Multibody System Dynamics **4:** 355–381, 2000.
© 2000 *Kluwer Academic Publishers. Printed in the Netherlands.*

Dynamics of Flexible Multibody Systems Using Virtual Work and Linear Graph Theory

PENGFEI SHI and JOHN McPHEE
Systems Design Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1

(Received: 1 September 1998; accepted in revised form: 22 February 2000)

Abstract. By combining linear graph theory with the principle of virtual work, a dynamic formulation is obtained that extends graph-theoretic modelling methods to the analysis of flexible multibody systems. The system is represented by a linear graph, in which nodes represent reference frames on rigid and flexible bodies, and edges represent components that connect these frames. By selecting a spanning tree for the graph, the analyst can choose the set of coordinates appearing in the final system of equations. This set can include absolute, joint, or elastic coordinates, or some combination thereof. If desired, all non-working constraint forces and torques can be automatically eliminated from the dynamic equations by exploiting the properties of virtual work. The formulation has been implemented in a computer program, DynaFlex, that generates the equations of motion in symbolic form. Three examples are presented to demonstrate the application of the formulation, and to validate the symbolic computer implementation.

Key words: flexible multibody dynamics, virtual work, graph theory, symbolic programming.

1. Introduction

A main goal of multibody dynamics is to develop formulations that automatically generate the equations of motion for complex mechanical systems. It has been shown [1] that, for systems of rigid bodies, a graph-theoretic (GT) formulation can generate a smaller set of equations than those obtained using traditional absolute and joint coordinate formulations. Furthermore, the analyst can choose the coordinates to appear in the equations of motion, by selecting a spanning tree for the system graph. GT-based approaches explicitly separate the linear topological equations for the entire system from the nonlinear constitutive equations for individual components, resulting in very modular and efficient algorithms [2]. These, and other positive features of GT formulations, have been addressed in a number of previous analyses of rigid multibody systems [3–5].

However, these systematic procedures have yet to be extended to the dynamic analysis of flexible multibody systems, with the exception of two ad hoc formulations. In the first, Richard and Tennich [6] incorporate models of Euler–Bernoulli beams experiencing small deflections relative to a local frame. The change in moments of inertia of these components is neglected, as is the coupling between

the gross motion of the frame and the deformations of the elastic beams. Their formulation is essentially the same as earlier kineto-elastodynamic formulations [7, 8].

In a subsequent publication, Tennich [9] uses a finite element representation of flexible beams and shells relative to a local moving frame. Although the flexibility effects are treated quite rigorously, the linear graph of the system is only used to generate kinematic constraint equations for the absolute coordinates employed by the author. The dynamic equations are generated using an assembly procedure very similar to that proposed by Shabana [10], which does not require (or exploit) the topological equations available from the GT model. As a result, a very large system of differential-algebraic equations (DAEs) are obtained for relatively simple systems.

In a previous paper [11], we have proposed and validated the use of virtual work as a graph-theoretic 'through variable', a choice that is not intuitive to GT practitioners. By using virtual work (VW) as a new through variable, we have developed a GT formulation that is capable of automatically generating the motion equations for both rigid and flexible multibody systems, given only a description of the system as input. In this paper, the details of this new GT formulation are given, and its application to the dynamic analysis of rigid and flexible multibody systems is discussed.

In the next section, our linear graph representation of a multibody system is presented, and the associated topological equations are discussed. In order to create a system graph that results in correct kinematic and dynamic equations for any choice of spanning tree, it is necessary to introduce a new 'dependent virtual work' element. Thus, our linear graph is somewhat different from the representations used in previous graph-theoretic approaches [1–6]. The constitutive (or 'terminal') equations for the dependent virtual work element and other components in the system model are given.

By combining the system topological equations with the elemental constitutive equations, a systematic formulation procedure is obtained. The details of this formulation are presented and a symbolic computer implementation, developed using Maple and named DynaFlex, is described. A planar slider-crank mechanism is used to demonstrate the steps in the formulation procedure. The DynaFlex program has been validated by its application to a number of benchmark multibody systems; two spatial examples are presented here. The first is an open-loop system of rigid bodies, while the second is a spatial slider-crank mechanism with a flexible connecting rod.

2. System Representation by Linear Graph

Our linear graph representation of a multibody system is most easily described by means of an example: the planar slider-crank mechanism shown in Figure 1. The system consists of only two rigid bodies, a crank and a connecting rod, with

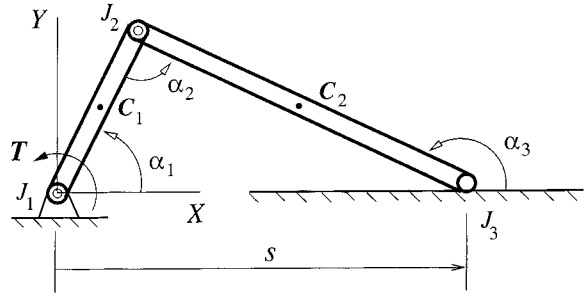

Figure 1. Planar slider-crank mechanism.

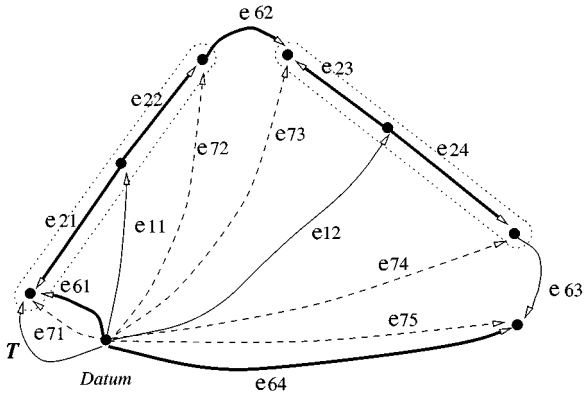

Figure 2. Graph for slider-crank.

mass centers C_1 and C_2, respectively. A driving torque T causes the crank to rotate about the revolute joint J_1, and a second revolute joint J_2 attaches the crank to the connecting rod. The connecting rod is in turn attached to the ground by a revolute-prismatic joint J_3, i.e. a pin moving in a horizontal slot. The variables α_1–α_3 represent angular displacements of the three joints, while s is the translational displacement of joint J_3.

The linear graph representation of the slider-crank is shown in Figure 2. As in previous graph-theoretic models [1, 4], nodes (or vertices) are used to represent reference frames in the system, while edges (lines) represent physical elements that connect these frames. Thus, edge $e22$ represents the position and orientation of a reference frame at the distal end of the crank relative to a reference frame at C_1. A similar interpretation applies to the other 'rigid-arm' elements $e21$, $e23$, and $e24$. Edges $e61$ and $e62$ represent revolute joints J_1 and J_2, respectively, while the compound revolute-prismatic joint J_3 is represented by edges $e64$ (for translation) and $e63$ (for rotation). Edges $e11$ and $e12$ represent the motion of the two rigid bodies relative to the inertial XY frame, represented by the datum node, and edge T is the driving torque.

Similar to conventional graph-theoretic representations, the structure of the physical system is easily seen in the linear graph; this point has been emphasized by drawing the crank and connecting rod in dotted lines on top of the graph. Unlike some previous approaches though, a single linear graph is used to represent both translational and rotational motions of the individual components. This provides an advantageous simplification of the graph representation. This is also a disadvantage though, since different spanning trees cannot be selected for individual translational and rotational graphs in order to further reduce the number of kinematic equations in a systematic manner [1].

A second difference between previous linear graphs and that shown in Figure 2 is the appearance of 'dependent virtual work' elements $e71$–$e75$. They are shown in dashed lines because they can be automatically added to the system graph once the topology is defined, as explained below. Quite simply, they must be added to the system graph in order to generate the correct topological equations at every vertex, as described next.

2.1. DYNAMIC EQUATIONS: VIRTUAL WORK

There are two main types of topological equations that can be automatically generated from any graph: the vertex equations and the circuit equations. The vertex equations, from which the cutset and chord transformation equations are obtained, represent a balance of through variables at every node in the graph. They take the form of Kirchoff's current law for electrical systems, and provide dynamic equilibrium equations when used with multibody systems. For the slider-crank example, the vertex equation for the node representing the distal point of the crank is

$$\underline{F}_{22} + \underline{F}_{72} - \underline{F}_{62} = 0, \tag{1}$$

by using conventional through variables: force vectors. With $\underline{F}_{72}$ defined to be zero for the dependent virtual work element, it is clear that the resulting vertex equation $\underline{F}_{22} = \underline{F}_{62}$ represents a balance of forces at the corresponding node. Based on the results in [11], we can also use virtual work δW as a through variable:

$$\delta W_{22} + \delta W_{72} - \delta W_{62} = 0, \tag{2}$$

where each term corresponds directly to a physical component.

By definition, the virtual work of the pin reaction force $\underline{F}_{62}$ on the crank depends upon the absolute displacement of the distal point of the crank. Since edge $e62$ represents only relative displacements, it is necessary to introduce the dependent virtual work element $e72$ to correctly capture the virtual work done by joint J_2 on the crank. In a similar manner, edge $e73$ represents the virtual work done by the same joint on the connecting rod. In general, whenever two bodies are connected by a component (e.g., joint, spring, damper), two dependent virtual work elements should automatically be included in the system model, each originating from the datum node and terminating at the end points of the component.

For ideal joints, the net virtual work must equal zero, since a frictionless joint cannot add or remove energy from the system. As an example, $\delta W_{62} = 0$ for revolute joint J_2. Furthermore, the sum of the virtual works done by joint J_2 on the crank and connecting rod must also be zero:

$$\delta W_{72} + \delta W_{73} = 0. \tag{3}$$

This fact can be exploited in our GT formulation to eliminate all joint reactions, which appear in the constitutive equations for dependent virtual work elements, from the final equations of motion. Alternatively, joint reactions can be retained in the GT formulation by means of Lagrange multipliers.

An alternative to the vertex equations is the set of 'cutset equations', in which there is one independent equation for each edge in a spanning tree of the graph. A tree is a collection of edges that connects all nodes and doesn't contain any closed loops, or 'circuits'. The tree for the slider-crank, shown in bold in Figure 2, contains 7 edges, or 'branches'. The remaining edges making up the complement of the tree, or cotree, are known as chords. The 7 cutset equations corresponding to the branches can be generated automatically [2] from the system graph. As an example, the cutset equation for the tree joint element $e62$ is

$$\delta W_{62} + \delta W_{73} + \delta W_{12} + \delta W_{74} - \delta W_{63} = 0 \tag{4}$$

which could also be written with force or torque in place of the virtual work δW. The cutset equations represent linear combinations of the vertex equations and, as such, they provide dynamic equilibrium equations for entire bodies or subsystems. Equation (4), for example, is the sum of vertex equations for all nodes on the connecting rod. Thus, it represents the equation that would be obtained by applying the principle of virtual work to a free body diagram of the connecting rod.

2.2. KINEMATIC EQUATIONS: THREE-DIMENSIONAL ROTATIONS

The second main set of topological equations, the circuit equations, can be generated automatically from the set of cutset equations. There is one independent circuit equation for each chord in the graph. As an example, the circuit equation for edge $e12$ is

$$\underline{r}_{12} + \underline{r}_{23} - \underline{r}_{62} - \underline{r}_{22} + \underline{r}_{21} - \underline{r}_{61} = 0, \tag{5}$$

where $\underline{r}$ represents the translational displacement of the corresponding element. It is easy to see that this linear equation represents the vector loop closure condition for the circuit containing $e12$. Since each circuit equation only contains one cotree displacement, the circuit equations can be solved for the cotree displacements as functions of the tree displacements, giving the so-called branch transformation equations. These transformation equations are used in the GT formulation to eliminate all cotree variables from the final equations of motion.

The across variable $\underline{r}$ in Equation (5) can be replaced with the translational velocity $\underline{v}$ or acceleration $\underline{a}$, or the angular velocity $\underline{\omega}$ or acceleration $\underline{\dot{\omega}}$. The resulting velocity and acceleration equations are equally valid, and can also be re-written as branch transformation equations. We can also replace $\underline{r}$ with the virtual displacement $\delta\underline{r}$ or even the virtual rotation $\delta\underline{\theta}$, since infinitesimal rotations will satisfy the resulting vector sum.

However, one cannot replace $\underline{r}$ with a finite rotational displacement, since three-dimensional rotations do not satisfy the parallelogram law for vector addition. Instead, one must define an 'equivalent' circuit equation in terms of the rotation transformation matrices associated with each element. As an example, the corresponding equation for edge $e12$ would take the form:

$$[R_{12}][R_{23}][R_{62}]^T[R_{22}]^T[R_{21}][R_{61}]^T = [1], \tag{6}$$

where [1] is unit matrix. Examining Equations (5) and (6), one can see that the additive nature of the conventional circuit equation is replaced by a multiplicative expression in its rotational counterpart, and the direction of edges around the circuit is now accounted for by using (or not) the transpose operator. It is clear that the rotational equation (6) can easily be generated from the conventional circuit equation (5).

If desired, Equation (6) can be re-written as a branch transformation equation:

$$[R_{12}] = [R_{61}][R_{21}]^T[R_{22}][R_{62}][R_{23}]^T, \tag{7}$$

making use of the orthogonal nature ($[R]^{-1} = [R]^T$) of rotation transformation matrices. Each of the rotation matrices will be functions of the rotation variable(s) for the corresponding element, i.e. $[R_{61}] = [R_{61}(\alpha_1)]$. Thus, equation (7) can be used to express the orientation of the connecting rod as a function of the angular variables associated with all edges selected into the tree.

In summary, these branch transformation equations can be used to express the kinematic and dynamic equations solely in terms of the across variables associated with branches in the tree. For obvious reasons, these unknown tree across variables are called 'branch coordinates' [1]. Thus, a powerful feature of a GT formulation is that the analyst has control over the branch coordinates appearing in the final equations of motion, through the selection of edges into the spanning tree.

3. Component Models

To generate a complete set of motion equations for a multibody system, the topological cutset and circuit equations must be supplemented by constitutive, or 'terminal', equations for individual components. These terminal equations are empirical relationships between the through and across variables for the element, and time. In terms of the conventional through and across variables $\underline{r}$, $\underline{v}$, $\underline{a}$, $\underline{\omega}$, $\underline{\dot{\omega}}$, $\underline{F}$, and $\underline{T}$, the terminal equations for a number of components that commonly appear in multibody system models are given in [1] and [4]. In this section, we present

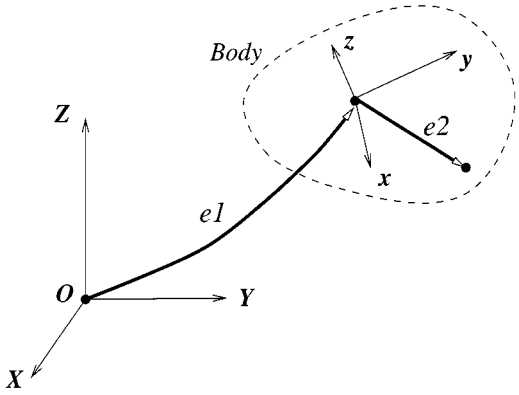

Figure 3. Graphs of body and arm elements.

the terminal equations for many of the same components, but expressed in terms of virtual displacements and virtual work.

3.1. BODY ELEMENTS

A body element represents a rigid or flexible body in the system. As shown in Figure 3, the graph for this element consists of an edge $e1$ from the datum node O to a local xyz reference frame on the body. For a rigid body, the local origin is taken to coincide with the center of mass. For a flexible body, the location and orientation of the local frame will depend upon a number of factors, including the approach taken to represent its flexibility.

For a rigid body, the virtual work terminal equation is

$$\delta W_1 = -m_1 \underline{\ddot{r}}_1 \cdot \delta \underline{r}_1 - \left[\underline{\underline{I}}_1 \cdot \underline{\dot{\omega}}_1 + \underline{\omega}_1 \times \underline{\underline{I}}_1 \cdot \underline{\omega}_1 \right] \cdot \delta \underline{\theta}_1, \tag{8}$$

where m_1 is the mass and $\underline{\underline{I}}_1$ is the inertia dyadic of the body, $\underline{\ddot{r}}_1$ and $\underline{\dot{\omega}}_1$ are the translational and rotational accelerations of the element, and $\delta \underline{r}_1$ and $\delta \underline{\theta}_1$ are virtual translational and rotational displacements. Again note that the virtual rotation $\delta \underline{\theta}_1$ can be treated as a vector. It is clear that Equation (8) represents the virtual work δW_1 done by the inertial force and torque associated with the body.

For a flexible body, the constitutive virtual work equation is

$$\delta W_1 = -\int_V \{\delta\epsilon\}^T \{\sigma\} \, \mathrm{d}V + \int_V \{\delta r\}^T \left(\{f_b\} - \gamma \{\ddot{r}\} \right) \mathrm{d}V, \tag{9}$$

where $\{\delta\epsilon\}$ is the column matrix of varied strain components in the local frame, $\{\sigma\}$ contains the corresponding stress components, $\{f_b\}$ is the body force, including gravity, and $-\gamma\{\ddot{r}\}$ is the inertial force per unit volume. The first volume integral in Equation (9) represents the strain energy, while the second integral is the virtual work of body forces, including gravity and inertial forces. For use with our dynamic

formulation, all terms in Equation (9) should be written in terms of generalized elastic coordinates $\mathbf{q}_f(t)$ that describe the deformation of the flexible body. One method for doing this is described in [12], using Euler–Bernoulli beam theory, polynomial shape functions, and a second-order deformation field.

3.2. ARM ELEMENTS

An arm element is used to represent the relative position and orientation of two reference frames on the same body. As shown in Figure 3, the graph for an arm element consists of an edge $e2$ from the origin of the local xyz body frame to another point on the body. This second reference frame may be needed to locate the point of application of a force, torque, joint, or imposed motion.

If the body is rigid, the relative orientation of the two body-fixed reference frames is constant and the terminal equations in terms of virtual displacements are

$$\delta\underline{\theta}_2 = 0, \tag{10}$$

$$\delta\underline{r}_2 = \delta\underline{\theta}_1 \times \underline{r}_2, \tag{11}$$

where $\underline{r}_2$ is the position vector associated with the rigid-arm element; it is a function of $\underline{\theta}_1$ since the direction of $\underline{r}_2$ varies as the rigid body rotates. For an arm on a flexible body, the relative displacements and rotations will also be functions of the elastic coordinates $\mathbf{q}_f$ describing the deformation of the body [12].

3.3. ABSOLUTE DRIVERS

A 'driver' element is one for which the through or across variables are explicit functions of time. A motion driver is used to specify the time-varying position or orientation of a body-fixed reference frame with respect to an inertial frame, while a force driver is used to model an external force or torque applied to some point on a body.

In either case, the graph of the element consists of a single edge from the datum node to the body-fixed point, and the virtual work terminal equation is

$$\delta W = \underline{F} \cdot \delta\underline{r} + \underline{T} \cdot \delta\underline{\theta}, \tag{12}$$

where the force $\underline{F}$ or torque $\underline{T}$ are given functions for force drivers, and the position $\underline{r}$ or orientation $\underline{\theta}$ are known functions of time for motion drivers. This does not imply that $\delta\underline{r}$ or $\delta\underline{\theta}$ are equal to zero for motion drivers; they cannot be, since the virtual work is not zero. Instead, $\delta\underline{r}$ and $\delta\underline{\theta}$ are expressed in terms of the variations of branch coordinates using the branch transformation equations.

3.4. DEPENDENT VW ELEMENTS

As discussed in a previous section, this new element is used to correctly represent the virtual work done on each of two bodies by a physical component (e.g. joint, spring, damper, actuator) that connects the two bodies. If the force and torque associated with this inter-body component are designated as $\underline{F}_I$ and $\underline{T}_I$, respectively, the terminal equation for a dependent virtual work element is simply:

$$\delta W = \underline{F}_I \cdot \delta \underline{r} + \underline{T}_I \cdot \delta \underline{\theta}, \tag{13}$$

where $\delta \underline{r}$ and $\delta \underline{\theta}$ are the virtual displacement and rotation of the dependent virtual work element. The graph of this element consists of an edge from the datum node to the point on the body where the inter-body component is attached.

3.5. INTER-BODY COMPONENTS

Included in this category are ideal joints, springs, dampers, motion drivers between bodies, and force and torque actuators (motors). Although the conventional terminal equations for these components are very different [13], these components all have the same terminal equation in terms of virtual work:

$$\delta W = 0. \tag{14}$$

This is because the virtual work done by these components is accounted for by a pair of dependent virtual work elements that terminate at the endpoints of the inter-body component. Thus, the linear graph representation of any one of these components consists of three edges: one between the two endpoints, and two dependent virtual work elements. Several examples of this 3-edge model for an ideal joint have already been shown in Figure 2 and discussed.

4. Automated Formulation of Equations

Together, the topological and terminal equations form a necessary and sufficient set of motion equations for a multibody system. Thus, one could obtain the kinematic or dynamic response of a system by generating all of the cutset, circuit, and terminal equations and solving them simultaneously. A more efficient approach is to reduce the number of equations that need to be solved simultaneously by using the branch transformation equations for a particular tree selection. Such an approach has been taken in the graph-theoretic formulation that is presented in this section. A symbolic computer implementation of this 4-step formulation procedure, developed using Maple, is also described.

From a description of the physical components making up a multibody system model, a linear graph representation can be automatically generated. It is then necessary to select a spanning tree for the graph; this can also be automated, or left to the analyst to choose. In either case, the arm elements should always be selected

into the tree first, since their across variables are essentially known quantities. This has the desirable effect of reducing the number of branch coordinates $\{q\}$ and, consequently, the final number of equations to be solved [1].

By allowing the analyst to choose the tree, the set of coordinates appearing in the final equations can be controlled. As an example, the equations can be generated in joint coordinates [14] by selecting as many joints as possible into the tree (hereafter called a 'joint tree'). Alternatively, the equations will be in terms of absolute coordinates [15] if all bodies are selected into the tree. In either case, the elastic coordinates $\mathbf{q}_f$ will also appear in the equations of motion if there are flexible bodies in the system. A mix of bodies, joints, or even force and torque elements, can be selected into the tree to obtain a hybrid set of branch coordinates and equations. The freedom to control the coordinates appearing in the equations generated by a GT formulation has both practical and pedagogical advantages.

With the ability to generate equations in terms of absolute, joint, or elastic coordinates, or some combination thereof, our GT formulation represents a generalization of traditional absolute and joint coordinate formulations. We know of one other formulation [16] that allows the user to choose absolute or joint coordinates, or some mix of both, but this is accomplished through the use of object-oriented programming techniques. Thus, one cannot easily see the effects of different coordinate selections on the symbolic form of the equations of motion.

Our GT formulation also allows the analyst to retain non-working constraint reactions in the dynamic equations via Lagrange multipliers, or to eliminate these reactions by exploiting the principle of virtual work. These two approaches have been called [10, 14] the 'augmented formulation' and the 'embedding technique', respectively, although other names have been used in the literature. Both approaches have been included in our GT formulation. In the embedding technique, it is necessary to identify a subset of $\{q\}$ that are independent. For a system with f degrees of freedom, there will be f independent coordinates $\{q_i\}$. Once again, these independent coordinates can be selected automatically [17], or specified by the analyst.

Once the linear graph has been constructed, the equations of motion for the multibody system can be automatically generated by systematically applying the following 4-step procedure.

4.1. STEP 1: BRANCH TRANSFORMATIONS

The first step is to generate the branch transformation equations, described previously in Section 2.2, for all bodies, springs, dampers, actuators, forces, torques, and dependent virtual work elements in the cotree. These transformations will be used in subsequent steps to replace the across variables for these cotree elements with expressions involving only the branch coordinates $\{q\}$. If a joint tree is used, then these transformation equations for cotree bodies will be identical to the velo-

city transformation equations [18, 19] employed by other researchers in multibody dynamics.

The chord transformation equations that are used in conventional graph-theoretic formulations [1, 4] are not required in this virtual work approach.

4.2. STEP 2: KINEMATIC EQUATIONS

Depending on the topology of the physical system and the tree specified for the linear graph, the branch coordinates $\{q\}$ may not be independent quantities. If the number of coordinates n is greater than the degrees of freedom f, then $m = n - f$ kinematic 'constraint' equations are required to express the dependency between coordinates.

These constraint equations are obtained directly from the joints and motion drivers in the cotree, by projecting their circuit equations onto the joint reaction space [1]. One can obtain an independent set of branch coordinates for an open-loop system by selecting all of the joints into the tree, thereby avoiding the need for constraint equations. For systems with closed kinematic chains however, an independent set of $\{q\}$ cannot be obtained by a clever tree selection; at least one joint will appear in the cotree, giving constraint equations in terms of $\{q\}$.

Upon substitution of the terminal and branch transformation equations from Step 1 into these circuit equations, the constraint equations are obtained in the following form:

$$\{\Phi(q, t)\} = 0 \tag{15}$$

which constitute a set of m nonlinear algebraic equations. By generating the circuit equations with displacements replaced by velocities, one obtains

$$\{\Phi\}_q\{\dot{q}\} = -\{\Phi\}_t, \tag{16}$$

where the q and t subscripts indicate partial derivatives with respect to $\{q\}$ and time, respectively. Replacing the velocities with accelerations, one gets

$$\{\Phi\}_q\{\ddot{q}\} = -\left(\{\Phi\}_q\{\dot{q}\}\right)_q\{\dot{q}\} - 2\{\Phi\}_{qt}\{\dot{q}\} - \{\Phi\}_{tt}. \tag{17}$$

Alternatively, the velocity and acceleration equations (16) and (17) can be obtained by direct symbolic differentiation of the position-level constraint equations (15).

Symbolic differentiation can also be used to obtain a transformation from $\{q\}$ to the set of f independent coordinates $\{q_i\}$, which is needed for the embedding technique. This is done by partitioning $\{q\}$ into $\{q_i\}$ and m dependent coordinates $\{q_d\}$:

$$\{q\} = \begin{Bmatrix} q_d \\ q_i \end{Bmatrix} \tag{18}$$

and then taking the variation of Equation (15):

$$\{\Phi\}_{q_d}\{\delta q_d\} + \{\Phi\}_{q_i}\{\delta q_i\} = 0, \tag{19}$$

where $\{\Phi\}_{q_d}$ is a symbolic $m \times m$ matrix that is non-singular as long as the given physical constraints are not redundant. Thus, this last equation can be solved for the dependent variations:

$$\{\delta q_d\} = \{\Phi\}_{q_d}^{-1}\{\Phi\}_{q_i}\{\delta q_i\} \tag{20}$$

$$= [J]\{\delta q_i\} \tag{21}$$

which can be used to express the virtual work equations solely in terms of the independent variations $\{\delta q_i\}$. Equation (21) can also be obtained from the velocity constraint equation (16) by using the 'kinematical approach' described by Ginsberg [20].

If f independent motion drivers are acting on the system, then a kinematic analysis is required. In this case, the number of branch coordinates (unknown tree across variables) is reduced to $n - f = m$, which are obtained from the solution of Equation (15). Subsequently, Equations (16) and (17) can be solved for $\{\dot{q}\}$ and $\{\ddot{q}\}$, respectively.

If less than f motion drivers are given, then the problem requires a dynamic analysis. In this case, the m constraint equations must be supplemented by the dynamic equations that are generated by the next two steps.

4.3. STEP 3: CUTSET EQUATIONS

The dynamic equations will be extracted from a single expression for the virtual work of the entire system, consistent with a conventional application of the virtual work principle [20]. To get the contribution of all physical components to the system virtual work, the cutset equations are written for all bodies and 'border joints' in the tree. A border joint connects the ground body to one or more bodies that are in the cotree, through an unbroken path of branches. By summing the cutset equations for all tree bodies and border joints, the contributions of all physical components to the system virtual work equation are captured:

$$\delta W = \sum_{j=1}^{n} Q_j \, \delta q_j = \{Q\}^T\{\delta q\} = 0, \tag{22}$$

where Q_j is the generalized force associated with branch coordinate q_j. Note that only a subset of the system cutset equations are needed.

4.4. STEP 4: DYNAMIC EQUATIONS

The last step is to extract the dynamic equations from the system virtual work equation (22). One can take two different approaches to do this [10, 13, 14], depending upon whether one wishes to include the reaction loads for cotree joints in the dynamic equations.

4.4.1. *Augmented Formulation*

In the augmented formulation, all n branch coordinates are treated as independent, so that the corresponding generalized forces can be set equal to zero:

$$Q_j(q, \dot{q}, \ddot{q}, \lambda, t) = 0, \quad j = 1, \ldots, n, \tag{23}$$

giving n differential equations in which the reaction forces and torques for cotree joints appear as Lagrange multipliers $\{\lambda\}$. Equation (23) can be written in matrix form as

$$[M]\{\ddot{q}\} + \{\Phi\}_q^T\{\lambda\} = \{F(q, \dot{q}, t)\}, \tag{24}$$

where $[M]$ is the symmetric mass matrix, and $\{F\}$ contains forcing and quadratic velocity terms.

In general, the branch coordinates are not independent, but are related by the m kinematic constraint equations (15). Thus, these constraint equations must be appended to the set of dynamic equations (24), giving $n + m$ differential-algebraic equations (DAEs) to solve for $\{q(t)\}$ and the m reaction loads $\{\lambda(t)\}$. If the branch coordinates are independent, the Lagrange multipliers disappear from Equation (24), which reduces to a set of n ordinary differential equations in $\{q\}$.

4.4.2. *Embedding Formulation*

In the embedding technique, Equation (21) is used to eliminate the dependent variations $\{\delta q_d\}$ from the system virtual work equation (22), giving

$$\begin{aligned} \delta W &= \{Q\}^T \begin{bmatrix} J \\ 1 \end{bmatrix} \{\delta q_i\} \\ &= \{Q_i\}^T\{\delta q_i\} = 0, \end{aligned} \tag{25}$$

where Q_i is the generalized force associated with the independent variation δq_i.

By employing independent variations in this second approach, the joint reaction loads $\{\lambda\}$ are automatically eliminated from the virtual work expression. This is because the virtual work of the two dependent elements associated with each joint are guaranteed to cancel out, as predicted by equations such as (3). Extracting the generalized forces associated with each independent variation gives one kinetic equation for each system degree of freedom:

$$Q_i(q, \dot{q}, \ddot{q}, t) = 0, \quad i = 1, \ldots, f, \tag{26}$$

or, in matrix form:

$$[\tilde{M}]\{\ddot{q}\} = \{\tilde{F}(q, \dot{q}, t)\}, \tag{27}$$

where $[\tilde{M}]$ is an unsymmetric $(f \times n)$ mass matrix. Together, the $m + f$ DAEs (15) and (27) can be solved for the n branch coordinates $\{q\}$. Clearly, this is a smaller

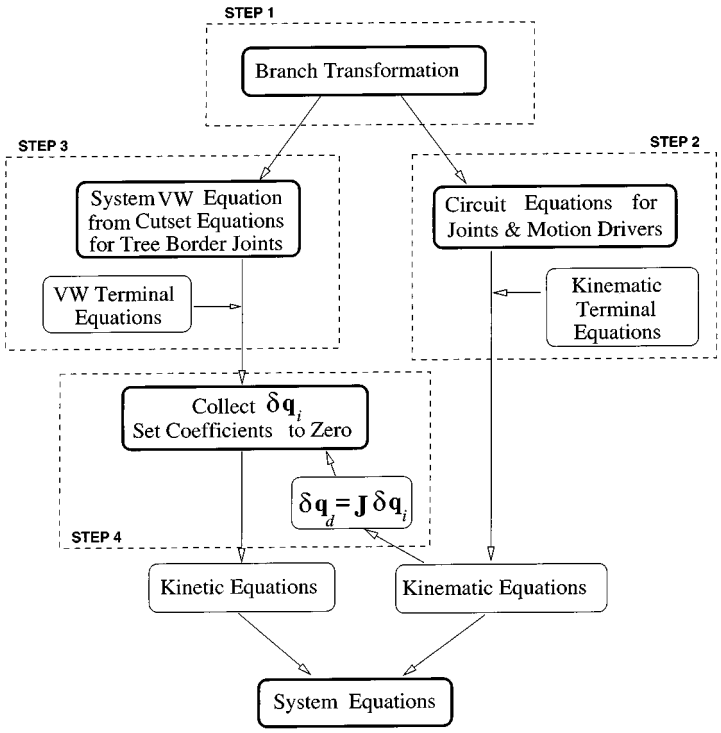

Figure 4. DynaFlex flowchart.

number of motion equations than those generated by the augmented formulation, which is used in many commercial multibody computer programs.

4.5. COMPUTER IMPLEMENTATION

Using the symbolic language Maple, we have implemented the four steps of the previous formulation into a computer program called DynaFlex. Given only a physical description of a multibody system, this program automatically generates the equations of motion in symbolic form. A joint tree is used to keep the number n of branch coordinates, and therefore the number m of constraint equations, relatively small. This is particularly important for the embedding technique because the transformation equation (21) requires the symbolic inversion of an $m \times m$ matrix. For large systems with many closed kinematic chains, a 'dummy' matrix inverse can be used in the symbolic formulation, with the matrix entries evaluated during the subsequent numerical solution. This symbolic-numeric approach was successfully demonstrated on the inverse dynamic analysis of a Gough–Stewart platform [21].

Using a joint tree, only the cutset equations for tree border joints are needed for the system virtual work expression, since all bodies must be in the cotree. For open-loop systems, a joint tree results in independent branch coordinates and a set

of ordinary differential equations, which are generally easier to solve than DAEs. Finally, the automatic use of a joint tree relieves the user of the task of selecting a tree for the system. Instead, the only extra input required is the selection of the augmented or embedding formulation and, for the latter case, the identification of the independent coordinates $\{q_i\}$. Thus, the user is not required to have a deep understanding of graph-theoretic modelling.

The flowchart for the DynaFlex algorithm is shown in Figure 4. One can see that the computer algorithm closely mirrors the steps of the formulation given above. The use of Maple greatly facilitates the manipulation of variables and symbolic extraction of dynamic equations. By generating the equations of motion in symbolic form, it is easier to check the validity of the equations and to extract additional information regarding their structure and any first integrals of motion. To obtain numerical solutions for the system response, DynaFlex exports the system equations in optimized C or Fortran format, and solves them using established numerical algorithms for differential-algebraic equations [13]. Numerical methods must also be used to handle lockup or bifurcation configurations, for which the Jacobian matrices $\{\Phi\}_q$ and $\{\Phi\}_{q_d}$ become singular when evaluated. There are several numerical methods for singularity handling [14, 15] that can be applied to our symbolic equations. Although a detailed discussion is beyond the scope of this paper, we speculate that these methods might by facilitated by our formulation, which could re-generate system equations in a new set of coordinates whenever a singular configuration is approached.

5. Examples

In this section, the dynamic analyses of three different multibody systems are presented. In the first example, the 4-step derivation of the equations of motion for the slider-crank mechanism from Figure 1 is presented in some detail, to illustrate the formulation procedure. The second example is a three-dimensional chain of rigid bodies, for which the symbolic equations of motion are generated using DynaFlex. The third example is a spatial slider-crank with a flexible connecting rod; the results from a numerical solution of the differential-algebraic equations generated by DynaFlex are compared against the results from a finite element approach.

5.1. SLIDER-CRANK RE-VISITED

As mentioned previously, the joint tree for the linear graph representation of the slider-crank is shown in bold in Figure 2. The branch coordinates $\{q\}$ consist of α_1, α_2, and s, the coordinates associated with tree joint elements $e61$, $e62$, and $e64$, respectively. We have used the embedding formulation to eliminate the reaction force for cotree joint element $e63$, and selected α_1 as the independent generalized coordinate for this one degree of freedom system.

Step 1. From the linear graph representation, the following branch transformation equations are automatically generated.

For cotree edge T,

$$\theta_T = \theta_{61} = \alpha_1, \tag{28}$$

noting that two-dimensional angular displacements θ can be treated as vectors.

For body edge $e11$,

$$\theta_{11} = \theta_{61} - \theta_{21} = \alpha_1, \tag{29}$$

$$\begin{aligned} \{r_{11}\} &= \{r_{61}\} - \{r_{21}\} \\ &= [R(\theta_{11})]\{r_{21}\}', \end{aligned} \tag{30}$$

where $\{r_{21}\}'$ is a 2×1 column matrix containing the constant components of $\underline{r}_{21}$ in a local frame attached to the crank, and

$$[R(\theta_{11})] = \begin{bmatrix} \cos\theta_{11} & -\sin\theta_{11} \\ \sin\theta_{11} & \cos\theta_{11} \end{bmatrix} \tag{31}$$

is the rotation transformation matrix from this local frame to the inertial XY frame.

For body edge $e12$,

$$\theta_{12} = \theta_{61} - \theta_{21} + \theta_{22} + \theta_{62} - \theta_{23} = \alpha_1 + \alpha_2, \tag{32}$$

$$\begin{aligned} \{r_{12}\} &= \{r_{61}\} - \{r_{21}\} + \{r_{22}\} + \{r_{62}\} - \{r_{23}\} \\ &= [R(\theta_{11})](\{r_{22}\}' - \{r_{21}\}') - [R(\theta_{12})]\{r_{23}\}', \end{aligned} \tag{33}$$

where, once again, the symbol $\{\}'$ denotes components in a frame fixed to the body on which the arm resides.

Note that kinematic terminal equations, such as $\{r_{61}\} = 0$ and $\theta_{61} = \alpha_1$ have been substituted into the branch transformation equations (28–33). Note also that these equations can be written directly in terms of $\underline{v}$, $\underline{a}$, $\underline{\omega}$, and $\underline{\dot{\omega}}$. Using the branch transformation equations, all displacements (velocities, accelerations) can be written as explicit functions of $\{q\} = [\alpha_1, \alpha_2, s]^T$.

Step 2. Since the system has $f = 1$ degree of freedom and $n = 3$ branch coordinates, $m = 2$ constraint equations are required. These are obtained from the circuit equations for cotree joints and motion drivers: in this case, only joint element $e63$. The corresponding circuit equation, projected onto the joint reaction space, is

$$\{r_{61}\} + \{r_{22}\} - \{r_{21}\} + \{r_{62}\} + \{r_{24}\} - \{r_{23}\} - \{r_{64}\} = 0, \tag{34}$$

or, after substitution of kinematic terminal equations,

$$[R(\theta_{11})](\{r_{22}\}' - \{r_{21}\}') + [R(\theta_{12})](\{r_{24}\}' - \{r_{23}\}') - \{r_{64}\} = 0. \tag{35}$$

Substituting the branch transformation equations (29, 32) into this circuit equation and evaluating, one obtains the two constraint equations:

$$\{\Phi(q)\} = \begin{Bmatrix} l_{11}\cos\alpha_1 - l_{12}\cos(\alpha_1+\alpha_2) - s \\ l_{11}\sin\alpha_1 - l_{12}\sin(\alpha_1+\alpha_2) \end{Bmatrix} = 0, \tag{36}$$

where l_{11} and l_{12} are the lengths of the crank and connecting rod, respectively.

Using symbolic differentiation of the constraint equation (36) as described in the previous section, one gets the transformation corresponding to Equation (21):

$$\begin{aligned} \begin{Bmatrix} \delta\alpha_2 \\ \delta s \end{Bmatrix} &= \begin{bmatrix} l_{12}\sin(\alpha_1+\alpha_2) & -1 \\ -l_{12}\cos(\alpha_1+\alpha_2) & 0 \end{bmatrix}^{-1} \begin{Bmatrix} -l_{11}\sin\alpha_1 \\ l_{11}\cos\alpha_1 \end{Bmatrix} \{\delta\alpha_1\} \\ &= \begin{Bmatrix} l_{11}\cos\alpha_1 / l_{12}\cos(\alpha_1+\alpha_2) \\ l_{11}\sin\alpha_1 - l_{11}\cos\alpha_1\tan(\alpha_1+\alpha_2) \end{Bmatrix} \{\delta\alpha_1\} \\ &= [J]\{\delta\alpha_1\}. \end{aligned} \tag{37}$$

Step 3. Only tree joint element $e61$ connects the system of bodies back to the ground body through a path consisting entirely of branches. Writing the cutset equation for this tree body joint in terms of virtual work,

$$\begin{aligned} \delta W_T &+ \delta W_{71} + \delta W_{61} + \delta W_{11} + \delta W_{72} \\ &+ \delta W_{73} + \delta W_{12} + \delta W_{74} - \delta W_{63} = 0. \end{aligned} \tag{38}$$

Note that δW_{61} and δW_{63} are both zero by virtue of their terminal equation in Section 3.5, and that δW_{72} and δW_{73} cancel out by virtue of Equation (3). Since the ideal joints $e61$ and $e63$ can do no work when connected to the ground body, the virtual work of the corresponding dependent elements δW_{71} and δW_{74} must also equal zero. Thus, the system virtual work equation reduces to

$$\delta W = \delta W_T + \delta W_{11} + \delta W_{12} = 0 \tag{39}$$

which includes the contributions of all physical elements capable of doing work.

Substituting the terminal equations from Section 3 into Equation (39),

$$\begin{aligned} \delta W &= T\delta\theta_T - m_{11}\{a_{11}\}^T\{\delta r_{11}\} - I_{11}\,\dot{\omega}_{11}\delta\theta_{11} \\ &\quad - m_{12}\{a_{12}\}^T\{\delta r_{12}\} - I_{12}\,\dot{\omega}_{12}\,\delta\theta_{12} = 0, \end{aligned} \tag{40}$$

using the two-dimensional form of the rigid body terminal equations, where I_{11} and I_{12} are the centroidal moments of inertia of the crank and connecting rod, respectively, about a local z axis. Substituting the branch transformation equations into Equation (40), one obtains an expression for the system virtual work entirely in terms of the branch coordinates:

$$\delta W = Q_1\,\delta\alpha_1 + Q_2\,\delta\alpha_2 = 0, \tag{41}$$

where the generalized forces Q_1 and Q_2 are complicated functions of α_1, α_2, and their derivatives:

$$\begin{aligned} Q_1 &= T - \left[I_{11} + I_{12} + \frac{m_{11} l_{11}^2}{4} + m_{12} \left(l_{11}^2 + \frac{l_{12}^2}{4} \right) - m_{12} l_{11} l_{12} \cos \alpha_2 \right] \ddot{\alpha}_1 \\ &\quad - \left[I_{12} + m_{12} \frac{l_{12}^2}{4} - m_{12} \frac{l_{11} l_{12}}{2} \cos \alpha_2 \right] \ddot{\alpha}_2 \\ &\quad - m_{12} \frac{l_{11} l_{12}}{2} \left(\dot{\alpha}_2^2 + 2 \dot{\alpha}_1 \dot{\alpha}_2 \right) \sin \alpha_2, \end{aligned} \tag{42}$$

$$\begin{aligned} Q_2 &= - \left[I_{12} + m_{12} \frac{l_{12}^2}{4} - m_{12} \frac{l_{11} l_{12}}{2} \cos \alpha_2 \right] \ddot{\alpha}_1 \\ &\quad - \left[I_{12} + m_{12} \frac{l_{12}^2}{4} \right] \ddot{\alpha}_2 + m_{12} \frac{l_{11} l_{12}}{2} \dot{\alpha}_1^2 \sin \alpha_2. \end{aligned} \tag{43}$$

It is interesting to note that the branch coordinate s does not appear in the expanded form of Equation (41); this is because the torque T and the two bodies do no work for a virtual displacement of joint $e64$, which is not in the path of branches between the two bodies and the ground. This situation would change if a sliding mass was included in the system model, or if one selected a tree that included joints $e64$ and $e63$.

Step 4. To express Equation (41) solely in terms of the independent variation $\delta\alpha_1$, the transformation equation (37) is used to eliminate $\delta\alpha_2$:

$$\begin{aligned} \delta W &= (Q_1 + Q_2 J_1)\, \delta\alpha_1 \\ &= Q_i\, \delta\alpha_1 = 0, \end{aligned} \tag{44}$$

where J_1 is the first row of the matrix $[J]$. One can now set the coefficient of $\delta\alpha_1$ equal to zero, to obtain the single differential equation:

$$Q_i(q, \dot{q}, \ddot{q}) = 0, \tag{45}$$

which can be solved simultaneously with the two constraint equations (36) for the three branch coordinates.

5.2. OPEN KINEMATIC CHAIN: PENDULATING ROTOR

The pendulating rotor shown in Figure 5 is taken from Ginsberg [20]. The system consists of two massless shafts and a uniform disk connected in series by revolute joints D, A, and B. The spin rate $\dot{\phi}$ of the disk relative to the second shaft, of length L, is maintained at a constant value by a servomotor at joint B. Similarly, the precession rate $\dot{\psi}$ about a vertical axis is also kept constant by a servomotor at joint D. We have used DynaFlex to obtain the differential equation governing the

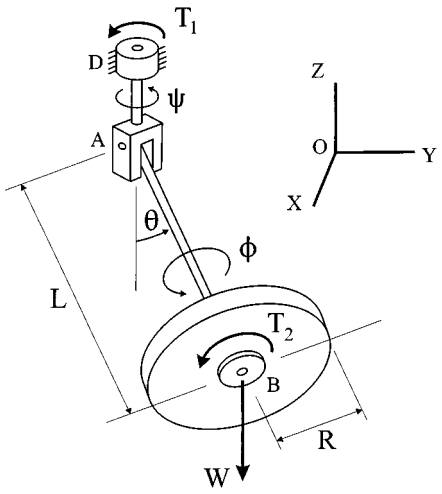

Figure 5. The spinning disk (from Ginsberg [20]).

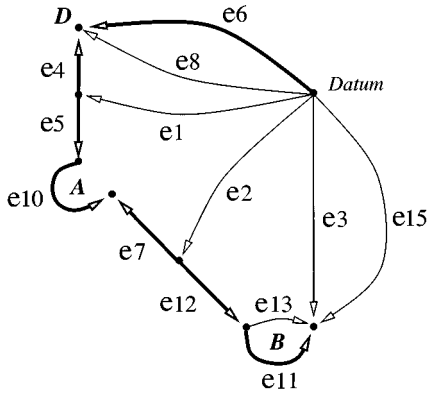

Figure 6. Graph of the spinning disk.

nutation angle θ, and to derive expressions for the torques T_1 and T_2 generated by the two servomotors.

A linear graph of the pendulating rotor is shown in Figure 6. The three rigid bodies are represented by edges $e1$, $e2$, and $e3$, while joints B, A, and D are represented by edges $e11$, $e10$, and $e6$, respectively. Note that the DynaFlex program does not require any particularly numbering of edges. The servomotors at D and B are represented by the motion drivers $e8$ and $e13$, respectively, while the locations of the three joints on the two massless shafts are given by the rigid-arm elements $e4$, $e5$, $e7$, and $e12$. Finally, the weight of the disk is represented by the force driver element $e15$. Once again, the structure of the physical system is visible in the linear graph representation. Note that two dependent virtual work elements for each interbody component are automatically included by the DynaFlex code; for reasons of clarity, they are not shown in Figure 6.

For this open-loop system, all three joints can be selected into the spanning tree, which is shown in bold in Figure 6. From this choice of tree selection, one obtains the set of branch coordinates $\{q\} = [\psi, \theta, \phi]^T$. Since these coordinates are independent, the dynamic equations of motion will consist of three ordinary differential equations, i.e. there are no constraint equations or corresponding Lagrange multipliers.

Note that joint elements $e6$ and $e11$ were selected into the tree instead of the corresponding motion drivers $e8$ and $e13$, in order to obtain explicit expressions for the torques associated with these two servomotors. Had the motion drivers been selected into the tree, a single differential equation in terms of the nutation angle θ would be generated.

The previous topological information is entered into DynaFlex, which generates the equations of motion using a default set of symbolic geometric and inertial parameters. To convert these default parameters into the notation shown in Figure 5, the following data file is read in:

```
> r7z:=-L/2; r12z:=L/2; J31:=(m3*R^2)/4;
> J35:=J31; J39:=(m3*R^2)/2;
> Gamma.8(t):=T.1(t); Gamma.13(t):= T.2(t);
> theta.6(t):=psi(t); theta.10(t):=theta(t);
> theta.11(t):=phi(t);
```

$$r7z := -\frac{1}{2}L, \qquad \Gamma 8(t) := T1(t)$$

$$r12z := \frac{1}{2}L, \qquad \Gamma 13(t) := T2(t)$$

$$J31 := \frac{1}{4}m3\,R^2, \qquad \theta 6(t) := \psi(t)$$

$$J35 := \frac{1}{4}m3\,R^2, \qquad \theta 10(t) := \theta(t)$$

$$J39 := \frac{1}{2}m3\,R^2, \qquad \theta 11(t) := \phi(t)$$

where $J31$, $J35$, and $J39$ are the default DynaFlex symbols for the principal moments of inertia of the disk, and R is the disk radius. Upon substitution of these parameter and variable names, the following equations are obtained:

$$
\begin{aligned}
dyn := \\
&\left[\left(\frac{1}{4}\,m3\,R^2+\frac{1}{4}\,m3\,R^2\cos(\theta(t))^2-m3\,L^2\cos(\theta(t))^2+m3\,L^2\right)\right.\\
&\left(\frac{\partial^2}{\partial t^2}\,\psi(t)\right)-\frac{1}{2}\,m3\,R^2\cos(\theta(t))\left(\frac{\partial^2}{\partial t^2}\,\phi(t)\right)\\
&+\frac{1}{2}\,m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\,\phi(t)\right)\left(\frac{\partial}{\partial t}\,\theta(t)\right)\\
&+\left(-\frac{1}{2}\sin(\theta(t))\,m3\,R^2\cos(\theta(t))+2\,m3\cos(\theta(t))\,L^2\sin(\theta(t))\right)\\
&\left.\left(\frac{\partial}{\partial t}\,\psi(t)\right)\left(\frac{\partial}{\partial t}\,\theta(t)\right)-T1(t)\right]\\
&\left[\left(\frac{1}{4}\,m3\,R^2+m3\,L^2\right)\left(\frac{\partial^2}{\partial t^2}\,\theta(t)\right)\right.\\
&+\left(-m3\cos(\theta(t))\,L^2\sin(\theta(t))+\frac{1}{4}\sin(\theta(t))\,m3\,R^2\cos(\theta(t))\right)\\
&\left(\frac{\partial}{\partial t}\,\psi(t)\right)^2-\frac{1}{2}\,m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\,\psi(t)\right)\left(\frac{\partial}{\partial t}\,\phi(t)\right)\\
&\left.-\mathrm{W}\sin(\theta(t))\left(-\sin(\psi(t))^2\,L-\cos(\psi(t))^2\,L\right)\right]\\
&\left[-\frac{1}{2}\,m3\,R^2\cos(\theta(t))\left(\frac{\partial^2}{\partial t^2}\,\psi(t)\right)+\frac{1}{2}\,m3\,R^2\left(\frac{\partial^2}{\partial t^2}\,\phi(t)\right)\right.\\
&\left.+\frac{1}{2}\,m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\,\theta(t)\right)\left(\frac{\partial}{\partial t}\,\psi(t)\right)-T2(t)\right],
\end{aligned}
$$

where the expressions between brackets [ ] are equal to zero.

As expected, three equations are generated in terms of the branch coordinates ψ, θ, and ϕ, and the driving torques T_1 and T_2 of the two servomotors. Note that DynaFlex leaves the equations in this general form until the exact nature of the motion drivers is specified. In this case, the desired equations are obtained by substituting $\ddot{\psi}=0$ and $\ddot{\phi}=0$ into the preceding general expressions:

```
> dyn1:=map(eval,subs(diff(psi(t),t$2)=0,
> diff(phi(t),t$2)=0, eval(dyn)));
```

$$
\begin{aligned}
&dyn1 := \\
&\left[\frac{1}{2}\,m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\phi(t)\right)\left(\frac{\partial}{\partial t}\theta(t)\right) - T1(t)\right. \\
&+\left(-\frac{1}{2}\sin(\theta(t))\,m3\,R^2\cos(\theta(t)) + 2\,m3\cos(\theta(t))\,L^2\sin(\theta(t))\right) \\
&\left.\left(\frac{\partial}{\partial t}\psi(t)\right)\left(\frac{\partial}{\partial t}\theta(t)\right)\right] \\
&\left[\left(\frac{1}{4}\,m3\,R^2 + m3\,L^2\right)\left(\frac{\partial^2}{\partial t^2}\theta(t)\right)\right. \\
&+\left(-m3\cos(\theta(t))\,L^2\sin(\theta(t)) + \frac{1}{4}\sin(\theta(t))\,m3\,R^2\cos(\theta(t))\right) \\
&\left(\frac{\partial}{\partial t}\psi(t)\right)^2 - \frac{1}{2}\,m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\psi(t)\right)\left(\frac{\partial}{\partial t}\phi(t)\right) \\
&\left.+\,\mathrm{WL}\sin(\theta(t))\right] \\
&\left[\frac{1}{2}\,m3\,R^2\sin(\theta(t))\left(\frac{\partial}{\partial t}\theta(t)\right)\left(\frac{\partial}{\partial t}\psi(t)\right) - T2(t)\right].
\end{aligned}
$$

This set of three equations agrees exactly with those given by Ginsberg [20]. The second equation is the ordinary differential equation governing the nutation angle θ. Given constant values for $\dot{\psi}$ and $\dot{\phi}$, this equation can be solved for $\theta(t)$, after which the servomotor torques at D and B can be found by solving the first and third equations for T_1 and T_2, respectively.

It is noteworthy that, although the masses and moments of inertia for the two shafts are set to zero in the Maple input file, this does not cause any singularity difficulty, as it often does with a purely numerical approach. This certainly is one of the advantages of a symbolic approach over a numerical one.

5.3. SPATIAL SLIDER-CRANK WITH FLEXIBLE CONNECTING ROD

The three-dimensional slider-crank shown in Figure 7 has been previously analyzed by Jonker [22]. It is obtained from a two-dimensional slider-crank (moving in the XOY plane) by rotating the joint axis at O by 45° about the global Y axis, so that the crank OA spins to form a cone whose apex is located at O. Instead of revolute joints at points A and B, as in the planar case, there is a spherical joint at A and a universal joint at B. Crank OA and link AB are initially aligned

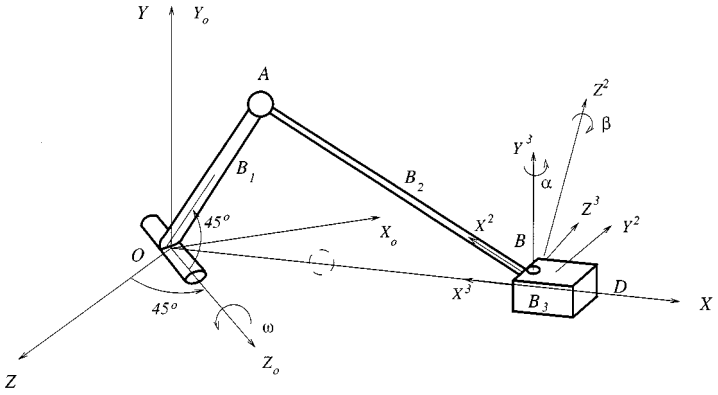

Figure 7. Spatial flexible slider-crank.

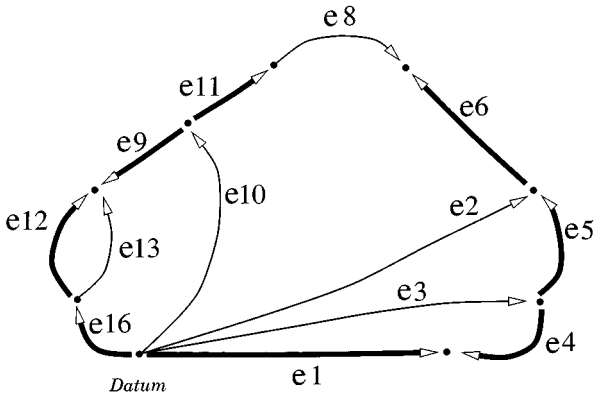

Figure 8. Graph of the spatial flexible slider-crank.

with the global X axis. In this configuration, the link's local body reference frame $BX^2Y^2Z^2$ is coincident with that of the slider, i.e., $BX^3Y^3Z^3$.

The rigid crank has a length of 0.15 m and is rotated at a constant rate of $\omega = 150$ rad/s. The link is flexible, with a length of 0.3 m, a circular cross-sectional area of 2.826×10^{-5} m^2, an area moment of inertia of 6.361×10^{-11} m^4, a Young's modulus of 2.0×10^{11} N/m^2, and a mass density of 7.87×10^3 kg/m^3. The rigid slider has a mass of 3.34×10^{-2} kg, which is half that of the link.

The graph of the system is shown in Figure 8. Numbers assigned to the edges are random, and edges shown in bold comprise the spanning tree. Edges $e9$, $e10$, and $e11$ are for the crank, with $e10$ being the rigid body and $e9$ and $e11$, both rigid arm elements, representing the body-fixed locations of the revolute and spherical joints at O and A, respectively. Edges $e2$ and $e6$ model the flexible link, with flexible arm element $e6$ locating the spherical joint on it and flexible body element $e2$ representing the link. Edge $e3$ models the rigid slider body. The prismatic joint at D is represented by two edges: $e1$ for the translational motion and $e4$ for the constant orientation of the slider relative to the ground. The spherical joint at A

and universal joint at B are represented by joint edges $e8$ and $e5$, respectively. The revolute joint at O is modelled with joint edge $e12$ and rigid arm $e16$, which has zero length but a constant rotation matrix for the orientation of the revolute joint axis relative to the global $OXYZ$ frame. Finally, the input motion to the crank is modelled by a motion driver $e13$, which is in parallel with the revolute joint element $e12$.

The flexible link has been modelled using Euler–Bernoulli beam theory and a complete second-order deformation field [12]. The bending of the link about the Y^2 and Z^2 axes is represented by the deformation variables $v(x,t)$ and $w(x,t)$, while the twist and longitudinal deformation are neglected for this particular example [22]. The two deformation variables are each discretized using three Taylor monomials, for a total of six generalized elastic coordinates $\mathbf{q}_f$.

To obtain the smallest number of constraint equations corresponding to loop closure, the spherical joint edge $e8$ is selected into the cotree. When the corresponding circuit equation is projected onto the reaction space of the joint (i.e. the X, Y, and Z axes), three algebraic constraint equations are obtained. With the other joints in the tree, the final system equations will be in terms of $\mathbf{q_r} = [s_1,\ \alpha_5,\ \beta_5,\ \theta_{12}]$ (the subscripts correspond to edge numbers), in addition to the elastic coordinates $\mathbf{q}_f$ for the link.

With 10 branch coordinates $\{q\} = [\mathbf{q}_r\ ,\ \mathbf{q}_f]^T$ and three constraint equations, the system has seven degrees of freedom. We have used DynaFlex to generate the system equations using the embedding formulation, with θ_{12} selected as an independent generalized coordinate; the elastic coordinates $\mathbf{q}_f$ are automatically chosen to be independent. Ten equations of motion, equal in number to $\{q\}$, are then generated and truncated to second order in the elastic coordinates. The seven dynamic equations and three constraint equations are symbolically generated in the form of (15) and (27) as

$$\begin{cases} [\tilde{M}]\{\ddot{q}\} = \{\tilde{F}(q,\dot{q},T_{13},t)\}, \\ \{\Phi(q,t)\} = 0, \end{cases} \tag{46}$$

in which T_{13}, which appears in only one of the dynamic equations, is the torque required to drive the crank with the prescribed motion.

As in the previous example, these 10 symbolic differential-algebraic equations (46) can be solved for any set of 10 unknown variables. For this problem, the motion of the crank $\theta_{12}(t)$ is specified, and the remaining nine generalized coordinates are to be calculated, along with T_{13}. This was accomplished by numerically integrating the six dynamic equations that do not contain T_{13}, simultaneously with the three constraint equations written at the acceleration level, i.e. in the form of Equation (17). The HHT-α numerical integration method [23] was used, and corrections to the displacements $\{q\}$ and velocities $\{\dot{q}\}$ were computed using Equations (15) and (16), as described in [24]. The driving torque $T_{13}(t)$ was then obtained by substituting the solution for $\{q(t)\}$ back into the remaining dynamic equation.

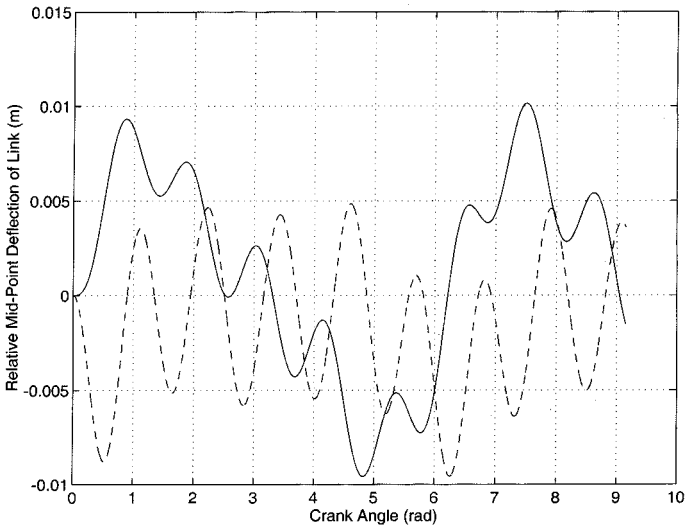

Figure 9. Mid-point link deflection on XOY (—) and XOZ (- - -) planes.

Shown in Figure 9 are the numerical results for the deflection of the mid-point on the link, projected onto the global XOY and XOZ planes, and plotted against the crank angle θ_{12}. The results are in good agreement with those obtained in [22], in which a finite element analysis of the flexible link was employed. From a series of numerical experiments, it was found that this good agreement requires that the inertial forces associated with the flexible link be expanded to at least second order in the elastic coordinates $\mathbf{q}_f$. When the dynamic equations were truncated to first order and solved, this good agreement was lost. A detailed examination of the order of $\mathbf{q}_f$ used in the deformation field for a flexible beam, and the effects of this order on numerical simulation results for flexible multibody systems, is presented in [12].

6. Conclusions

By combining linear graph theory with the principle of virtual work, graph-theoretic modelling methods have been extended to the dynamic analysis of flexible multibody systems. Given a description of a physical system, a linear graph representation can be systematically constructed, from which the governing kinematic and dynamic equations are automatically generated. Using the principle of virtual work, all non-working constraint forces can be eliminated from the equations of motion, if the analyst desires. Alternatively, constraint reactions can be included using Lagrange multipliers. Thus, both the embedding technique and the augmented formulation are encompassed by our graph-theoretic formulation, which was demonstrated using a planar slider-crank mechanism as an example.

A notable feature of this formulation is that the analyst can choose the set of coordinates appearing in the final equations of motion, by simply selecting the

spanning tree for the linear graph. This set can include absolute, joint, and elastic coordinates, or some combination thereof.

Due to the systematic nature of this formulation, it was relatively easy to develop a Maple implementation called DynaFlex that generates the symbolic equations of motion in terms of joint coordinates, using either an augmented formulation or the embedding technique. The analyses of two spatial multibody systems, one open-loop system of rigid bodies and one closed-loop system with a flexible body, were presented to validate both the graph-theoretic formulation and the DynaFlex algorithm.

In the future, it is desirable to extend this formulation to the analysis of mechatronic and other multidisciplinary systems, by again exploiting the attractive features of linear graph theory and virtual work.

Acknowledgement

Financial support of this research by the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

1. McPhee, J.J., 'Automatic generation of motion equations for planar mechanical systems using the new set of "branch coordinates"', *Mechanism and Machine Theory* **33**, 1998, 805–823.
2. McPhee, J.J., 'On the use of linear graph theory in multibody system dynamics', *Nonlinear Dynamics* **9**, 1996, 73–90.
3. Baciu, G. and Kesavan, H.K., 'Graph-theoretic modeling of particle-mass and constrained rigid body systems', *Mechanism and Machine Theory* **30**, 1995, 953–967.
4. Andrews, G.C., Richard, M.J. and Anderson, R., 'A general vector-network formulation for dynamic systems with kinematic constraints', *Mechanism and Machine Theory* **33**, 1988, 243–256.
5. Otter, M., Elmqvist, H. and Cellier, F., 'Modeling of multibody systems with the object-oriented modeling language Dymola', *Nonlinear Dynamics* **9**, 1996, 91–112.
6. Richard, M.J. and Tennich, M., 'Dynamic simulation of flexible multibody systems using vector-network techniques', *Dynamics of Flexible Multibody Systems: Theory and Experiment*, ASME AMD-Vol. 141, 1992, 165–173.
7. Winfrey, R.C., 'Elastic link mechanism dynamics', *Journal of Engineering for Industry* **93**, 1971, 268–272.
8. Erdman, A.G., Sandor, G.N. and Oakberg, R.G., 'A general method for kineto-elastodynamic analysis and synthesis of mechanisms, *Journal of Engineering for Industry* **94**, 1972, 1193–1205.
9. Tennich, M., 'Dynamique de systèmes multi-corps flexibles, une approche générale', Ph.D. Thesis, Université Laval, Québec, 1994.
10. Shabana, A.A., *Dynamics of Multibody Systems*, 2nd edn., Cambridge University Press, Cambridge, 1998.
11. Shi, P. and McPhee, J., 'On the use of virtual work in a graph-theoretic formulation for multibody dynamics', in *Proceedings of ASME Design Engineering Technical Conferences*, Sacramento, CA, 1997, Paper DETC97/VIB-4199.
12. Shi, P., McPhee, J. and Heppler, G., 'A deformation field for Euler–Bernoulli beams with application to flexible multibody dynamics', *Multibody System Dynamics* **5**, 2001, in press.

13. Shi, P., 'Flexible multibody dynamics: A new approach using virtual work and graph theory', Ph.D. Thesis, University of Waterloo, Canada, 1998.
14. Amirouche, F.M.L., *Computational Methods in Multibody Dynamics*, Prentice-Hall, Englewood Cliffs, NJ, 1992.
15. Haug, E.J., *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon, Boston, MA, 1989.
16. Anantharanam, M. and Hiller, M., 'Dynamic analysis of complex multibody systems using methods for differential-algebraic equations', in *Advanced Multibody System Dynamics*, by W. Schiehlen (ed.), Kluwer Academic Publishers, Dordrecht, 1993, 173-194.
17. Oshinowo, O.M. and McPhee, J.J., 'Object-oriented implementation of a graph-theoretic formulation for planar multibody dynamics', *International Journal for Numerical Methods in Engineering* **40**, 1997, 4097–4118.
18. Jerkovsky, W., 'The structure of multibody dynamics equations', *Journal of Guidance and Control* **1**, 1978, 173–182.
19. Wittenburg, J., *Dynamics of Systems of Rigid Bodies*, Teubner, Stuttgart, 1977.
20. Ginsberg, J.H., *Advanced Engineering Dynamics*, 2nd edn., Cambridge University Press, Cambridge, 1995.
21. McPhee, J., Shi, P. and Piedboeuf, J.-C., 'Inverse dynamics of multibody systems using virtual work and symbolic programming', in *Euromech Colloquium on Advances in Computational Multibody Dynamics*, Lisboa, Portugal, 20–23 September, J. Ambrósio and W. Schiehlen (eds), 1999, 801–817.
22. Jonker, B., 'A finite element dynamic analysis of spatial mechanisms with flexible links', *Computer Methods in Applied Mechanics and Engineering* **79**, 1989, 17–40.
23. Hilber, H.H., Hughes, T.J.R. and Taylor, R.L., 'Improved numerical dissipation for time integration algorithms in structural dynamics', *Earthquake Engineering and Structural Dynamics* **5**, 1977, 283–292.
24. Shi, P., McPhee, J. and Heppler, G., 'Polynomial shape functions and numerical methods for flexible multibody dynamics', *Mechanics of Structures and Machines*, 1999, submitted.