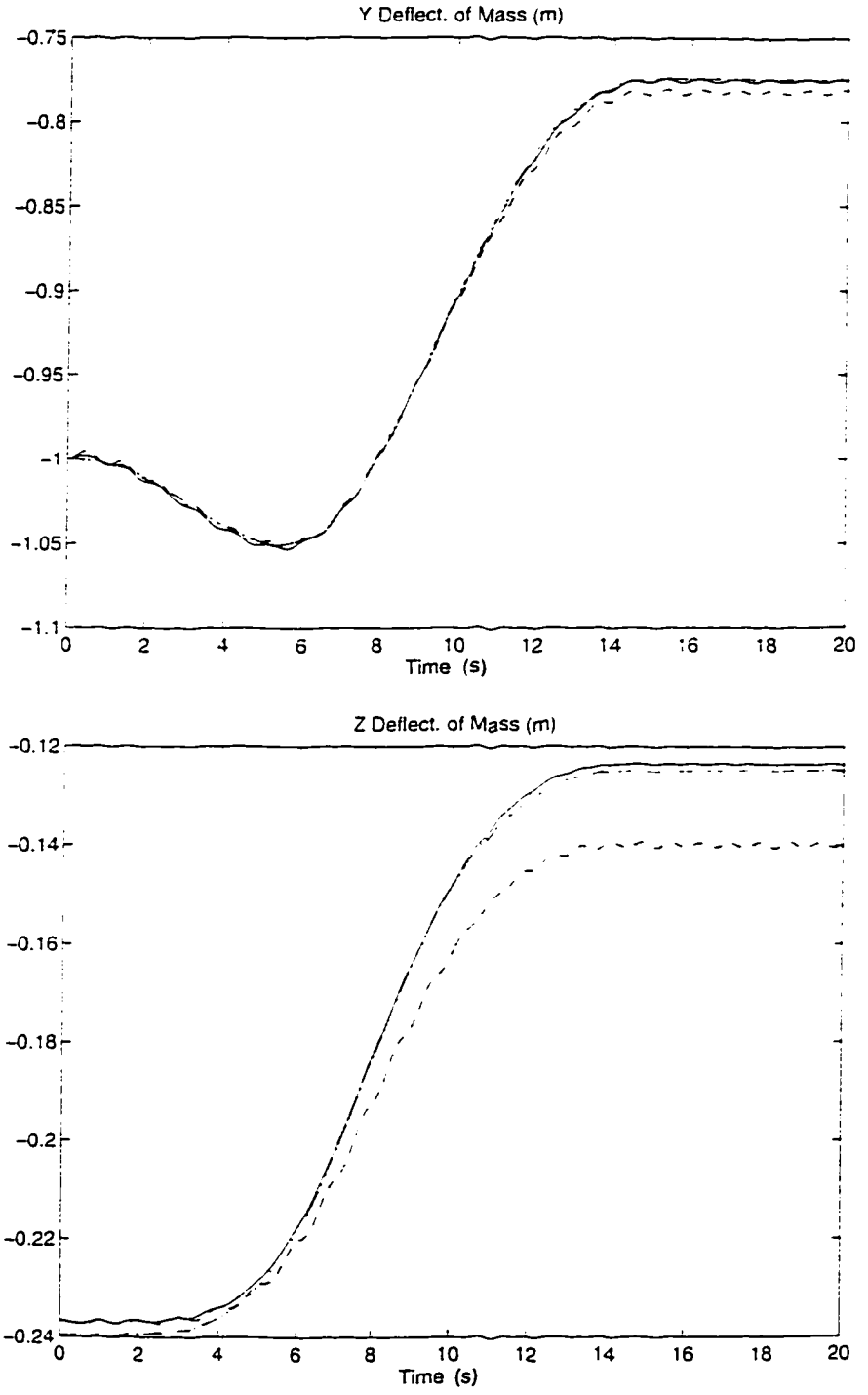



Figure 7.13: Deformations of Spin-Up Beam with Off-Set Tip Mass (c). solid line–DynaFlex $O(2)$ results; dashed line–DynaFlex $O(1)$ results; dot-dashed line–results from [88]

This set of results have also been supported by using finer discretization schemes, other types of monomials, and higher-order system equation. Specifically, both $u0333$ and $u0444$ in Taylor monomials combined with the HHT-$\alpha$ solver produced the same set of deformations. Scheme $u2222$ with Legendre monomials and the HHT-$\alpha$ solver also produced the same results. The $O(2)$ system equations resulting from scheme $u2222$ in Taylor monomials and solved with the HHT-$\alpha$ method gave the same results as well.

With other factors remaining unchanged, experiments show that DynaFlex works twice as efficiently with Taylor monomials than with Legendre ones in terms of Maple memory space required and $CPU$ time, for the cases investigated.

In short, it is seen that when the floating frame approach is adopted in dealing with flexible Bernoulli-Euler beams, a complete second-order deformation field is of paramount importance for some cases. As pointed out earlier in the thesis, failure to use a complete second-order deformation approximation results in the loss of some first-order terms in the inertial forces of the system equations. The current example is a numerical manifestation of this statement. The missing first-order terms can be identified with the current symbolic approach, if so wished.

## 7.5 Planar Flexible Two-Link Manipulator

This manipulator, subjected to two joint torque drivers, moves in a plane free of gravity, as shown in Figure 7.14. Both links have the same length of 0.8 $m$, area of 0.0004 $m^2$, mass density of 7850.00 $kg/m^3$, and Young's modulus of $2.0 \times 10^{11}$ $N/m^2$. The area moment of inertia for link 1 is $5.333 \times 10^{-8}$ $m^4$, and $1.333 \times 10^{-8}$ $m^4$ for link 2. The two torque drivers are both specified as $T = 20sin(10t)$ $N \cdot m$ each.

The system graph is shown in Figure 7.15. The pin joint is modelled with edge $e1$, the force driver of joint type with edge $e8$, and body 1 with edge $e6$. Flexible arm $e2$ is

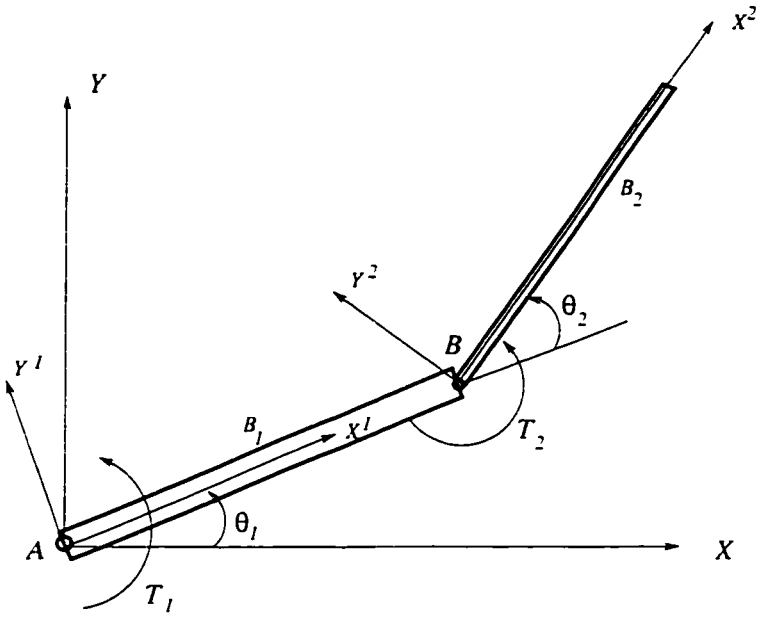



Figure 7.14: The Planar Flexible Two-Link Manipulator

used to locate pin joint $B$ in body 1's reference frame $X^1Y^1Z^1$. Joint $B$ is modelled with joint edge $e4$, and the force driver of joint type at the same joint with edge $e9$. Flexible arm $e10$ is employed to locate the tip of link 2. It also gives the deflection of the tip. Finally, body 2 is represented by edge $e7$.

Accordingly, a Maple input file is constructed with each link being discretized using the scheme $u1200$ in Taylor monomials; see Appendix A.7. With this, DynaFlex is executed with the $O(2)$ elastic rotation matrix, numerical system parameters are substituted, symbolic system equations of first order in generalized elastic coordinates are exported in Fortran, and solved with the HHT-$\alpha$ method.

The tip deflection of the outer link along the local $Y^2$ axis is shown in Figure 7.16 together with the results produced by ADAMS [45].

The maximum amplitude of DynaFlex's results is 0.0147 $m$ while that given by ADAMS is 0.0141 $m$. Thus, it is seen that the predicted tip deflection by DynaFlex

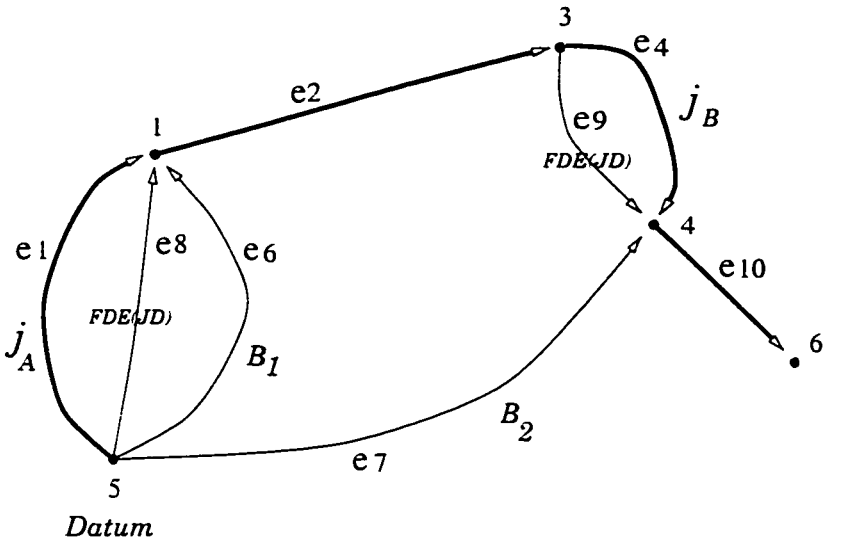



Figure 7.15: System Graph of the Planar Flexible Two-Link Manipulator

is within 5% of that by ADAMS. Note that the DynaFlex results are probably more accurate because of the fact that the ADAMS model uses only the lumped mass and moment of inertia approach while DynaFlex uses the consistent mass matrix approach to representing system inertial forces, and the consistent mass matrix is generally more accurate [114].

DynaFlex results from using more refined schemes, such as two monomials in Legendre for the axial deformation and three for the bending for each link, also agree with the results obtained above, and employment of the $O(1)$ elastic rotation matrix has almost no effect on the deflection, as in the planar spin-up beam example.

Experiments still show that DynaFlex works more efficiently with Taylor monomials than with Legendre ones. For example, DynaFlex spends 10430 $K$ of Maple memory and 1399 $s$ of $CPU$ time on the scheme $u0300$ in Legendre for each of the links for generating system equations, whereas it spends only 4095 $K$ of Maple memory and 330.3 $s$ of $CPU$ time on the same scheme in Taylor. Thus, once again one sees that Taylor monomials

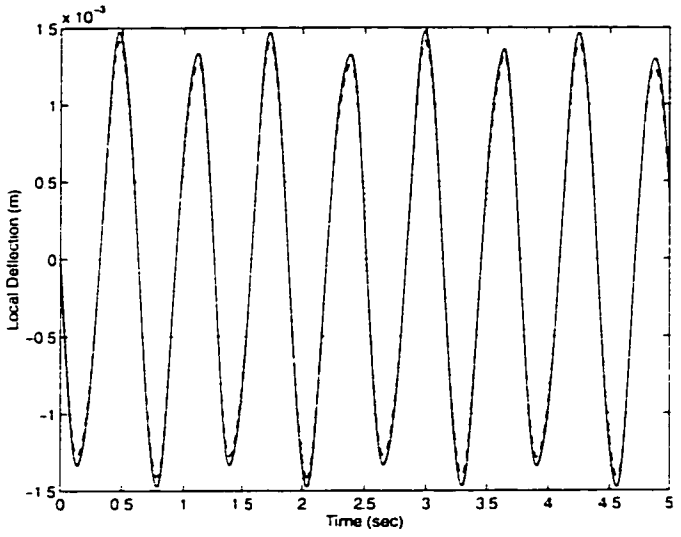



Figure 7.16: Tip Deflection of the Planar Flexible Two-Link Manipulator: solid line–DynaFlex results; dashed line–ADAMS results

allow DynaFlex to work much more efficiently than Legendre ones, as far as the symbolic generation of equations is concerned.

## 7.6 Spatial Slider-Crank with Flexible Link

This spatial slider-crank is taken from [114]. A planar version of it is investigated first, also for comparison to [114].

This planar system, together with the coordinate systems used, is shown in Figure 7.17. The crank is considered as rigid and it has a length of 0.15 $m$. The link of 0.3 $m$ length is flexible, having a circular cross-sectional area of $2.826 \times 10^{-5}\ m^2$, area moment of inertia of $6.361 \times 10^{-11}\ m^4$, Young's modulus of $2.0 \times 10^{11}\ N/m^2$ and a mass density of $7.87 \times 10^{3}\ kg/m^3$. The rigid slider has a mass of $3.34 \times 10^{-2}\ kg$, which is half that of the link. A constant rotational motion driver of 150 $rad/s$ acts on the rigid crank.

The flexible link is assumed to be initially straight along the global $X$ axis and, during

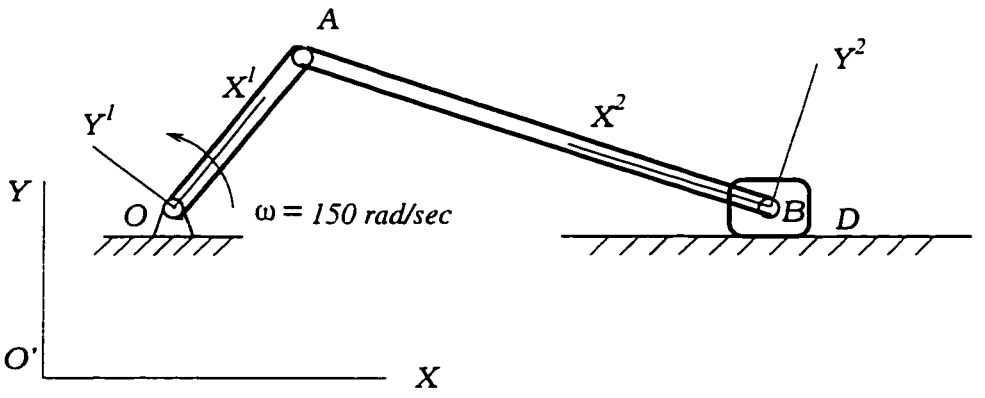



Figure 7.17: The Planar Flexible Slider-Crank

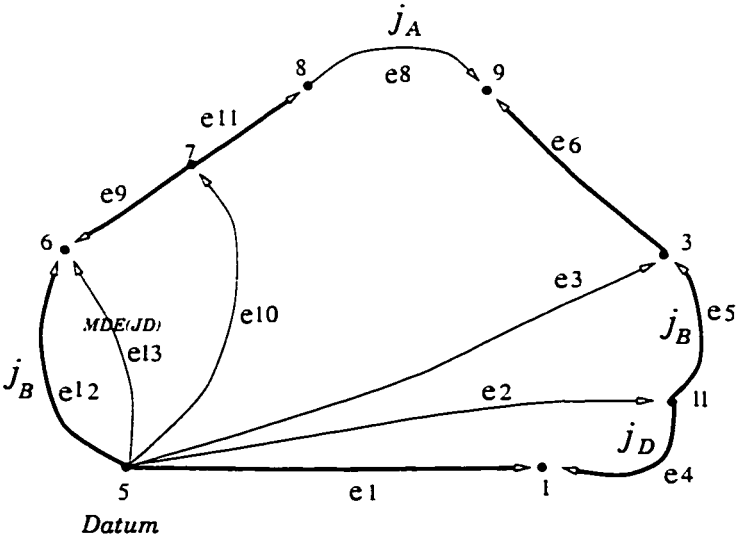



Figure 7.18: Graph of The Planar Flexible Slider-Crank

motion, its axial deformation is neglected.

The system graph is constructed (see Figure 7.18), as discussed in Chapter 5. With pin joint $A$ selected into the cotree, the system branch coordinates are $\theta 12(t)$, $s1(t)$, $\theta 5(t)$, and $\mathbf{q}_f$ (the generalized elastic coordinates for the link), and there are two constraint equations. Furthermore, since $\theta 12(t)$ is specified, the system degrees of freedom are equal to the number of $\mathbf{q}_f$.

The Maple input file (see Appendix **A**.8) is created with a discretization scheme of $u0300$, i.e., with only bending in the local $Y^2$ direction, in Taylor monomials for the link.

To solve the symbolic system equations generated by DynaFlex, the index 2 DAEs for the system are converted in Maple into their state space form. They are then exported in Fortran, and solved with the HHT-$\alpha$ method (see Figure 6.6). The plotted solution for the mid-point deflection of the link in the $Y^2$ direction is shown in Figure 7.19 in solid along with the results recorded in [114]. It is seen that the two sets of results agree well with each other.

Experiments have also been carried out with other numerical solvers for the same discretization scheme. The NAG *D02NAF* failed to converge. It entered infinite loops in the Newton iteration. The BDF Euler and Trapezoidal methods were successful.

Further numerical experiments show that while both the HHT-$\alpha$ method and the BDF Euler were capable of solving equations using up to four Taylor monomials, the Trapezoidal rule succeeded with six of the monomials.

The spatial slider-crank is shown in Figure 7.20. This mechanism is obtained by rotating the joint axis at $O$ by 45° about the global $Y$ axis so that the crank spins to form a cone whose apex is located at $O$. Instead of the pin joints at point $A$ and $B$, as in the planar case, now there is a ball joint at $A$ and a universal joint at $B$. As for the planar case, the crank and the link are initially aligned with the global $X$ axis. At this configuration, the flexible link's local body reference frame $BX^2Y^2Z^2$ is coincident with

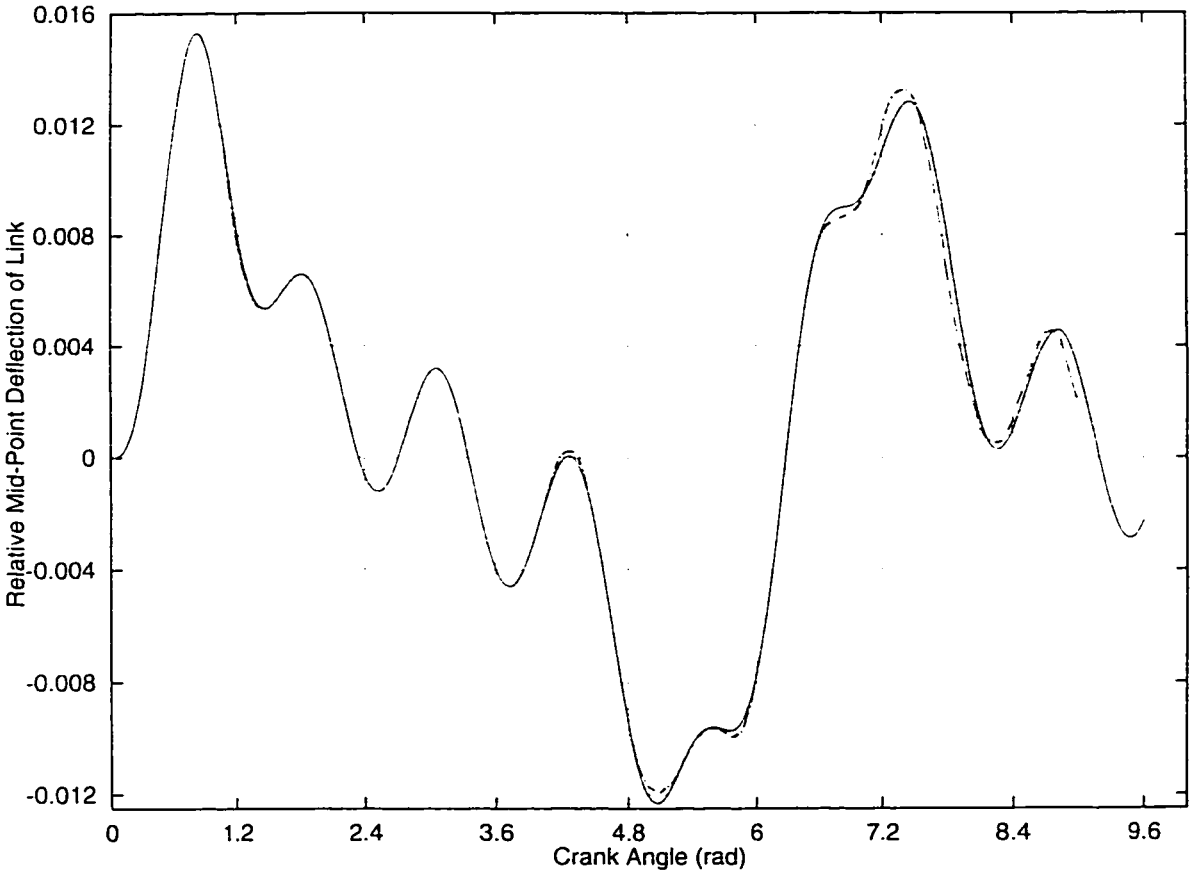



Figure 7.19: Mid-Point $Y^2$ Deflection of the Planar Link: solid line–DynaFlex results; dot-dashed line–results from [114]

that of the slider, i.e., $BX^3Y^3Z^3$.

For this example, both the longitudinal deformation and the twist of the link are suppressed. Furthermore, it is assumed that the rotatory inertia of the link is negligible too. The graph of the system, shown in Figure 7.21, is the same as for the planar case

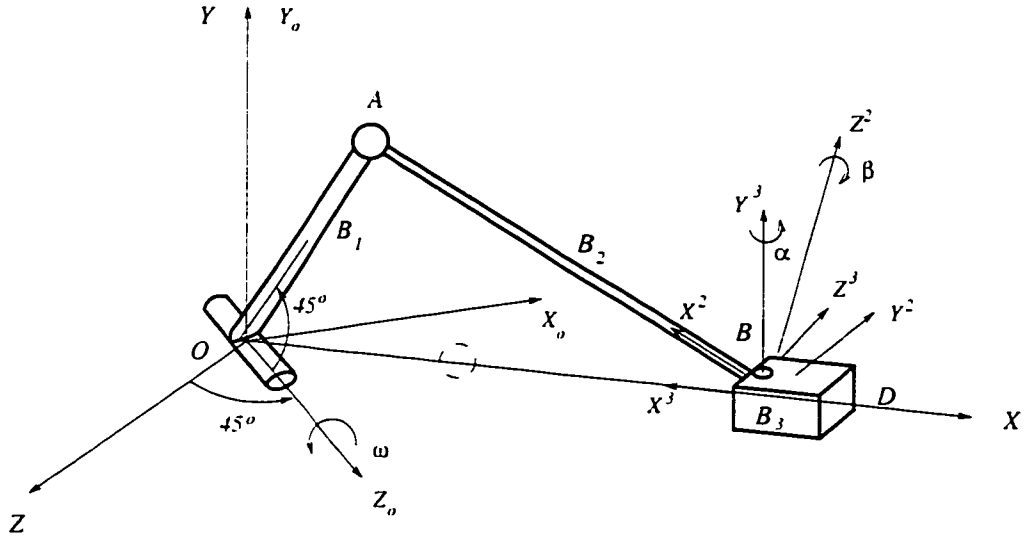



Figure 7.20: The Spatial Flexible Slider-Crank

except for edge 16. This arm element, whose length is zero, is used by virtue of its *rotmat* index to represent the fact that now the axis of the pin joint at $O$ is no longer coincident with the global $Z$ axis, but rotated a certain angle about the global $Y$ axis.

The Maple input file is included in Appendix A.9. It was executed with three Taylor monomials for each of the bending deformations (scheme u0330), kinetic equations up to second order in the generalized elastic coordinates were generated, and the HHT-$\alpha$ method was used to solve them. The numerical results are plotted as solid lines in Figs 7.22 and 7.23, where the results recorded in Jonker [114] are presented as dot-dashed lines. It is seen that DynaFlex's numerical solution is very close to Jonker's.

Three Legendre monomials were also employed in place of the Taylor ones. Numerical

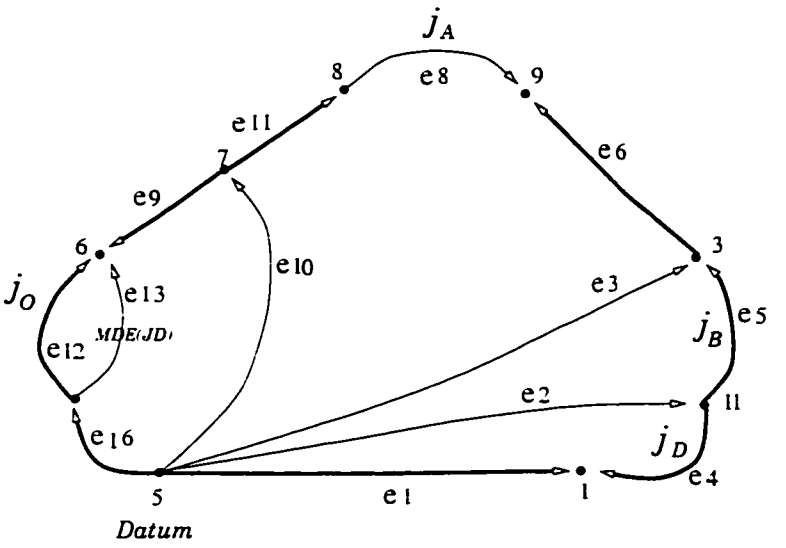



Figure 7.21: Graph of the Spatial Flexible Slider-Crank

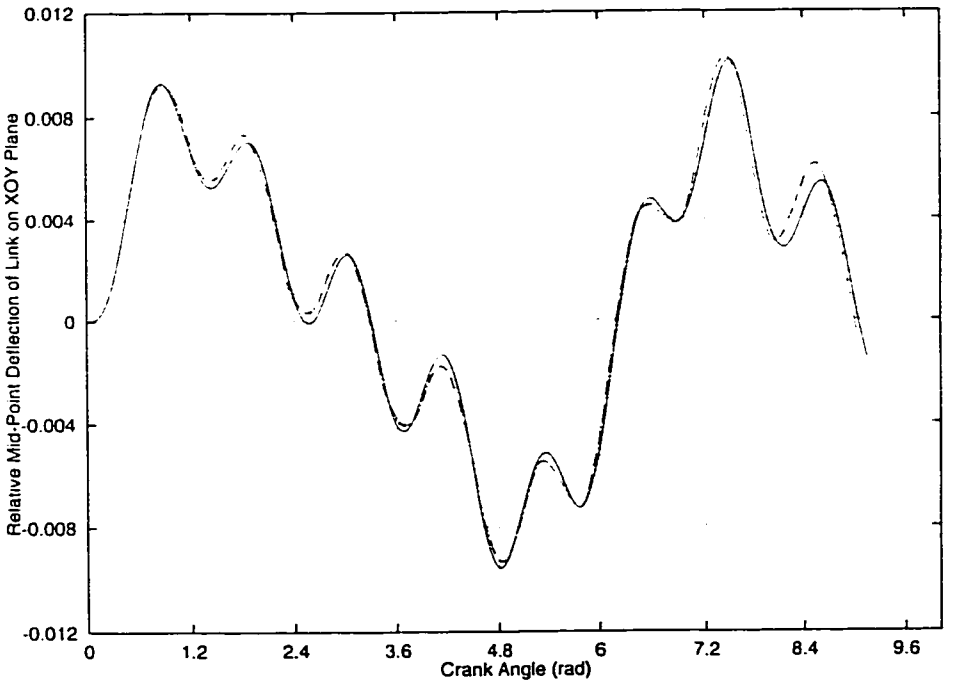



Figure 7.22: Mid-Point Deflection of Link on XOY Plane: solid line– DynaFlex results; dot-dashed line–Jonker [114]

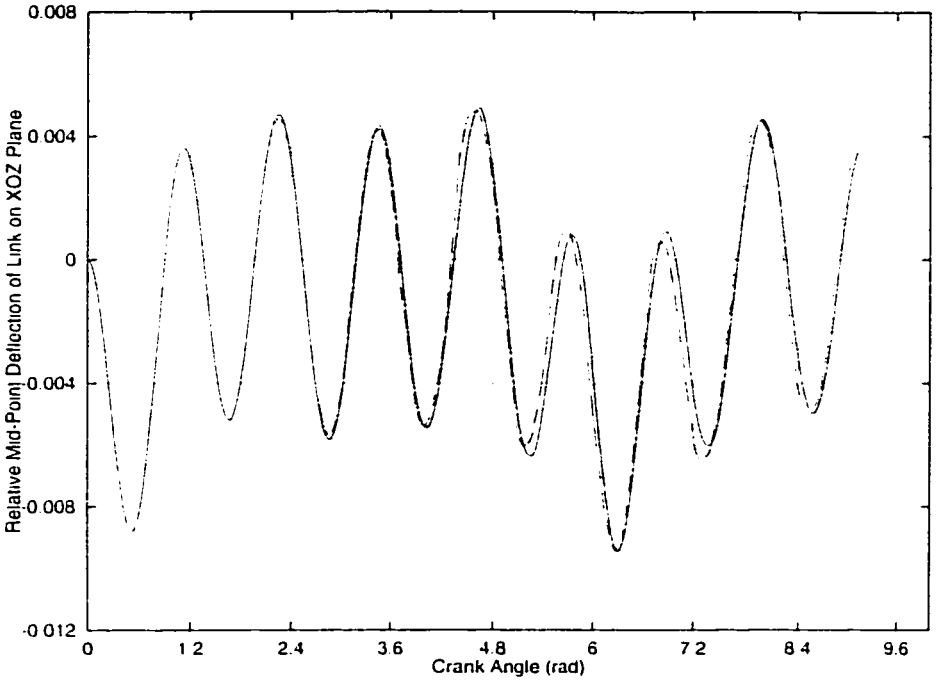



Figure 7.23: Mid-Point Deflection of Link on XOZ Plane: solid line– DynaFlex results; dot-dashed line–Jonker [114]

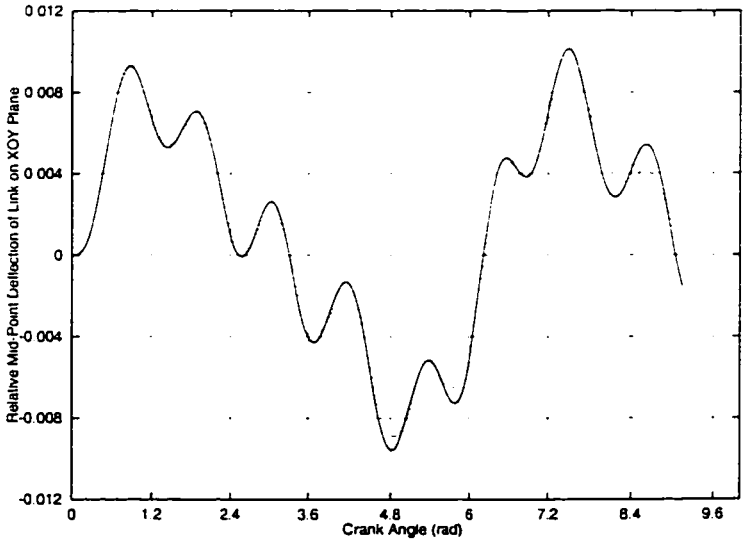



Figure 7.24: Numerical Results from $O(1)$ and $O(2)$ System Equations: solid line–$O(2)$ results; dotted line–$O(1)$ results

results obtained were the same as presented here. With all other factors unchanged, the DynaFlex run with the Taylor monomials consumed only about half the Maple memory and $CPU$ time it consumed with Legendre monomials. Ironically, the *simplify* command applied to the elements of the mass matrix and the force column of the kinetic equations not only slowed down the execution markedly, but it generated Fortran files about twice as large as when it was not used.

For other examples considered thus far that have flexible bodies, retaining only first-order terms in generalized elastic coordinates in system kinetic equations has not given different numerical results from retaining terms of up to second order. This, however, is not the case with the current spatial slider crank. Shown in Figure 7.24 is a comparison of the results (in dotted line) from solving the system equations that are comprised of the first-order kinetic equations and second-order kinematic loop closure equations and results (solid line) from the system equations that are comprised of the second-order kinetic equations adjoined with the same kinematic equations. There is an appreciable disparity between them. This is obviously caused by the neglect of the second-order terms in the kinetic equations. Further numerical experiments showed that neglecting the second-order terms in the elastic forces did not affect the results. Thus the second-order inertial forces are important, and must be included in the simulation. This finding complements the conclusion reached by Damaren and Sharf [56] that for fast maneuvers of a manipulator, all terms are required for an accurate simulation.

In addition, with the HHT-$\alpha$ method, the system equations with both second-order kinetic equations and algebraic constraint equations have much better numerical properties than the system equations with only first-order kinetic equations. The HHT-$\alpha$ method solved the $O(2)$ system equations with a combination of the parameters HHT-$\alpha$ and time step (in seconds) that range from $-0.31$ to 0.01 and $10^{-4}$ to $10^{-6}$ respectively, whereas it only succeeded with the other set of equations on very narrow intervals of

the same parameters, with the Newton iteration matrix running into singularities outside those intervals.

It is also interesting to point out that although second-order inertial forces are important for an accurate simulation, employment of the traditional first-order deformation displacement field for the link did not make any difference in the results.

## 7.7 Concluding Remarks

The implemented symbolic-numerical computer package works well for the tested problems, in terms of both the symbolic or numerical results it generates and the computer resources it requires to run. This in turn indicates that the proposed VW graph-theoretic approach is valid. The approach's ability to reduce the number of system equations has been demonstrated in the rigid spatial slider-crank example.

The new second-order elastic rotation matrix for Bernoulli-Euler beams is important. Applying the traditional first-order elastic rotation matrix in place of it will result in incorrect solutions for some problems. Second-order terms in system kinetic equations are indispensable for better numerical properties and more accurate simulations for some problems too. It is also found with the flexible spatial slider crank example that for some problems, second-order terms in the system kinetic equations can be important. However, it remains to be seen whether other second-order terms that are missing in the kinetic equations, due to the use of only a second-order deformation field, would have additional effects.

Taylor, Chebyshev, and Legendre monomials can all be used as the base functions in the Rayleigh-Ritz method to approximate the deformation variables of a Bernoulli-Euler beam. While Taylor monomials save on $CPU$ time and memory space with the symbolic generation, Chebyshev and Legendre monomials tend to produce better-conditioned sys-

tem equations that are more amenable to numerical solution.

In case of the incompetency of some ODE solvers, one should try others, but experience so far shows that the HHT-$\alpha$ method seems to be very versatile.

Maple as a programming language should be improved in its ability to handle large problems that often occur in engineering. Specifically, the matrix multiplication operation needs improving so that it works faster than using the *do* loop approach, and is able to carry out the operation with the objects that are at least as large as those that the *do* loop can handle. In addition, it is desirable that the *simplify* command be made less time and memory-space consuming to use, and when it is used in combination with the *Fortran* command to output files in Fortran, the output files are as small and efficient as possible in terms of the number of operations needed to numerically evaluated them. This is especially important for large problems since it is impossible to obtain a closed form solution for them and numerical approximation seems to be the only option.

# Chapter 8

# Closure

The goal of extending existing graph-theoretic methods for multibody dynamics to include flexible bodies has been successfully completed using the novel idea of adopting virtual work as a through variable for a new set of elemental graphs. The validity of virtual work (VW) as a though variable has been demonstrated both philosophically, mathematically, and with general examples. The new VW graph-theoretic (GT) approach treats both rigid body and flexible body systems in the same fashion, all in terms of their virtual work. An additional feature of this new approach is that it may reduce the number of system equations as compared with the conventional absolute or joint coordinate formulation for multibody systems with closed loops.

New GT elements have been created. They include the flexible body element, the flexible arm element, and the dependent VW element. Construction of a system graph has been demonstrated with examples, and only one graph is needed for each system. With the graph-theoretic models of common components presented in this thesis, each edge of a system graph also has the traditional across and through variables associated with it; these traditional across and through variables satisfy their respective topological equations. This means that the new graph-theoretic approach subsumes most of the

existing approaches to multibody dynamics as special cases.

A systematic procedure for formulating system equations, including kinetic and kinematic constraint equations, has been put forward that preserves the methodical nature of traditional GT methods. The proposed procedure can produce DAEs in either index 3 or index 2 form, depending on whether the joint constraint equations at the variational level are substituted into the system VW equation(s) to extract the system kinetic equations. With a body tree, system kinetic equations are extracted from system VW equation(s) independently of the joint constraint equations, generating the same set of index 3 system DAEs in absolute coordinates as a pure mechanics approach does (Haug [111]). With a joint tree, system kinetic equations are extracted from a single system VW equation after the joint constraint equations at variational level have been substituted, generating a set of index 2 system DAEs which are free of Lagrange multipliers, i.e., LMF index2 DAEs. In addition, the joint tree approach is capable of reducing the number of system DAEs compared with existing pure mechanics approaches, as demonstrated in the Rigid Spatial Slider-Crank example. The key to the reduction is the combination of the century-old zero-virtual-work property of ideal joints with the novel idea of adopting virtual work as a through variable in the GT approach.

Modelling of flexible bodies in the context of multibody dynamics has been discussed. Dynamics of a Bernoulli-Euler beam (the simplest mechanical model of flexible bodies) has been revisited with a focus on its very fundamental three-dimensional kinematics, in particular, the deformation displacement field approximation that has been in use for hundreds of years. The surprising finding that the commonly-used deformation field approximation is unable to produce a complete $O(2)$ elastic rotation matrix precipitated a months-long search for a more advanced deformation displacement field approximation that is capable of generating an elastic rotation matrix complete up to any desired order.

The new complete O(2) deformation displacement field has been found to be of crucial

importance in modelling some problems that involve beams, as has been illustrated by the Three-Dimensional Spin-Up Beam with Off-Set Tip Mass. The same idea used to develop the complete O(2) deformation field is applicable to obtaining deformation fields of higher orders.

Although Alkire [104] introduced the two-Euler angle approach to the elastic rotation matrix of a Bernoulli-Euler beam, and consequently Pai and Nayfeh [107] borrowed it, none of them used or even mentioned the potential of the approach to constructing a complete $n$-th order deformation displacement approximation. Furthermore, the foreshortening of the beam was not included in their papers.

The problem-independent approach (Jen [93], Valembois, Fisette and Samine [88]) to discretizing the deformation variables of a beam has been extended to include Chebyshev and Legendre polynomials. Numerical experiments have shown that they are good alternatives to the Taylor polynomials, especially in terms of the numerical properties of the resulting system equations.

A computer implementation has been developed for the joint tree formulation of the proposed VW graph-theoretic approach. It generates index 2 DAEs for systems with closed loops and second-order ODEs for systems with open loops. A user with minimal knowledge of graph theory can learn to use it within a couple of hours. The implementation consists of two distinct parts. The first is the Maple generation of symbolic system equations. It generates system equations given only a system description as input. The second is a numerical solution of the system equations, executed in Fortran. Different numerical solvers that are capable of handling either first-order ODEs or index 1 DAEs, to which the symbolic system equations are readily converted either in Maple or in Fortran, have been implemented.

The four types of numerical solvers, the NAG ODE solvers, the HHT-$\alpha$ method, the BDF Euler, and the Trapezoidal rule, have performed well for the examples presented in

the last chapter. Numerical difficulties have been encountered by some solvers but not others for the same set of system equations, and no one single solver has been the most effective for all of the problems. Note that when the crank of the spatial slider crank mechanism with a flexible link presented in the last chapter is mounted perpendicular, instead of 45°, to its rotating axis, none of the solvers is able to solve accurately. The conclusion is that DAEs for flexible systems are much more difficult to solve than those for rigid systems.

Six example problems have been solved with the computer package. They range from rigid to flexible systems, from open to closed loop, including both forward, inverse, and hybrid dynamics problems. All the results, either numerical or symbolic, have been compared favorably to those recorded in the literature or obtained from existing multibody software. The successful solution of these examples also serves to support the new VW graph-theoretic approach.

For some flexible problems, a set of $O(2)$ system equations are required in order to achieve modelling fidelity, and the ruthless linearization of system equations with respect to the generalized elastic coordinates $\mathbf{q}_f$ (Padilla and Flowtow [16]) is not enough, as shown in the example of the spatial slider crank with a flexible link.

As to Maple as a programming language, it is powerful enough to handle all the operations involved in the formulation procedure. However, there are some anomalies, such as its inability to perform the matrix multiplications with its *multiply* command that an old-fashioned *do* loop can achieve when the objects are large.

Before closing the chapter, four future research directions are suggested.

The presented GT approach can be used with other types of flexible bodies because of the general definition for the virtual work terminal equation for a flexible body. First the Timoshenko beam can be incorporated. It is suspected that a new complete second- or even higher-order deformation displacement field approximation for the beam is also

needed for simulation fidelity for some systems, as it is for the Bernoulli-Euler beam. This approximation can be obtained by expanding its elastic rotation matrix (a general three-dimensional rotation matrix) up to the desired order. A similar second- or higher-order deformation field approximation might also be required of more complex bodies, such as plates and shells.

It is also worthwhile to explore combining the projection method [103] with the new GT approach so that a further reduction in the number of system equations can be achieved for some problems.

Extension of the new GT approach to incorporate the inertial frame approach to flexible multibody dynamics can also be done without much difficulty. Within the current framework, each flexible body can first be divided into a number of finite elements. Each finite element can be regarded as a flexible body itself, and its virtual work can be calculated accordingly in terms of the absolute coordinates that are employed to describe its configuration. The problem of having too many coordinates and too dense a graph may occur. This could be dealt with using superelements (Cardona and Geradin [51]).

A fourth area where research is needed is to find one or more sets of shape functions that are independent of the boundary conditions of a flexible body and computationally efficient, and from which the resulting system DAEs have good numerical properties at the same time. Closely related to this is the need for more robust numerical solvers for ODEs and DAEs, especially those representing the motion of flexible multibody systems.

# Bibliography

[1] R.L. Huston, Computer methods in flexible multibody dynamics, International Journal for Numerical Methods in Engineering, Vol.32, pp.1657-1668,1991.

[2] C. Fuhrer, B.J. Leimkuhler, Numerical solution of differential-algebraic equations for constrained mechanical Motion, Numerische Mathematik, Vol.59, pp.55-69, 1991.

[3] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical solution of initial-value problems in differential-algebraic equations, Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[4] D.E. Rosenthal, M.A. Sherman, High performance multibody simulations via symbolic equation manipulation and Kane's method, The Journal of Astronautical Sciences, Vol.34, pp.223-238, 1986

[5] B.S. Thompson, C.K. Sung, A survey on finite element techniques for mechanism designs, Mechanism and Machine Theory, Vol.21, pp.351-359, 1986.

[6] A.A. Shabana, Dynamics of multibody systems, 1989 by John Wiley & Sons, Inc.

[7] A.A. Shabana, Computational dynamics, 1994 by John Wiley & Sons, Inc.

[8] F.R. Vigneron, Comments on "mathematical modeling of spinning elastic bodies for modal analysis", AIAA Journal, Vol.13, pp.126-127, 1975.

[9] K.R.V. Kaza, R.G. Kvaternik, Nonlinear flap-lag-axial equations of a rotating beam, AIAA Journal, Vol.15, pp.871-874, 1977

[10] T.R. Kane, A.K. Banerjee, Dynamics of a beam attached to a moving base, Proc. AASS/AIAA Astrodynamics Specialist Conference, Vail, CO, Paper AAS-85-390, 1985.

[11] S.K. Ider, F.M.L. Amirouche, Nonlinear modelling of flexible multibody systems dynamics, Trans. ASME, Vol.56, pp.444-450, 1989.

[12] S. Hanagud, S. Sarkar, Problem of the dynamics of a cantilever beam attached to a moving base, Journal of Guidance, Control, and Dynamics, Vol.12, pp.438-441, 1989.

[13] S.K. Ider, F.M.L. Amirouche, Influence of geometric nonlinearities in dynamics of flexible treelike structures, Journal of Guidance, Control, and Dynamics, Vol.12, pp.444-450, 1989.

[14] W.C. Hurty, M.F. Rubinstein, Dynamics of structures, Prentice-Hall, Englewood Cliff, New Jersey, 1964.

[15] A.K. Banerjee, J.M. Dikens, Dynamics of an arbitrary flexible body in large rotation and translation, Journal of Guidance, Control, and Dynamics, Vol.13, pp.221-227, 1990.

[16] C.E. Padilla, A.H. von Flotow, Nonlinear strain displacement relations and flexible multibody dynamics, Journal of Guidance, Control, and Dynamics, Vol.15, pp.128-136, 1992.

[17] J. Mayo, J. Dominguez, Geometrically nonlinear coupling between axial and flexural modes of deformation of multibody systems, AMD-Vol.141/DSC-Vol.37, Dynamics of Flexible Multibody Systems: Theory and Experiment, ASME, pp.95-103, 1992.

[18] E. M. Bakr, A.A. Shabana, Geometrically nonlinear analysis of multibody systems, Computers & Structures, Vol.23, pp.739-751, 1986.

[19] M.J. Sadigh, A.K. Misra, More on the so-called dynamic stiffening effect, Journal of the Astronautical Science, Vol.43, pp.101-125, 1995.

[20] J.C. Simo, L. Vu-Quoc, On the dynamics of flexible beams under large overall motions—The plane case: part I, Trans. ASME, Vol.53, pp.849-854, 1986.

[21] T.B. McDonough, Formulation of the global equations of motion of a deformable body, AIAA Journal, Vol.14, pp.656-660, 1976.

[22] J.O. Song, E.J. Haug, Dynamic analysis of planar machines, Computer Methods in Applied Mechanics and Engineering, Vol.24, pp.359-381, 1980.

[23] R.C. Winfrey, Elastic link mechanism dynamics, J. Eng. Ind., Trans. ASME B 93, pp.268-272,1971.

[24] A.G. Erdman, G.N. Sandor, R.G. Oakberg, A general method for kino-elastodynamic analysis and synthesis of mechanisms, J. Eng. Ind., Trans. ASME B 94, pp.1193-1205, 1972.

[25] I. Imam, G.N. Sandor, S.N. Kramer, Deflection and stress analysis in high speed mechanisms with elastic links, J. Eng. Ind., Trans. ASME B 95, pp.541-548, 1973.

[26] D.A. Turcic, A. Midha, Generalized equations of motions for the dynamic analysis of elastic mechanisms systems, Journal of Dynamic Systems, Measurement, and Control, Dec., Vol.106, pp.243-248, 1984.

[27] D. A. Turcic, A. Midha, Dynamic analysis of elastic mechanisms systems. Part I: Applications, Journal of Dynamic Systems, Measurement, and Control, Dec., Vol.106, pp.249-254, 1984.

[28] D. A. Turcic, A. Midha, J.R. Bosnik, Dynamic analysis of elastic mechanisms systems. Part II: Experimental Results, Journal of Dynamic Systems, Measurement, and Control, Dec., Vol.106, pp.255-260, 1984.

[29] E.M. Bakr, A.A. Shabana, Timoshenko beam and flexible multibody dynamics, Journal of Sound and Vibration, Vol.116, pp.87-107, 1987.

[30] R.P. Han, Z.C. Zhao, Dynamics of general flexible multibody systems, International Journal for Numerical Methods in Engineering, Vol.30, pp.77-97, 1990.

[31] R.P. Han, M.Z.K. Xu, J. Zu, On the eigensolutions of elastic mechanisms, CSME Mechanical Engineering Forum 1990.

[32] G.B. Sincarsin, P.C. Hughes, Dynamics of an elastic multibody chain: Part A—Body motion equations, Dynamics and Stability of Structures, Vol.4, pp.209-226, 1989.

[33] G.B. Sincarsin, P.C. Hughes, Dynamics of an elastic multibody chain: Part B—Global Dynamics, Dynamics and Stability of Structures, Vol.4, pp.226-244, 1989.

[34] W.C. Hurty, Vibrations of structural systems by component synthesis, Journal of the Engineering Mechanics Division, Proc. ASCE, pp.51-69, 1960.

[35] W.T. Thomson, Theory of Vibration with Application, Prentice-Hall, pp.246-251, 1988.

[36] R.R. Craig Jr., M.C. Bampton, Coupling of substructures for dynamic analysis, AIAA Journal, Vol.6, pp.1313-1319, 1968.

[37] G. M.L. Gladwell, Branch mode analysis of vibrating systems, Journal of Sound and Vibration, Vol.1, pp.41-59, 1964.

[38] C.K. Sung, B.S. Thompson, T. M. Xing, C. H. Wang, An experimental study on the nonlinear elastodynamic response of linkage mechanisms, Mechanism and Machine Theory, Vol.21, pp.121-133, 1986.

[39] E.J. Haug, S.C. Wu, S.S. Kim, Dynamics of flexible machines: A variational approach, Dynamics of Multibody Systems, IUTAM/IFToMM, Udine, 1985, Editors: G. Bianchi and W. Schiehlen.

[40] R.R. Ryan, Flexibility modelling in multibody dynamics, AAS 87-431, pp.365385, 1987.

[41] A.A. Shabana, On the use of finite element method and classic approximation techniques in nonlinear dynamics of multibody systems, International Journal of Nonlinear Mechanics, Vol.25, pp.153-162, 1990.

[42] Z. Xia, C.H. Menq, Modelling and control of flexible manipulators: Part I—Dynamic analysis and characterization, AMD-Vol.141/DSC-Vol.37, Dynamics of Flexible Multibody Systems: Theory and Experiment ASME, pp.105-114, 1992.

[43] T. Belytschko, B.J. Hsieh, Nonlinear transient finite element analysis with convected coordinates, International Journal for Numerical Methods in Engineering, Vol.7, pp.255-271, 1973.

[44] DADS (Dynamic Analysis and Design Systems) user's guide, CADSI Inc., Coralville, Iowa, 1995.

[45] ADAMS: using ADAMS/flex, Mechanical Dynamics Inc., 1997.

[46] MOBILE user's guide and reference manual by Andres Kecskemethy

[47] T. Belytschko, L. Schwer, Large displacement, transient analysis of space frames, International Journal for Numerical Methods in Engineering, Vol.11, pp.65-84, 1977.

[48] J.C. Simo, L. Vu-Quoc, On the dynamics of flexible beams under large overall motions—The plane case: Part I, Trans. ASME, Vol.53, pp.849-854, 1986.

[49] J.C. Simo, L. Vu-Quoc, On the dynamics of flexible beams under large overall motions—The plane case: Part II, Trans. ASME, Vol.53, pp.855-863, 1986.

[50] M. Geradin, A. Cardona, D. Granville, Finite element modelling of flexible multibody systems, ACD&D'89 Conference, Sept6-8, 1989, Tokyo, Japan.

[51] A. Cardona, M. Geradin, Modeling of superelement in mechanism analysis, International Journal for Numerical Methods in Engineering, Vol.32, pp.1565-1593, 1991.

[52] A. Avello, J.G.D. Jalon, Dynamics of flexible multibody systems using Cartesian coordinates and large displacement theory, International Journal for Numerical Methods in Engineering, Vol.32, pp.1543-1563, 1991.

[53] R.K. Cavin III, R.A. Dusto, Hamilton's principle: finite element methods and flexible body dynamics, AIAA Journal, Vol.15, pp.1684-1690, 1977.

[54] O.P. Agrawal, A.A. Shabana, Application of deformable body mean axis to flexible multibody system dynamics, Computer Methods in Applied Mechanics and Engineering, Vol.56, pp.217-245, 1986.

[55] W.P. Koppens, A.A.H.J. Sauren, F.E. Veldpaus, D.H. van Campen, The dynamics of a deformable body experiencing large displacements, Trans. ASME, Vol.55, pp.676-680, 1988.

[56] C. Damaren, I. Sharf, Simulation of flexible link manipulators with inertial and geometric nonlinearities, Journal of Dynamic Systems, Measurement, and Control, Vol.117, pp.74-87, 1995.

[57] I. Sharf, Geometrical non-linear beam element for dynamics simulation of multibody systems, International Journal for Numerical Methods in Engineering, Vol.39, pp.763-786, 1996.

[58] I. Sharf, Geometric stiffening in multibody dynamics formulations, Journal of Guidance, Control, and Dynamics, Vol.18, pp.882-890, 1995.

[59] W. Schiehlen, Multibody systems handbook, Springer-Verlag Berlin, 1990.

[60] G.C. Andrews, The vector-network model: A topological approach to mechanics, PhD thesis, University of Waterloo, 1971.

[61] G.C. Andrews, H.K. Kesavan, The vector-network model: A new approach to vector dynamics, Mechanism and Machine Theory, Vol.10, pp.57-75, 1975.

[62] M.K. Ormrod, ADVENT: A vector-network computer program for simulating constrained mechanical systems, Master's Thesis, University of Waterloo, 1985.

[63] G.C. Andrews, M.J. Richard, R. Anderson, A general vector-network formulation for dynamic systems with kinematic constraints, Mechanism and Machine Theory, Vol.33, pp.243-256, 1988.

[64] J.J. McPhee, Automatic generation of motion equations for planar mechanical systems using the new set of "branch coordinates", Mechanism and Machine Theory, Vol.33, pp.805-823, 1998.

[65] J.J. McPhee, Three-dimensional dynamics of unconstrained rigid body systems: a vector-network formulation, Master's Thesis, University of Waterloo, 1986.

[66] M.J. Richard, M. Tennich, Dynamic simulation of flexible multibody systems using vector-network techniques, AMD-Vol.141/DSC-Vol.37, Dynamics of Flexible Multibody Systems: Theory and Experiments, ASME, pp.165-173, 1992.

[67] M. Tennich, An extension of the vector network method for the dynamic analysis of 3D flexible multibody systems, 1996 CSME proceedings, pp.156-162.

[68] S. Shin, W. Yoo, J. Tang, Effects of mode selection, scaling, and orthogonalization on the dynamic analysis of flexible multibody systems, Mech. Struct. & Mach., Vol.21, pp.507-527, 1993.

[69] T.W. Li, Dynamics of rigid body systems: a vector-network approach, Master's thesis, University of Waterloo.

[70] L.Lo, On the development of a kinematic and dynamic expert system, Master's thesis, University of Waterloo.

[71] E.P. Popov, Engineering mechanics of solids, Prentice-Hall, In.,1990.

[72] H.E. Koenig, Y. Tokad, H.K. Kesavan, Analysis of discrete physical systems, McGraw-Hill, Inc., 1967.

[73] G.C. Andrews, Dynamics using vector network techniques, course notes, University of Waterloo, Canada, 1977.

[74] M. Chandrashekar, G.J. Savage, Engineering systems: modelling, analysis and design, course notes, University of Waterloo, 1998.

[75] J.J. McPhee, On the use of linear graph theory in multibody system dynamics, J. Nonlinear Dynamics, Vol.9, pp.73-90, 1996.

[76] F.A. Firestone, The mobility method of computing the vibration of linear mechanical and acoustical systems: mechanical-electrical analogies, Journal of Applied Physics, Vol.9, pp.373-387, 1938.

[77] H.M. Trent, Isomorphisms between oriented linear graphs and lumped physical systems, The Journal of the Acoustical Society of America, Vol.27, pp.500-527, 1955.

[78] D.T. Greenwood, Principles of dynamics, Prentice-Hall, Inc., 1988, pp. 150-152.

[79] G.C. Andrews, Dynamics using vector-network techniques, University of Waterloo Lecture Notes, pp.78-84, 1977.

[80] N. Deo, Graph theory with applications to engineering and computer science, Prentice-Hall, Inc., pp.384-415, 1974.

[81] S. Wu, E.J. Haug, Geometric nonlinear substructuring for dynamics of flexible mechanical systems, International Journal of Numerical Method in Engineering, Vol.26, pp.2211-2226, 1988.

[82] G.J. Savage, H.K. Kesavan, The graph-theoretic field model—I. modelling and formulations, Journal of The Franklin Institute, Vol.307, pp.107-265, 1979.

[83] G.J. Savage, H.K. Kesavan, The graph-theoretic field model—II. the application of multi-terminal representation to field problems, Journal of The Franklin Institute, Vol.309, 1980.

[84] R.A. Wehage, Generalized coordinate partitioning in dynamic analysis of mechanical systems, Ph.D. Thesis, University of Iowa, Dec. 1980.

[85] The NAG Fortran library, Vol.5, The Numerical Algorithms Group Limited, 1991.

[86] J. Lieh, A alternative method for formulating close-form dynamics for elastic mechanical systems using symbolic process, Mechanisms, Structures and Machines, Vol.20, pp.253-279, 1992.

[87] J.-C. Piedboeuf, Symbolic manipulation of flexible manipulators, Paper AAS 95-357, AAS/AIAA Astrodynamics Specialist Conference, Halifax, Canada, 1995.

[88] R.E. Valembois, P. Fisette, J.C. Samin, Comparison of various techniques for modelling flexible beams in multibody dynamics, Nonlinear Dynamics, Vol.12, pp.367-397, 1997.

[89] J. Irving, N. Mullineux, Mathematics in physics and engineering, Academic Press Inc., 1959.

[90] G. Dhatt, G. Touzot, The finite element method displayed, Chap.3, Variational formulation of engineering problems, John Wiley & Sons, 1984.

[91] L. Meirovitch, M.K. Kwak, Convergence of the classical Rayleigh-Ritz method and the finite element method, AIAA Journal, Vol.28(8), pp.1509-1516, 1990.

[92] L. Meirovitch, M.K. Kwak, Rayleigh-Ritz based substructure synthesis for flexible multibody systems, AIAA Journal, Vol.29(10), pp.1709-1719, 1991.

[93] C.W. Jen, D.A. Johson, F. Dubois, Numerical modal analysis of structures based on a revised substructure synthesis approach, Journal of Sound and Vibration, Vol.180(2), pp.185-203, 1995.

[94] H.H. Hilber, T.J.R. Hughes, R.L.Taylor, Improved numerical dissipation for time integration algorithms in structural dynamics, Earthquake Engineering and Structural Dynamics, Vol.5, pp.283-292, 1977.

[95] U.M. Ascher, L.R. Petzold, Stability of computational methods for constrained dynamics systems, SIAM J. Sci. Comput., Vol.14, No.1, pp95-120, Jan., 1993.

[96] J.W. Kamman, R.L. Huston, Dynamics fo constrained multibody systems, Journal of Applied Mechanics, Vol.51, pp.899-903. 1984.

[97] W.C. Rheinboldt, Differential-algebraic systems as differential equations on manifolds, Math., Comput. Vol.43, pp.473-482, 1984.

[98] J.W. Starner, A numerical algorithm for the solution of implicit algebraic-differential systems of equations, Ph.D Thesis, University of New Mexico, 1976.

[99] E. Haug, J. Kuhl, F.-F. Tsai, Real-time dynamics for operator-in-the-loop simulation, Proceedings of the SIAM Regional Conference on Geometric Aspects of Industrial Design, pp.154-176, 1992

[100] C. Schaller, C. Führer, B. Simeon, Simulation of multibody systems by a parallel extrapolation method, Mech. Struct. & Mach., Vol.22, pp.473-486, 1994.

[101] I. Shames, C.L. Dym, Energy and finite element methods in structural mechanics, Chapter 3, Hemisphere Publishing Corporation, 1985.

[102] A.A. Shabana, Computational Dynamics, 1994, Chapter 4, John Wiley & Sons, Inc.

[103] W. Blajer, A projection method approach to constraint dynamic analysis, Journal of applied Mechanics, Vol.59, pp.643-649, 1992.

[104] K. Alkire, An analysis of rotor blade twist variables associated with different Euler sequences and pretwist treatments, NASA-TM-84394.

[105] D.H. Hodges, E.H. Dowell, Nonlinear equations of motion for the elastic bending and torsion of twisted nonuniform rotor blades, NASA-TN-D7818.

[106] D.H. Hodges, R.A. Ormiston, David A. Peters, On the nonlinear deformation geometry of Euler-Bernoulli beams, NASA Technical Report 80-A-1.

[107] P.F. Pai, A.H. Nayfeh, A fully nonlinear theory of curved and pretwisted composite rotor blades accounting for warping and three-dimensional stress effects, Int. J. Solids Structures, Vol.31, pp.1309-1340, 1994.

[108] S.K. Ider, F.M.L. Amirouche, Nonlinear modelling of flexible multibody systems dynamics subjected to variable constraints, Journal of Applied Mechanics, Vol.56, pp.444-450, 1989.

[109] G. Wempner, Mechanics of solids with applications to thin bodies, Chapters 2 and 9, Sijthoff & Noordhoff International Publishers, 1981.

[110] D. Redfern, The maple handbook, Springer-Verlag New York, Inc., pp.238-256, 1993.

[111] E.J. Haug, DADS theoretical manual, Allyn and Bacon, 1989.

[112] F. Amirouche, Computational methods in multibody dynamics, Prentice-Hall, Inc., 1992.

[113] P.E. Nikravesh, G. Gim, Systematic construction of the equations of motion for multibody systems containing closed kinematic loops, Journal of Mechanical Design, Vol.115, pp.143-149, 1993.

[114] B. Jonker, A finite element dynamic analysis of spatial mechanisms with flexible links, Computer Methods in Applied Mechanics and Engineering, Vol.79, pp.17-40, 1989.

[115] J. Ginsberg, Advanced Engineering Dynamics, 1995.

[116] P. Shi, J. McPhee, On the use of virtual work in a graph-theoretic formulation for multibody dynamics, 1997 ASME Design Engineering Technical Conference.

# Appendix A

## A.1 Derivation of $\dot{\mathbf{r}}_P$, $\delta\mathbf{r}_P$ and $\ddot{\mathbf{r}}_P$

From kinematics, one has

$$\underset{\sim}{\dot{r}}_P = \underset{\sim}{\dot{R}} + \underset{\sim}{\omega} \times \underset{\sim}{r} + \overset{\circ}{\underset{\sim}{r}} \tag{A.1}$$

which, in the absolute frame $X_a Y_a Z_a$, takes the following scalar form:

$$\dot{\mathbf{r}}_P = \dot{\mathbf{R}} - \mathbf{C}_{ab}\,\tilde{\mathbf{r}}^b \mathbf{C}_{ab}^T\,\omega + \mathbf{C}_{ab}\,\dot{\mathbf{r}}^b \tag{A.2}$$

where $\mathbf{C}_{ab} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is the rotation matrix from the body frame to the absolute frame (see Figure ??); $\tilde{\mathbf{r}}^b = \begin{bmatrix} 0 & 0 & r_y \\ 0 & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix}$ is the skew-symmetric matrix of the coordinates of vector $\vec{r}$ in body frame; $\omega = \begin{Bmatrix} 0 \\ 0 \\ \dot{\theta} \end{Bmatrix}$; $\dot{\mathbf{r}}^b = \begin{Bmatrix} \dot{r}_x \\ \dot{r}_y \\ 0 \end{Bmatrix}$.

From the assumed displacement field, we further know:

$$\mathbf{r}^b = \left\{ \begin{array}{c} r_x \\ r_y \\ 0 \end{array} \right\} = \left\{ \begin{array}{c} r_0 \\ x^2 q_1 + x^3 q_2 \\ 0 \end{array} \right\} = \left\{ \begin{array}{c} x \\ x^2 q_1 + x^3 q_2 \\ 0 \end{array} \right\} \tag{A.3}$$

$$\dot{\mathbf{r}}^b = \left\{ \begin{array}{c} \dot{r}_x \\ \dot{r}_y \\ 0 \end{array} \right\} = \left\{ \begin{array}{c} 0 \\ \dot{v} \\ 0 \end{array} \right\} = \left[ \begin{array}{cc} 0 & 0 \\ x^2 & x^3 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} \dot{q}_1 \\ \dot{q}_2 \end{array} \right\} \tag{A.4}$$

Substituting Eqs.(A.3), (A.4) into Eq.(A.2) gives

$$\begin{aligned} \dot{\mathbf{r}}_P &= \left\{ \begin{array}{c} \dot{R}_X \\ \dot{R}_Y \\ 0 \end{array} \right\} - \left\{ \begin{array}{c} r_y \cos\theta + r_x \sin\theta \\ r_y \sin\theta - r_x \cos\theta \\ 0 \end{array} \right\} \dot{\theta} + \mathbf{C}_{ab} \left[ \begin{array}{cc} 0 & 0 \\ x^2 & x^3 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} \dot{q}_1 \\ \dot{q}_2 \\ 0 \end{array} \right\} \\ &= \left[ \begin{array}{ccccc} 1 & 0 & -(x^2 q_1 + x^3 q^2)\cos\theta - x\sin\theta & -x^2\sin\theta & -x^3\sin\theta \\ 0 & 1 & -(x^2 q_1 + x^3 q_2)\sin\theta + x\cos\theta & x^2\cos\theta & x^3\cos\theta \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \left\{ \begin{array}{c} \dot{R}_X \\ \dot{R}_Y \\ \dot{\theta} \\ \dot{q}_1 \\ \dot{q}_2 \end{array} \right\} \\ &= \mathbf{L}_1 \dot{\mathbf{q}} \end{aligned} \tag{A.5}$$

therefore,

$$\delta \mathbf{r}_P = \mathbf{L}_1 \delta \mathbf{q} \tag{A.6}$$

and

$$\ddot{\mathbf{r}}_P = \frac{d\,\dot{\mathbf{r}}_P}{dt} \tag{A.7}$$

$$= \begin{bmatrix} 1 & 0 & -(x^2 q_1 + x^3 q_2)\cos\theta - x\sin\theta & -x^2\sin\theta & -x^3\sin\theta \\ 0 & 1 & -(x^2 q_1 + x^3 q_2)\sin\theta + x\cos\theta & x^2\cos\theta & x^3\cos\theta \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{R}_X \\ \ddot{R}_Y \\ \ddot{\theta} \\ \ddot{q}_1 \\ \ddot{q}_2 \end{Bmatrix}$$

$$+ \begin{Bmatrix} -\dot{\theta}^2[x\cos\theta - (x^2 q_1 + x^3 q_2)\sin\theta] \\ -\dot{\theta}^2[x\sin\theta + (x^2 q_1 + x^3 q_2)\cos\theta] \\ 0 \end{Bmatrix} + 2\begin{Bmatrix} -\dot{\theta}(x^2\dot{q}_1 + x^3\dot{q}_2)\cos\theta \\ -\dot{\theta}(x^2\dot{q}_1 + x^3\dot{q}_2)\sin\theta \\ 0 \end{Bmatrix} \tag{A.8}$$

$$= \mathbf{L}_1\ddot{\mathbf{q}} + \mathbf{L}_2$$

## A.2 Derivation of $\delta \mathbf{r}_{21}$

Referring to Table 3.2 and Figure 7.7, one has:

$$\begin{aligned} \delta \underset{\sim}{r}_{21} = \underset{\sim}{F}^b \delta \mathbf{r}^b_{21} &= \delta \underset{\sim}{u}(l) + \delta\theta \hat{k} \times \underset{\sim}{r}_2(l) \\ &= \delta u\, \hat{i}^b + \delta v\, \hat{j}^b + \delta\theta\, \hat{k}^b \times [l\, \hat{i}^b + v(l)\, \hat{j}^b] \\ &= (l^2\delta q_1 + l^3 q_2)\, \hat{j}^b + \delta\theta\, l\, \hat{j}^b + \delta\theta \hat{k}^b \times (l^2 q_1 + l^3 q_2)\hat{j}^b \\ &= [\hat{i}^b \;\; \hat{j}^b \;\; \hat{k}^b] \begin{Bmatrix} -\delta\theta(l^2 q_1 + l^3 q_2) \\ l\,\delta\theta + l^2\delta q_1 + l^3 q_2 \\ 0 \end{Bmatrix} \end{aligned} \tag{A.9}$$

which, in the inertial frame, takes the form:

$$
\begin{aligned}
\delta \mathbf{r}_{21} &= \mathbf{C}_{ab}\, \delta \mathbf{r}_2^b \\
&= \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} -\delta\theta(l^2 q_1 + l^3 q_2) \\ l\delta\theta + l^2 \delta q_1 + l^3 q_2 \\ 0 \end{Bmatrix} \\
&= \begin{bmatrix} -l\sin\theta - \cos\theta(l^2 q_1 + l^3 q_2) & -l^2 \sin\theta & -l^3 \sin\theta \\ l\cos\theta - \sin\theta(l^2 q_1 + l^3 q_2) & l^2 \cos\theta & l^3 \cos\theta \end{bmatrix} \begin{Bmatrix} \delta\theta \\ \delta q_1 \\ \delta q_2 \end{Bmatrix}
\end{aligned} \tag{A.10}
$$

## A.3 Maple Input File and Partial Output for Spinning Disk

```
> read'gins2.mpl'; read'gen_sys';
```

$$NOofedges := 13$$

$$NOofnodes := 8$$

$$Datum := 1$$

$$edge_1 := \text{table}([$$

$$4 = \text{table}([$$

$$inert = [[0,0,0],[0,0,0],[0,0,0]]$$

$$mass = 0$$

$$])$$

$$1 = N$$

$$2 = [1,3]$$

$$3 = BE\_R$$

$$])$$

$$edge_2 := \text{table}([$$

$$4 = \text{table}([$$

$$inert = [[0,0,0],[0,0,0],[0,0,0]]$$

$mass = 0$

])

$1 = N$

$2 = [1, 6]$

$3 = BE\_R$

])

$edge_3$ := table([

4 = table([

$inert = [[a, 0, 0], [0, a, 0], [0, 0, a]]$

$mass = a$

])

$1 = N$

$2 = [1, 10]$

$3 = BE\_R$

])

$edge_4$ := table([

4 = table([

$coords = [0, 0, a]$

])

$1 = Y$

$2 = [3, 2]$

$3 = AE\_R$

])

$edge_5$ := table([

4 = table([

$coords = [0, 0, a]$

$rotmat = [[0, 0, -1], [0, -1, 0], [-1, 0, 0]]$

])

```
1 = Y
2 = [3, 4]
3 = AE_R
])
                edge_10 := table([
                    4 = RV
                    1 = Y
                    2 = [4, 5]
                    3 = JE
                    ])
edge_7 := table([
    4 = table([
    coords = [0, 0, a]
    rotmat = [[0, 1, 0], [0, 0, 1], [1, 0, 0]]
    ])
    1 = Y
    2 = [6, 5]
    3 = AE_R
    ])
        edge_12 := table([
            4 = table([
            coords = [0, 0, a]
            ])
            1 = Y
            2 = [6, 9]
            3 = AE_R
            ])
            edge_6 := table([
```

$4 = RV$

$1 = Y$

$2 = [1, 2]$

$3 = JE$

])

$edge_{11}$ := table([

$4 = RV$

$1 = Y$

$2 = [9, 10]$

$3 = JE$

])

$edge_8$ := table([

$4$ = table([

$force = \theta$

$\theta = \psi(t)$

$type = JD$

])

$1 = N$

$2 = [1, 2]$

$3 = MDE$

])

$edge_{13}$ := table([

$4$ = table([

$force = \theta$

$\theta = \phi(t)$

$type = JD$

])

$1 = N$

$2 = [9, 10]$

$3 = MDE$

])

$edge_{15} :=$ table([

$4 =$ table([

$force = gl$

$type = PD$

$fz = -\mathrm{W}$

])

$1 = N$

$2 = [1, 10]$

$3 = FDE$

])

$Iedge := []$

========== *topo_matrix starts* ==========

```
Warning: new definition for   norm
Warning: new definition for   trace
```

$edges := [1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 15]$

$branch\_list := [4, 5, 6, 7, 10, 11, 12]$

$chord\_list := [1, 2, 3, 8, 13, 15]$

$node\_list := [1, 2, 3, 4, 5, 6, 9, 10]$

$VW\_list := []$

$Total\_NOofedges := 13$

$INCIDENCE :=$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & -1 & -1 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & -1 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

========== *path_lists starts* ==========

$$tree\_top\_nodes := \{10\}$$

$$path10 := [6, 4, 5, 10, 7, 12, 11]$$

$$direct10 := [-1, 1, -1, -1, 1, -1, -1]$$

========== *transmatrix starts* ==========

$$body\_set := [1, 2, 3]$$

$$Path1 := [6, 4]$$

$$direction1 := [-1, 1]$$

$$Path2 := [6, 4, 5, 10, 7]$$

$$direction2 := [-1, 1, -1, -1, 1]$$

$$Path3 := [6, 4, 5, 10, 7, 12, 11]$$

$$direction3 := [-1, 1, -1, -1, 1, -1, -1]$$

$$Tr1 := \begin{bmatrix} \cos(\theta 6(t)) & -\sin(\theta 6(t)) & 0 \\ \sin(\theta 6(t)) & \cos(\theta 6(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$Tr2 :=$

$[\sin(\theta 6(t))\cos(\theta 10(t)), -\cos(\theta 6(t)), \sin(\theta 6(t))\sin(\theta 10(t))]$

$[-\cos(\theta 6(t))\cos(\theta 10(t)), -\sin(\theta 6(t)), -\cos(\theta 6(t))\sin(\theta 10(t))$

$]$

$[\sin(\theta 10(t)), 0, -\cos(\theta 10(t))]$

$Tr3 :=$

$[\sin(\theta 6(t))\cos(\theta 10(t))\cos(\theta 11(t)) - \cos(\theta 6(t))\sin(\theta 11(t)),$

$-\sin(\theta 6(t))\cos(\theta 10(t))\sin(\theta 11(t)) - \cos(\theta 6(t))\cos(\theta 11(t)),$

$\sin(\theta 6(t))\sin(\theta 10(t))]$

$[-\cos(\theta 6(t))\cos(\theta 10(t))\cos(\theta 11(t)) - \sin(\theta 6(t))\sin(\theta 11(t)),$

$\cos(\theta 6(t))\cos(\theta 10(t))\sin(\theta 11(t)) - \sin(\theta 6(t))\cos(\theta 11(t)),$

$-\cos(\theta 6(t))\sin(\theta 10(t))]$

$[\sin(\theta 10(t))\cos(\theta 11(t)), -\sin(\theta 10(t))\sin(\theta 11(t)),$

$-\cos(\theta 10(t))]$

## A.4 Maple Input File and Partial Output for Rigid Spatial Slider Crank

```
> read'D_splider1'; read'gen_sys';
```

$$NOofedges := 17$$

$$NOofnodes := 10$$

$$Datum := 5$$

$$edge_1 := \text{table}([$$

$$1 = Y$$

$$2 = [5, 1]$$

$$3 = AE\_S$$

$$4 = \text{table}([$$

$$coords = [0, 0, -1, 0, 0, 0]$$

$$])$$

$$])$$

$$edge_2 := \text{table}([$$

$$1 = N$$

$$2 = [5, 11]$$

$$3 = BE\_R$$

$$4 = \text{table}([$$

$$inert = [[a, 0, 0], [0, b, 0], [0, 0, c]]$$

$$mass = aa$$

$$])$$

$$])$$

$$edge_3 := \text{table}([$$

$$1 = N$$

$$2 = [5, 4]$$

$$3 = BE\_R$$

```
4 = table([
inert = [[a, 0, 0], [0, b, 0], [0, 0, c]]
mass = aa
])
])
        edge_4 := table([
            1 = Y
            2 = [11, 1]
            3 = JE
            4 = WELD
            ])
        edge_5 := table([
            1 = Y
            2 = [11, 3]
            3 = JE
            4 = CR
            5 = [0, a, 0]
            ])
    edge_6 := table([
        1 = Y
        2 = [4, 3]
        3 = AE_R
        4 = table([
        coords = [0, 0, a]
        ])
        ])
    edge_7 := table([
        1 = Y
```

```
        2 = [4, 9]
        3 = AE_R
        4 = table([
        coords = [0, 0, s]
        ])
        ])
        edge_8 := table([
            1 = N
            2 = [8, 9]
            3 = JE
            4 = SPH
            ])
    edge_9 := table([
        1 = Y
        2 = [7, 6]
        3 = AE_R
        4 = table([
        coords = [a, 0, 0]
        ])
        ])
edge_10 := table([
    1 = N
    2 = [5, 7]
    3 = BE_R
    4 = table([
    inert = [[a, 0, 0], [0, b, 0], [0, 0, c]]
    mass = aa
    ])
```

```
])
    edge_11 := table([
        1 = Y
        2 = [7, 8]
        3 = AE_R
        4 = table([
        coords = [a, 0, 0]
        ])
        ])
        edge_12 := table([
            1 = Y
            2 = [13, 6]
            3 = JE
            4 = RV
            ])
        edge_13 := table([
            1 = N
            2 = [13, 6]
            3 = MDE
            4 = table([
            θ = 2πt
            force = θ
            type = JD
            ])
            ])
edge_16 := table([
    1 = Y
    2 = [5, 13]
```

$3 = AE\_R$

$4 = \text{table}([$

$rotmat = [[0, -1, 0], [1, 0, 0], [0, 0, 1]]$

$coords = [a, a, 0]$

$])$

$])$

$edge_{20} := \text{table}([$

$1 = N$

$2 = [5, 11]$

$3 = FDE$

$4 = \text{table}([$

$fy = -m2\,grav$

$force = gl$

$type = PD$

$])$

$])$

$edge_{21} := \text{table}([$

$1 = N$

$2 = [5, 4]$

$3 = FDE$

$4 = \text{table}([$

$fy = -m3\,grav$

$force = gl$

$type = PD$

$])$

$])$

$edge_{22} := \text{table}([$

$1 = N$

$2 = [5, 7]$

$3 = FDE$

$4 = \text{table}([$

$fy = -m10\,grav$

$force = gl$

$type = PD$

$])$

$])$

$Iedge := [12]$

========== *reading my_pack1* ==========

========== *reading my_pack2* ==========

========== *reading my_pack3* ==========

========== *topo_matrix starts* ==========

```
Warning: new definition for   norm
Warning: new definition for   trace
```

$$edges := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 20, 21, 22]$$

$$branch\_list := [1, 4, 5, 6, 7, 9, 11, 12, 16]$$

$$chord\_list := [2, 3, 8, 10, 13, 20, 21, 22]$$

$$node\_list := [1, 3, 4, 5, 6, 7, 8, 9, 11, 13]$$

$$VW\_list := []$$

$$Total\_NOofedges := 17$$

$$INCIDENCE := \begin{bmatrix} -1,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 \\ 0,0,-1,-1,0,0,0,0,0,0,0,0,0,0,0,0,0 \\ 0,0,0,1,1,0,0,0,0,0,-1,0,0,0,0,-1,0 \\ 1,0,0,0,0,0,0,0,1,1,1,0,1,0,1,1,1 \\ 0,0,0,0,0,-1,0,-1,0,0,0,0,0,-1,0,0,0 \\ 0,0,0,0,0,1,1,0,0,0,0,0,0,-1,0,0,0,-1 \\ 0,0,0,0,0,0,-1,0,0,0,0,1,0,0,0,0,0 \\ 0,0,0,0,-1,0,0,0,0,0,0,-1,0,0,0,0,0 \\ 0,1,1,0,0,0,0,0,0,-1,0,0,0,0,-1,0,0 \\ 0,0,0,0,0,0,0,1,-1,0,0,0,0,1,0,0,0 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1,0,0,0,0,0,0,0,0,0,1,1,1,0,0,1,1,0 \\ 0,1,0,0,0,0,0,0,0,0,-1,-1,-1,0,0,-1,-1,0 \\ 0,0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0 \\ 0,0,0,1,0,0,0,0,0,0,0,-1,-1,0,0,0,-1,0 \\ 0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0 \\ 0,0,0,0,0,1,0,0,0,0,0,0,1,-1,0,0,0,-1 \\ 0,0,0,0,0,0,1,0,0,0,0,-1,0,0,0,0,0 \\ 0,0,0,0,0,0,0,1,0,0,0,-1,1,1,0,0,1 \\ 0,0,0,0,0,0,0,0,1,0,0,-1,1,0,0,0,1 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} -1,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0 \\ -1,1,-1,1,0,0,0,0,0,0,1,0,0,0,0,0,0 \\ -1,1,-1,1,-1,-1,1,1,1,0,0,1,0,0,0,0,0 \\ 0,0,0,0,0,1,0,-1,-1,0,0,0,1,0,0,0,0 \\ 0,0,0,0,0,0,0,-1,0,0,0,0,0,1,0,0,0 \\ -1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0 \\ -1,1,-1,1,0,0,0,0,0,0,0,0,0,0,0,1,0 \\ 0,0,0,0,0,1,0,-1,-1,0,0,0,0,0,0,0,1 \end{bmatrix}$$

========== *path_lists starts* ==========

$$tree\_top\_nodes := \{8,9\}$$

$$path8 := [16,12,9,11]$$

$$direct8 := [-1,-1,1,-1]$$

$$path9 := [1,4,5,6,7]$$

$$direct9 := [-1,1,-1,1,-1]$$

========== *transmatrix starts* ==========

$$body\_set := [2,3,10]$$

$$Path2 := [1,4]$$

$$direction2 := [-1,1]$$

$$Path3 := [1,4,5,6]$$

$$direction3 := [-1,1,-1,1]$$

$$Path10 := [16,12,9]$$

$$direction10 := [-1,-1,1]$$

$$Tr2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$Tr3 :=$

$$[\cos(\alpha 5(t)), -\sin(\alpha 5(t))\cos(\beta 5(t)), \sin(\alpha 5(t))\sin(\beta 5(t))]$$

$$[\sin(\alpha 5(t)), \cos(\alpha 5(t))\cos(\beta 5(t)), -\cos(\alpha 5(t))\sin(\beta 5(t))]$$

$$[0, \sin(\beta 5(t)), \cos(\beta 5(t))]$$

$$Tr10 := \begin{bmatrix} -\sin(\theta 12(t)) & -\cos(\theta 12(t)) & 0 \\ \cos(\theta 12(t)) & -\sin(\theta 12(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

---

## A.5 Maple Input File and Full Output of System Equations for Planar Spin-Up Beam

```
> restart; read'beam.mpl'; read'gen_sys';
```

$$NOofedges := 3$$

$$NOofnodes := 2$$

$$Datum := 2$$

$$edge_1 := \text{table}([$$

$$1 = Y$$

$$2 = [2, 1]$$

$$3 = JE$$

$$4 = RV$$

$$])$$

$$edge_2 := \text{table}([$$

$$1 = N$$

$$2 = [2, 1]$$

$$3 = BE\_F$$

$$4 = [2, 2, 0, 0]$$

$$Order\_Finert = 1$$

$$scheme = Taylor$$

$$Order\_Felastic = 1$$

$$])$$

$$edge_6 := \text{table}([$$

$$1 = N$$

$$2 = [2, 1]$$

$$3 = MDE$$

$$4 = \text{table}([$$

$$type = JD$$

$$\theta = \theta(t)$$

$$force = \theta$$

$$])$$

$$])$$

$$Iedge := []$$

$$==========\ reading\ my\_pack1\ ==========$$

$$==========\ reading\ my\_pack2\ ==========$$

$$==========\ reading\ my\_pack3\ ==========$$

$$==========\ topo\_matrix\ starts\ ==========$$

```
Warning: new definition for   norm
Warning: new definition for   trace
```

$$edges := [1, 2, 6]$$

$$branch\_list := [1]$$

$$chord\_list := [2, 6]$$

$$node\_list := [1, 2]$$

$$VW\_list := []$$

$$Total\_NOofedges := 3$$

$$INCIDENCE := \begin{bmatrix} -1 & -1 & -1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{========== } path\_lists \text{ starts } \text{==========}$$

$$tree\_top\_nodes := \{1\}$$

$$path1 := [1]$$

$$direct1 := [-1]$$

$$\text{========== } transmatrix \text{ starts } \text{==========}$$

$$body\_set := [2]$$

$$Path2 := [1]$$

$$direction2 := [-1]$$

$$Tr2 := \begin{bmatrix} \cos(\theta 1(t)) & -\sin(\theta 1(t)) & 0 \\ \sin(\theta 1(t)) & \cos(\theta 1(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{========== } angues \text{ starts } \text{==========}$$

$$V\_ang2 := \left[00\frac{\partial}{\partial t}\theta 1(t)\right]$$

$$W\_ang2 := \left[00\frac{\partial^2}{\partial t^2}\theta 1(t)\right]$$

$$\text{========== } kinemquatsstarts \text{ ==========}$$

$$Tree\_DOFs := 5$$

$$CJ\_eq := \begin{bmatrix} 0 \end{bmatrix}$$

$$JD\_eq := [\theta 1(t) = \theta(t)]$$

$$PD\_eq := \begin{bmatrix} 0 \end{bmatrix}$$

$$DD\_eq := \begin{bmatrix} 0 \end{bmatrix}$$

*A dynamics analysis is required for the system*

$$kin := \begin{bmatrix} 0 \end{bmatrix}$$

$$\text{========== } gen\_dyneqns \text{ starts } \text{==========}$$

$$Tree\_var\_vel := \left[\frac{\partial}{\partial t}\theta 1(t), \frac{\partial}{\partial t}u2_1(t), \frac{\partial}{\partial t}u2_2(t), \frac{\partial}{\partial t}v2_1(t), \frac{\partial}{\partial t}v2_2(t)\right]$$

$$Uvw := [u2_1(t), u2_2(t), v2_1(t), v2_2(t)]$$

$$Tvartable := \text{table}([$$

$$\frac{\partial}{\partial t}u2_2(t) = \chi^3$$

$$\frac{\partial}{\partial t}v2_1(t) = \chi^4$$

$$\frac{\partial}{\partial t} v2_2(t) = \chi^5$$
$$\frac{\partial}{\partial t} u2_1(t) = \chi^2$$
$$\frac{\partial}{\partial t} \theta 1(t) = \chi$$
$$])$$
$$Bjoint := [1]$$
$$Into\ VW\_Fbody, 2$$
$$Deform\_Coords\ for\ body, 2, \left[x u2_1(t) + x^2 u2_2(t) x^2 v2_1(t) + x^3 v2_2(t) 00\right]$$
$$VW\ of\ Elastic\ Forces\ Completed$$
$$Start\ Calculating\ VW\ of\ Rotational\ Inertial\ forces$$
$$End\ calculating\ VW\ of\ Rotational\ Inertial\ forces$$
$$Calculating\ VW\ of\ Translational\ Inertial\ forces$$
$$into\ Fbody[Vc]$$
$$out\ of\ Fbody[Vc]$$
$$VW\ of\ Translational\ Inertia\ Forces\ Completed$$
$$Start\ Calculating\ VW\ of\ Gravity$$
$$End\ Calculating\ VW\ of\ Gravity$$
$$Out\ of\ \ VW\_Fbody, 2$$
$$Independent\_var := \left[\frac{\partial}{\partial t}\theta 1(t), \frac{\partial}{\partial t} u2_1(t), \frac{\partial}{\partial t} u2_2(t), \frac{\partial}{\partial t} v2_1(t), \frac{\partial}{\partial t} v2_2(t)\right]$$
$$Start\ extracting\ mass\ matrix\ of\ kinetic\ equations$$
$$Acc\_col := \begin{bmatrix} \frac{\partial^2}{\partial t^2}\theta 1(t) \\ \frac{\partial^2}{\partial t^2} u2_1(t) \\ \frac{\partial^2}{\partial t^2} u2_2(t) \\ \frac{\partial^2}{\partial t^2} v2_1(t) \\ \frac{\partial^2}{\partial t^2} v2_2(t) \end{bmatrix}$$

$$M\_trix :=$$
$$\left[\frac{1}{2} Area2\, Ro2\, bL2^4 u2_2(t) + \frac{2}{3} Area2\, Ro2\, bL2^3 u2_1(t) + \frac{1}{3} Area2\, Ro2\, bL2^3, 0, 0\right.$$
$$\left., \frac{1}{4} Area2\, Ro2\, bL2^4, \frac{1}{5} Area2\, Ro2\, bL2^5\right]$$
$$\left[-\frac{1}{5} Area2\, Ro2\, bL2^5 v2_2(t) - \frac{1}{4} Area2\, Ro2\, bL2^4 v2_1(t), \frac{1}{3} Area2\, Ro2\, bL2^3,\right.$$

$$\frac{1}{4}Area2\,Ro2\,bL2^4, 0, 0\Big]$$

$$\Big[-\frac{1}{6}Area2\,Ro2\,bL2^6 v2_2(t) - \frac{1}{5}Area2\,Ro2\,bL2^5 v2_1(t), \frac{1}{4}Area2\,Ro2\,bL2^4,$$

$$\frac{1}{5}Area2\,Ro2\,bL2^5, 0, 0\Big]$$

$$\Big[\frac{1}{5}Area2\,Ro2\,bL2^5 u2_2(t) + \frac{1}{4}Area2\,Ro2\,bL2^4 u2_1(t) + \frac{1}{4}Area2\,Ro2\,bL2^4, 0, 0$$

$$, \frac{1}{5}Area2\,Ro2\,bL2^5, \frac{1}{6}Area2\,Ro2\,bL2^6\Big]$$

$$\Big[\frac{1}{5}Area2\,Ro2\,bL2^5 u2_1(t) + \frac{1}{5}Area2\,Ro2\,bL2^5 + \frac{1}{6}Area2\,Ro2\,bL2^6 u2_2(t), 0, 0$$

$$, \frac{1}{6}Area2\,Ro2\,bL2^6, \frac{1}{7}Area2\,Ro2\,bL2^7\Big]$$

*End extracting mass matrix of kinetic equations*

*Start extracting force column of kinetic equations*

*End extracting force column of kinetic equations*

*Form of System Kinetic Equations :*

*M_trix * Acc_col + F_col = 0*

*System Kinetic Equations Version 2*

$dyn\_eq2 :=$

$$\Big[\Big(\frac{1}{2}Area2\,Ro2\,bL2^4 u2_2(t) + \frac{2}{3}Area2\,Ro2\,bL2^3 u2_1(t) + \frac{1}{3}Area2\,Ro2\,bL2^3\Big)\,\%1$$

$$+ \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial^2}{\partial t^2}v2_1(t)\right) + \frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial^2}{\partial t^2}v2_2(t)\right)$$

$$+ \frac{2}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right) - \Gamma 6(t)$$

$$+ \frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)\Big]$$

$$\Big[\Big(-\frac{1}{5}Area2\,Ro2\,bL2^5 v2_2(t) - \frac{1}{4}Area2\,Ro2\,bL2^4 v2_1(t)\Big)\,\%1$$

$$+ \frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial^2}{\partial t^2}u2_1(t)\right) + \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial^2}{\partial t^2}u2_2(t)\right)$$

$$- \frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 - \frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 u2_1(t)$$

$$+ E2\,Area2\,u2_2(t)\,bL2^2 - \frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}v2_1(t)\right)$$

$$
\begin{aligned}
&-\frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 u2_2(t) + E2\,Area2\,u2_1(t)\,bL2 \\
&-\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}v2_2(t)\right)\Big] \\
&\Big[\left(-\frac{1}{6}Area2\,Ro2\,bL2^6 v2_2(t) - \frac{1}{5}Area2\,Ro2\,bL2^5 v2_1(t)\right)\%1 \\
&+\frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial^2}{\partial t^2}u2_1(t)\right) + \frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial^2}{\partial t^2}u2_2(t)\right) \\
&-\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}v2_1(t)\right) \\
&-\frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 u2_2(t) - \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 \\
&-\frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 u2_1(t) + \frac{4}{3}E2\,Area2\,u2_2(t)\,bL2^3 \\
&+E2\,Area2\,u2_1(t)\,bL2^2 - \frac{1}{3}Area2\,Ro2\,bL2^6\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}v2_2(t)\right)\Big] \\
&\Big[\left(\frac{1}{5}Area2\,Ro2\,bL2^5 u2_2(t) + \frac{1}{4}Area2\,Ro2\,bL2^4 u2_1(t) + \frac{1}{4}Area2\,Ro2\,bL2^4\right)\%1 \\
&+\frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial^2}{\partial t^2}v2_1(t)\right) + \frac{1}{6}Area2\,Ro2\,bL2^6\left(\frac{\partial^2}{\partial t^2}v2_2(t)\right) \\
&+\frac{1}{15}Area2\,Ro2\,bL2^5 v2_1(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 \\
&+\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right) \\
&+\frac{1}{12}Area2\,Ro2\,bL2^6 v2_2(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 + 6E2\,dI2z\,v2_2(t)\,bL2^2 \\
&+4E2\,dI2z\,v2_1(t)\,bL2 + \frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right)\Big] \\
&\Big[\left(\frac{1}{5}Area2\,Ro2\,bL2^5 u2_1(t) + \frac{1}{5}Area2\,Ro2\,bL2^5 + \frac{1}{6}Area2\,Ro2\,bL2^6 u2_2(t)\right)\%1 \\
&+\frac{1}{6}Area2\,Ro2\,bL2^6\left(\frac{\partial^2}{\partial t^2}v2_1(t)\right) + \frac{1}{7}Area2\,Ro2\,bL2^7\left(\frac{\partial^2}{\partial t^2}v2_2(t)\right) \\
&+12E2\,dI2z\,v2_2(t)\,bL2^3 + \frac{4}{35}Area2\,Ro2\,bL2^7 v2_2(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 \\
&+\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right) + 6E2\,dI2z\,v2_1(t)\,bL2^2 \\
&+\frac{1}{3}Area2\,Ro2\,bL2^6\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)
\end{aligned}
$$

$$+\frac{1}{12}Area2\,Ro2\,bL2^6\,v2_1(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2\Big]$$

$$\%1 := \frac{\partial^2}{\partial t^2}\theta1(t)$$

$$kin\_eq := \left[\ \theta1(t) - \theta(t)\ \right]$$

$$sys\_index2\_DAEs =,$$

$$\Big[\left(\frac{1}{2}Area2\,Ro2\,bL2^4\,u2_2(t) + \frac{2}{3}Area2\,Ro2\,bL2^3\,u2_1(t) + \frac{1}{3}Area2\,Ro2\,bL2^3\right)\%1$$
$$+\frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial^2}{\partial t^2}v2_1(t)\right) + \frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial^2}{\partial t^2}v2_2(t)\right)$$
$$+\frac{2}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right) - \Gamma6(t)$$
$$+\frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)\Big]$$

$$\Big[\left(-\frac{1}{5}Area2\,Ro2\,bL2^5\,v2_2(t) - \frac{1}{4}Area2\,Ro2\,bL2^4\,v2_1(t)\right)\%1$$
$$+\frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial^2}{\partial t^2}u2_1(t)\right) + \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial^2}{\partial t^2}u2_2(t)\right)$$
$$-\frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 - \frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 u2_1(t)$$
$$+E2\,Area2\,u2_2(t)\,bL2^2 - \frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}v2_1(t)\right)$$
$$-\frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 u2_2(t) + E2\,Area2\,u2_1(t)\,bL2$$
$$-\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}v2_2(t)\right)\Big]$$

$$\Big[\left(-\frac{1}{6}Area2\,Ro2\,bL2^6\,v2_2(t) - \frac{1}{5}Area2\,Ro2\,bL2^5\,v2_1(t)\right)\%1$$
$$+\frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial^2}{\partial t^2}u2_1(t)\right) + \frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial^2}{\partial t^2}u2_2(t)\right)$$
$$-\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}v2_1(t)\right)$$
$$-\frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 u2_2(t) - \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)^2$$
$$-\frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)^2 u2_1(t) + \frac{4}{3}E2\,Area2\,u2_2(t)\,bL2^3$$
$$+E2\,Area2\,u2_1(t)\,bL2^2 - \frac{1}{3}Area2\,Ro2\,bL2^6\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}v2_2(t)\right)\Big]$$

$$\left[\left(\frac{1}{5}Area2\,Ro2\,bL2^5\,u2_2(t)+\frac{1}{4}Area2\,Ro2\,bL2^4\,u2_1(t)+\frac{1}{4}Area2\,Ro2\,bL2^4\right)\%1\right.$$
$$+\frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial^2}{\partial t^2}v2_1(t)\right)+\frac{1}{6}Area2\,Ro2\,bL2^6\left(\frac{\partial^2}{\partial t^2}v2_2(t)\right)$$
$$+\frac{1}{15}Area2\,Ro2\,bL2^5\,v2_1(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2$$
$$+\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)$$
$$+\frac{1}{12}Area2\,Ro2\,bL2^6\,v2_2(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2+6E2\,dI2z\,v2_2(t)\,bL2^2$$
$$\left.+4E2\,dI2z\,v2_1(t)\,bL2+\frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right)\right]$$
$$\left[\left(\frac{1}{5}Area2\,Ro2\,bL2^5\,u2_1(t)+\frac{1}{5}Area2\,Ro2\,bL2^5+\frac{1}{6}Area2\,Ro2\,bL2^6\,u2_2(t)\right)\%1\right.$$
$$+\frac{1}{6}Area2\,Ro2\,bL2^6\left(\frac{\partial^2}{\partial t^2}v2_1(t)\right)+\frac{1}{7}Area2\,Ro2\,bL2^7\left(\frac{\partial^2}{\partial t^2}v2_2(t)\right)$$
$$+12E2\,dI2z\,v2_2(t)\,bL2^3+\frac{4}{35}Area2\,Ro2\,bL2^7\,v2_2(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2$$
$$+\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right)+6E2\,dI2z\,v2_1(t)\,bL2^2$$
$$+\frac{1}{3}Area2\,Ro2\,bL2^6\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)$$
$$\left.+\frac{1}{12}Area2\,Ro2\,bL2^6\,v2_1(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2\right]$$
$$[\theta1(t)-\theta(t)]$$
$$\%1:=\frac{\partial^2}{\partial t^2}\theta1(t)$$
$$sys\_MASS=,$$
$$\left[\frac{1}{2}Area2\,Ro2\,bL2^4\,u2_2(t)+\frac{2}{3}Area2\,Ro2\,bL2^3\,u2_1(t)+\frac{1}{3}Area2\,Ro2\,bL2^3,0,0\right.$$
$$\left.,\frac{1}{4}Area2\,Ro2\,bL2^4,\frac{1}{5}Area2\,Ro2\,bL2^5\right]$$
$$\left[-\frac{1}{5}Area2\,Ro2\,bL2^5\,v2_2(t)-\frac{1}{4}Area2\,Ro2\,bL2^4\,v2_1(t),\frac{1}{3}Area2\,Ro2\,bL2^3,\right.$$
$$\left.\frac{1}{4}Area2\,Ro2\,bL2^4,0,0\right]$$
$$\left[-\frac{1}{6}Area2\,Ro2\,bL2^6\,v2_2(t)-\frac{1}{5}Area2\,Ro2\,bL2^5\,v2_1(t),\frac{1}{4}Area2\,Ro2\,bL2^4,\right.$$

$$\frac{1}{5}Area2\,Ro2\,bL2^5, 0, 0\Big]$$

$$\Big[\frac{1}{5}Area2\,Ro2\,bL2^5\,u2_2(t) + \frac{1}{4}Area2\,Ro2\,bL2^4\,u2_1(t) + \frac{1}{4}Area2\,Ro2\,bL2^4, 0, 0$$

$$, \frac{1}{5}Area2\,Ro2\,bL2^5, \frac{1}{6}Area2\,Ro2\,bL2^6\Big]$$

$$\Big[\frac{1}{5}Area2\,Ro2\,bL2^5\,u2_1(t) + \frac{1}{5}Area2\,Ro2\,bL2^5 + \frac{1}{6}Area2\,Ro2\,bL2^6\,u2_2(t), 0, 0$$

$$, \frac{1}{6}Area2\,Ro2\,bL2^6, \frac{1}{7}Area2\,Ro2\,bL2^7\Big]$$

$$[1, 0, 0, 0, 0]$$

$$sys\_FORCE =,$$

$$\Big[\frac{2}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right) - \Gamma 6(t)$$

$$+ \frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)\Big]$$

$$\Big[-\frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 - \frac{1}{3}Area2\,Ro2\,bL2^3\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 u2_1(t)$$

$$+ E2\,Area2\,u2_2(t)\,bL2^2 - \frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}v2_1(t)\right)$$

$$- \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 u2_2(t) + E2\,Area2\,u2_1(t)\,bL2$$

$$- \frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}v2_2(t)\right)\Big]$$

$$\Big[-\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}v2_1(t)\right)$$

$$- \frac{1}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 u2_2(t) - \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2$$

$$- \frac{1}{4}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 u2_1(t) + \frac{4}{3}E2\,Area2\,u2_2(t)\,bL2^3$$

$$+ E2\,Area2\,u2_1(t)\,bL2^2 - \frac{1}{3}Area2\,Ro2\,bL2^6\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}v2_2(t)\right)\Big]$$

$$\Big[\frac{1}{15}Area2\,Ro2\,bL2^5\,v2_1(t)\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2$$

$$+ \frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta 1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)$$

$$+ \frac{1}{12}Area2\,Ro2\,bL2^6\,v2_2(t)\left(\frac{\partial}{\partial t}\theta 1(t)\right)^2 + 6\,E2\,dI2z\,v2_2(t)\,bL2^2$$

$$+4E2\,dI2z\,v2_1(t)\,bL2+\frac{1}{2}Area2\,Ro2\,bL2^4\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right)\Bigg]$$

$$\Bigg[12E2\,dI2z\,v2_2(t)\,bL2^3+\frac{4}{35}Area2\,Ro2\,bL2^7\,v2_2(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2$$

$$+\frac{2}{5}Area2\,Ro2\,bL2^5\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_1(t)\right)+6E2\,dI2z\,v2_1(t)\,bL2^2$$

$$+\frac{1}{3}Area2\,Ro2\,bL2^6\left(\frac{\partial}{\partial t}\theta1(t)\right)\left(\frac{\partial}{\partial t}u2_2(t)\right)$$

$$+\frac{1}{12}Area2\,Ro2\,bL2^6\,v2_1(t)\left(\frac{\partial}{\partial t}\theta1(t)\right)^2\Bigg]$$

$$\left[-\left(\frac{\partial^2}{\partial t^2}\theta(t)\right)\right]$$

*System ODEs in Second Order Derivatives :*

$$sys\_MASS * Acc\_col + sys\_FORCE = 0$$

## A.6 Maple Input File and Partial Output for Three-Dimensional Spin-Up Beam with Off-Set Tip Mass

```
> restart; read'beam1.mpl'; read'gen_sys';
```

$$NOofedges := 8$$

$$NOofnodes := 5$$

$$Datum := 2$$

$$edge_1 := \text{table}([$$

$$1 = Y$$

$$2 = [2, 1]$$

$$3 = JE$$

$$4 = RV$$

$$])$$

$$edge_4 := \text{table}([$$

$$1 = Y$$

$$2 = [1, 4]$$

$$3 = AE\_F$$

```
                4 = table([
                coords = [a, 0, 0]
                ])
                ])
        edge_2 := table([
            1 = N
            2 = [2, 1]
            3 = BE_F
            4 = [2, 2, 2, 2]
            scheme = Taylor
            Order_Felastic = 1
            Order_Finert = 1
            ])
edge_3 := table([
    1 = N
    2 = [2, 5]
    3 = BE_R
    4 = table([
    inert = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]
    mass = a
    ])
    ])
            edge_5 := table([
                1 = Y
                2 = [4, 3]
                3 = JE
                4 = WELD
                ])
```

$edge_6$ := table([

1 = $N$

2 = [2, 1]

3 = $MDE$

4 = table([

$\theta = \theta(t)$

$force = \theta$

$type = JD$

])

])

$edge_7$ := table([

1 = $N$

2 = [2, 5]

3 = $FDE$

4 = table([

$force = gl$

$type = PD$

$fz = -m3\,Grav$

])

])

$edge_8$ := table([

1 = $Y$

2 = [5, 3]

3 = $AE\_R$

4 = table([

$coords = [0, a, 0]$

])

])

$$Iedge := []$$

========== *reading my_pack1* ==========

========== *reading my_pack2* ==========

========== *reading my_pack3* ==========

========== *topo_matrix starts* ==========

```
Warning: new definition for   norm
Warning: new definition for   trace
```

$$edges := [1, 2, 3, 4, 5, 6, 7, 8]$$

$$branch\_list := [1, 4, 5, 8]$$

$$chord\_list := [2, 3, 6, 7]$$

$$node\_list := [1, 2, 3, 4, 5]$$

$$VW\_list := []$$

$$Total\_NOofedges := 8$$

$$INCIDENCE := \begin{bmatrix} -1 & 1 & 0 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & -1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$==========\ path\_lists\ starts\ ==========$$

$$tree\_top\_nodes := \{5\}$$

$$path5 := [1, 4, 5, 8]$$

$$direct5 := [-1, -1, -1, 1]$$

---

## A.7 Maple Input File and Partial Output for Two-Link Manipulator

```
> restart; read'ami.mpl'; read'gen_sys';
```

$$NOofedges := 7$$

$$NOofnodes := 4$$

$$Datum := 5$$

$$edge_1 := \text{table}([$$

$$1 = Y$$

$$2 = [5, 2]$$

$$3 = JE$$

$$4 = RV$$

$$])$$

$$edge_2 := \text{table}([$$

$$1 = Y$$

$$2 = [2, 3]$$

$$3 = AE\_F$$

$$4 = \text{table}([$$

$$coords = [x, 0, 0]$$

$$])$$

$$])$$

$$edge_4 := \text{table}([$$

$$1 = Y$$

$$2 = [3, 4]$$

$3 = JE$

$4 = RV$

])

$edge_6$ := table([

$1 = N$

$2 = [5, 2]$

$3 = BE\_F$

$4 = [1, 2, 0, 0]$

$Order\_Felastic = 1$

$scheme = Taylor$

$Order\_Finert = 1$

])

$edge_7$ := table([

$1 = N$

$2 = [5, 4]$

$3 = BE\_F$

$4 = [1, 2, 0, 0]$

$Order\_Felastic = 1$

$scheme = Taylor$

$Order\_Finert = 1$

])

$edge_8$ := table([

$1 = N$

$2 = [5, 2]$

$3 = FDE$

$4$ = table([

$type = JD$

$\theta = (0, 0, \Gamma 8)$

])

])

$edge_9$ := table([

1 = $N$

2 = [3, 4]

3 = $FDE$

4 = table([

$type = JD$

$\theta$ = (0, 0, $Torque9$)

])

])

$Iedge$ := []

========== *reading my_pack1* ==========

========== *reading my_pack2* ==========

========== *reading my_pack3* ==========

========== *topo_matrix starts* ==========

```
Warning: new definition for   norm
Warning: new definition for   trace
```

$edges$ := [1, 2, 4, 6, 7, 8, 9]

$branch\_list$ := [1, 2, 4]

$chord\_list$ := [6, 7, 8, 9]

$node\_list$ := [2, 3, 4, 5]

$VW\_list$ := []

$Total\_NOofedges$ := 7

$$INCIDENCE := \begin{bmatrix} -1 & 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & 0 & -1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$========== path\_lists\ starts ==========$$

$$tree\_top\_nodes := \{4\}$$

$$path4 := [1, 2, 4]$$

$$direct4 := [-1, -1, -1]$$

---

## A.8 Maple Input File and Partial Output for Planar Slider Crank with Flexible Link

```
> restart; read'F_slider1.mpl'; read'gen_sys';
```

$$NOofedges := 12$$

$$NOofnodes := 8$$

$$Datum := 5$$

$$edge_1 := \text{table}([$$

$$1 = Y$$

$$2 = [5, 1]$$

$$3 = AE\_S$$

$$4 = \text{table}([$$

```
  rotmat = [[-1, 0, 0], [0, 1, 0], [0, 0, -1]]
  coords = [1, 0, 0, 0, 0, 0]
 ])
 ])
edge_2 := table([
    1 = N
    2 = [5, 11]
    3 = BE_R
    4 = table([
    inert = [[a, 0, 0], [0, b, 0], [0, 0, c]]
    mass = aa
    ])
    ])
      edge_3 := table([
          1 = N
          2 = [5, 3]
          3 = BE_F
          4 = [0, 3, 0, 0]
          Order_Finert = 2
          Order_Felastic = 2
          scheme = Taylor
          ])
          edge_4 := table([
              1 = Y
              2 = [11, 1]
              3 = JE
              4 = WELD
              ])
```

$edge_5$ := table([
$1 = Y$
$2 = [11, 3]$
$3 = JE$
$4 = RV$
])
$edge_6$ := table([
$1 = Y$
$2 = [3, 9]$
$3 = AE\_F$
$4$ = table([
$coords = [x, 0, 0]$
])
])
$edge_8$ := table([
$1 = N$
$2 = [8, 9]$
$3 = JE$
$4 = SPH$
])
$edge_9$ := table([
$1 = Y$
$2 = [7, 6]$
$3 = AE\_R$
$4$ = table([
$coords = [a, 0, 0]$
])
])

$edge_{10}$ := table([

1 = $N$

2 = [5, 7]

3 = $BE\_R$

4 = table([

$inert$ = [[$a$, 0, 0], [0, $b$, 0], [0, 0, $c$]]

$mass$ = $aa$

])

])

$edge_{11}$ := table([

1 = $Y$

2 = [7, 8]

3 = $AE\_R$

4 = table([

$coords$ = [$a$, 0, 0]

])

])

$edge_{12}$ := table([

1 = $Y$

2 = [5, 6]

3 = $JE$

4 = $RV$

])

$edge_{13}$ := table([

1 = $N$

2 = [5, 6]

3 = $MDE$

4 = table([

$$\theta = \omega t$$

$$force = \theta$$

$$type = JD$$

])

])

$$Iedge := [12]$$

$$========== reading\ my\_pack1 ==========$$

$$========== reading\ my\_pack2 ==========$$

$$========== reading\ my\_pack3 ==========$$

$$========== topo\_matrix\ starts ==========$$

```
Warning: new definition for   norm
Warning: new definition for   trace
```

$$edges := [1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13]$$

$$branch\_list := [1, 4, 5, 6, 9, 11, 12]$$

$$chord\_list := [2, 3, 8, 10, 13]$$

$$node\_list := [1, 3, 5, 6, 7, 8, 9, 11]$$

$$VW\_list := []$$

$$Total\_NOofedges := 12$$

$$INCIDENCE := \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 1 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 & -1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

========== *path_lists starts* ==========

$tree\_top\_nodes := \{8, 9\}$

$path8 := [12, 9, 11]$

$direct8 := [-1, 1, -1]$

$path9 := [1, 4, 5, 6]$

$direct9 := [-1, 1, -1, -1]$

========== *transmatrix starts* ==========

$body\_set := [2, 3, 10]$

$Path2 := [1, 4]$

$direction2 := [-1, 1]$

$Path3 := [1, 4, 5]$

$direction3 := [-1, 1, -1]$

$Path10 := [12, 9]$

$direction10 := [-1, 1]$

$$Tr2 := \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$Tr3 := \begin{bmatrix} -\cos(\theta 5(t)) & \sin(\theta 5(t)) & 0 \\ \sin(\theta 5(t)) & \cos(\theta 5(t)) & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$Tr10 := \begin{bmatrix} \cos(\theta 12(t)) & -\sin(\theta 12(t)) & 0 \\ \sin(\theta 12(t)) & \cos(\theta 12(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

========== *angues starts* ==========

$$V\_ang2 := [000]$$

$$W\_ang2 := [000]$$

$$V\_ang3 := \left[00 - \left(\frac{\partial}{\partial t}\theta 5(t)\right)\right]$$

$$W\_ang3 := \left[00 - \left(\frac{\partial^2}{\partial t^2}\theta 5(t)\right)\right]$$

$$V\_ang10 := \left[00\frac{\partial}{\partial t}\theta 12(t)\right]$$

$$W\_ang10 := \left[00\frac{\partial^2}{\partial t^2}\theta 12(t)\right]$$

---

## A.9 Maple Input File and Partial Output for Spatial Slider Crank with Flexible Link

```
> restart; read'F_splider1.mpl'; read'gen_sys';
```

$$NOofedges := 13$$

$$NOofnodes := 9$$

$$Datum := 5$$

$$edge_1 := \text{table}([$$

$$1 = Y$$

$2 = [5, 1]$

$3 = AE\_S$

$4 = \text{table}([$

$rotmat = [[-1, 0, 0], [0, 1, 0], [0, 0, -1]]$

$coords = [1, 0, 0, 0, 0, 0]$

$])$

$])$

$edge_2 := \text{table}([$

$1 = N$

$2 = [5, 11]$

$3 = BE\_R$

$4 = \text{table}([$

$inert = [[a, 0, 0], [0, b, 0], [0, 0, c]]$

$mass = aa$

$])$

$])$

$edge_3 := \text{table}([$

$1 = N$

$2 = [5, 3]$

$3 = BE\_F$

$4 = [0, 3, 3, 0]$

$Order\_Finert = 2$

$Order\_Felastic = 2$

$scheme = Taylor$

$])$

$edge_4 := \text{table}([$

$1 = Y$

$2 = [11, 1]$

```
        3 = JE
        4 = WELD
        ])
    edge5 := table([
        1 = Y
        2 = [11, 3]
        3 = JE
        4 = UNIV
        ])
edge6 := table([
    1 = Y
    2 = [3, 9]
    3 = AE_F
    4 = table([
    coords = [x, 0, 0]
    ])
    ])
    edge8 := table([
        1 = N
        2 = [8, 9]
        3 = JE
        4 = SPH
        ])
edge9 := table([
    1 = Y
    2 = [7, 6]
    3 = AE_R
    4 = table([
```

$rotmat = [[\cos(\alpha 1), 0, \sin(\alpha 1)], [0, 1, 0], [-\sin(\alpha 1), 0, \cos(\alpha 1)]]$

$coords = [a, 0, 0]$

])

])

$edge_{10}$ := table([

1 = $N$

2 = [5, 7]

3 = $BE\_R$

4 = table([

$inert = [[a, 0, 0], [0, b, 0], [0, 0, c]]$

$mass = aa$

])

])

$edge_{11}$ := table([

1 = $Y$

2 = [7, 8]

3 = $AE\_R$

4 = table([

$coords = [a, 0, 0]$

])

])

$edge_{12}$ := table([

1 = $Y$

2 = [13, 6]

3 = $JE$

4 = $RV$

])

$edge_{13}$ := table([

$1 = N$

$2 = [13, 6]$

$3 = MDE$

$4 = \text{table}([$

$type = JD$

$\theta = \omega t$

$force = \theta$

$])$

$])$

$edge_{16} := \text{table}([$

$1 = Y$

$2 = [5, 13]$

$3 = AE\_R$

$4 = \text{table}([$

$rotmat = [[\cos(\alpha 1), 0, \sin(\alpha 1)], [0, 1, 0], [-\sin(\alpha 1), 0, \cos(\alpha 1)]]$

$coords = [a, a, 0]$

$])$

$])$

$Iedge := [12]$

========== *reading my_pack1* ==========

========== *reading my_pack2* ==========

========== *reading my_pack3* ==========

========== *topo_matrix starts* ==========

```
Warning: new definition for   norm
Warning: new definition for   trace
```

$edges := [1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 16]$

$branch\_list := [1, 4, 5, 6, 9, 11, 12, 16]$

$$chord\_list := [2, 3, 8, 10, 13]$$

$$node\_list := [1, 3, 5, 6, 7, 8, 9, 11, 13]$$

$$VW\_list := []$$

$$Total\_NOofedges := 13$$

$$INCIDENCE := \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$CUTSET := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}$$

$$CIRCUIT := \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

========== *path_lists starts* ==========

$$tree\_top\_nodes := \{8, 9\}$$

$$path8 := [16, 12, 9, 11]$$

$$direct8 := [-1, -1, 1, -1]$$

$$path9 := [1, 4, 5, 6]$$

$$direct9 := [-1, 1, -1, -1]$$

$$========== transmatrix\ starts ==========$$

$$body\_set := [2, 3, 10]$$

$$Path2 := [1, 4]$$

$$direction2 := [-1, 1]$$

$$Path3 := [1, 4, 5]$$

$$direction3 := [-1, 1, -1]$$

$$Path10 := [16, 12, 9]$$

$$direction10 := [-1, -1, 1]$$

$$Tr2 := \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$Tr3 := \begin{bmatrix} -\cos(\alpha 5(t))\cos(\beta 5(t)) & \cos(\alpha 5(t))\sin(\beta 5(t)) & -\sin(\alpha 5(t)) \\ \sin(\beta 5(t)) & \cos(\beta 5(t)) & 0 \\ \sin(\alpha 5(t))\cos(\beta 5(t)) & -\sin(\alpha 5(t))\sin(\beta 5(t)) & -\cos(\alpha 5(t)) \end{bmatrix}$$

$$Tr10 :=$$

$$\left[\cos(\alpha 1)^2\cos(\theta 12(t)) + 1 - \cos(\alpha 1)^2, -\cos(\alpha 1)\sin(\theta 12(t)),\right.$$
$$\left.-\cos(\alpha 1)\cos(\theta 12(t))\sin(\alpha 1) + \sin(\alpha 1)\cos(\alpha 1)\right]$$
$$[\cos(\alpha 1)\sin(\theta 12(t)), \cos(\theta 12(t)), -\sin(\alpha 1)\sin(\theta 12(t))]$$
$$\left[-\cos(\alpha 1)\cos(\theta 12(t))\sin(\alpha 1) + \sin(\alpha 1)\cos(\alpha 1), \sin(\alpha 1)\sin(\theta 12(t)),\right.$$
$$\left.\cos(\theta 12(t)) - \cos(\alpha 1)^2\cos(\theta 12(t)) + \cos(\alpha 1)^2\right]$$

$$========== angues\ starts ==========$$

$$V\_ang2 := [000]$$

$$W\_ang2 := [000]$$

$$V\_ang3 := \left[-\sin(\alpha5(t))\left(\frac{\partial}{\partial t}\beta5(t)\right)\frac{\partial}{\partial t}\alpha5(t) - \cos(\alpha5(t))\left(\frac{\partial}{\partial t}\beta5(t)\right)\right]$$

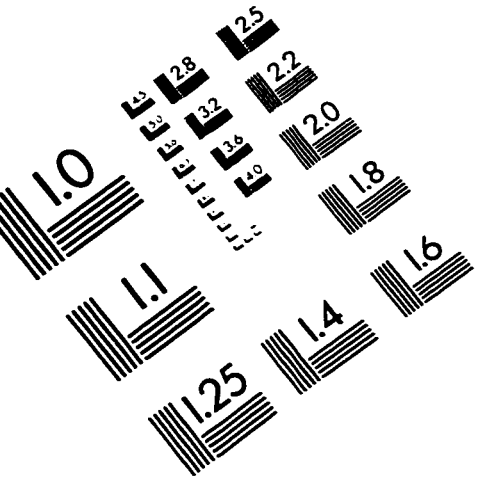
$$W\_ang3 := \left[-\cos(\alpha5(t))\left(\frac{\partial}{\partial t}\alpha5(t)\right)\left(\frac{\partial}{\partial t}\beta5(t)\right) - \sin(\alpha5(t))\left(\frac{\partial^2}{\partial t^2}\beta5(t)\right)\frac{\partial^2}{\partial t^2}\alpha5(t)\right.$$

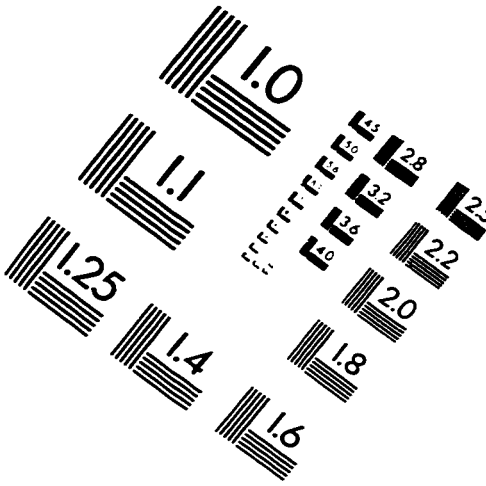
$$\left.\sin(\alpha5(t))\left(\frac{\partial}{\partial t}\alpha5(t)\right)\left(\frac{\partial}{\partial t}\beta5(t)\right) - \cos(\alpha5(t))\left(\frac{\partial^2}{\partial t^2}\beta5(t)\right)\right]$$

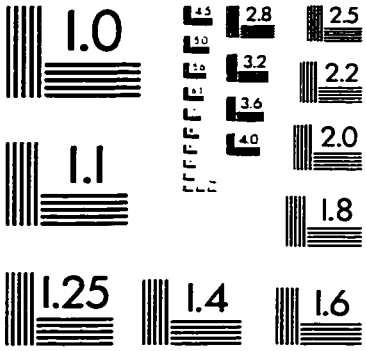
$$V\_ang10 := \left[\sin(\alpha1)\left(\frac{\partial}{\partial t}\theta12(t)\right)0\cos(\alpha1)\left(\frac{\partial}{\partial t}\theta12(t)\right)\right]$$

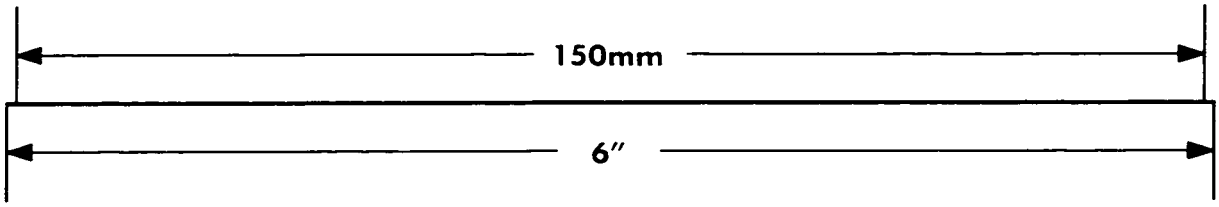
$$W\_ang10 := \left[\sin(\alpha1)\left(\frac{\partial^2}{\partial t^2}\theta12(t)\right)0\cos(\alpha1)\left(\frac{\partial^2}{\partial t^2}\theta12(t)\right)\right]$$

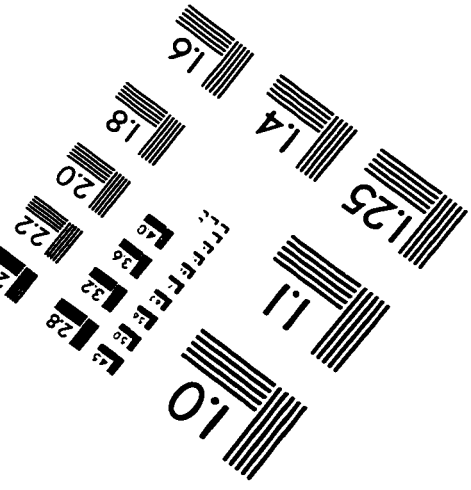
# IMAGE EVALUATION
# TEST TARGET (QA-3)

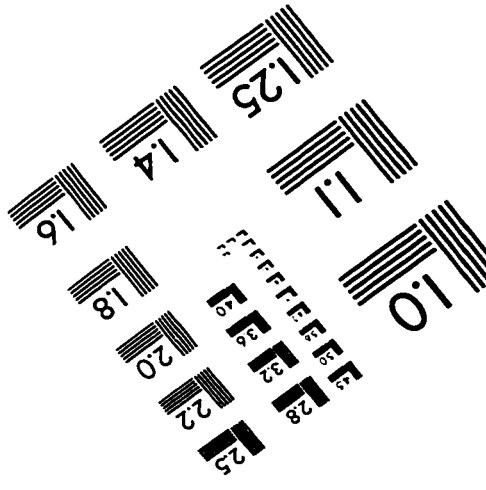



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989