

Dynamics of Multibody Systems: Conventional and Graph-Theoretic Approaches

by

Professor John J. McPhee, P.Eng.
Systems Design Engineering
(Cross-appointed to Mechanical Engineering)
University of Waterloo, Ontario
Canada N2L 3G1

Contents

1	Introduction	1
2	Basic Concepts of Graph-Theoretic Modelling	2
2.1	Linear Graph Terminology	2
2.2	Topological Equations	3
2.2.1	Vertex Postulate	3
2.2.2	Cutset Equations	4
2.2.3	Circuit Postulate and Equations	5
2.3	Orthogonality	6
2.4	Constitutive Equations	7
2.5	Formulation and Solution of System Equations	9
3	One-Dimensional Physical Systems	11
3.1	Electrical Network	11
3.2	Hydraulic Network	13
3.3	Mechanical System	15
3.3.1	Hybrid Branch-Chord Formulation	16
3.3.2	Alternative Formulation	17
4	Two-Dimensional Mechanical Systems	20
4.1	Linear Graph Representation	20
4.2	Terminal Equations	22
4.3	Topological Equations	24
4.4	Tree and Coordinate Selection	25
4.5	Graph-Theoretic Formulation Procedure	27
4.6	Examples	31
4.6.1	Dynamic Equations for Double Pendulum	31
4.6.2	Kinematic Equations for Quick-Return Mechanism	34
4.6.3	Dynamic Equations for Slider-Crank Mechanism	37
4.6.4	Dynamic Analysis of Squeezer Mechanism	39
	References	42

Notation

Symbol	Significance
a	general scalar variable
v	number of vertices (nodes)
e	number of edges
b	number of branches (tree)
c	number of chords (cotree)
n	number of generalized coordinates
m	number of constraint equations
f	number of degrees of freedom
$\{a\}$	column matrix of scalars
\underline{a}	general vector quantity
$\{\underline{a}\}$	column matrix of vectors
\hat{u}	unit vector
$\hat{i}, \hat{j}, \hat{k}$	unit vectors parallel to x, y, z axes
$[a]$	general matrix
$[0]$	null matrix
$[1]$	unit matrix
$[I]$	incidence matrix
$[A]$	reduced incidence matrix
$[A_f]$	fundamental cutset matrix
$[B_f]$	fundamental circuit matrix
τ	general “through variable”
α	general “across variable”
α_n	general “nodal variable”

1 Introduction

The Graph-Theoretic Method (GTM) is a simple, organized technique for creating mathematical models of discrete physical systems. GTM combines *linear graph theory* [1], which derives from Topology (the study of how things are interconnected – a branch of the mathematical field of Combinatorics), with the physical characteristics of engineering components. The resulting technique is ideally suited for computer creation and solution of the mathematical models of physical systems. More specifically, GTM provides algorithms for automatically generating the differential-algebraic equations governing the dynamic behaviour of a wide range of physical systems.

The theoretical basis of GTM began in the 1700s with the topological work of Leonhard Euler [2]; however, the elaboration of GTM as a unifying “system theory” and the application to engineering systems was first defined clearly in the text by Koenig, Tokad and Kesavan [3]. Many diverse research projects have extended that basic work into a number of different fields and applications [4].

The characteristic of GTM that distinguishes it from more traditional formulation methods is that the equations in the mathematical model that describe the interconnection of system elements are derived separately from the equations that describe their physical characteristics. This means that once a basic set of topological terms have been mastered, the Graph-Theoretic Method can be very easily understood and applied to systems with widely-different characteristics. In fact, GTM has been applied to a broad range of very different types of systems: electrical circuits, fluid flow networks, mechanical and mechatronic systems, and even socio-economic systems.

The purpose of this document is to describe the application of GTM to the modelling of multi-dimensional mechanical systems, otherwise known as “multibody systems”. The goal of multibody dynamics researchers is to develop algorithms that automatically generate the equations of motion, given only a description of the system as input. Such algorithms, which have been developed into a number of commercial software packages [5], relieve a dynamicist of the tedious and error-prone task of deriving these equations by hand. Furthermore, by incorporating numerical methods to generate approximate solutions to these nonlinear differential-algebraic equations (DAEs) of motion, powerful tools for computer-aided design have been obtained.

GTM is ideally suited for the development of multibody dynamics formulations. By separating the system topological equations from the component constitutive equations, very efficient formulations and computer algorithms can be obtained. The explicit representation of topology provides physical insight into the structure of the DAEs of motions. Furthermore, there is no need to pre-define a set of generalized coordinates, as is done in all conventional multibody formulations. In addition, both matrix and recursive formulations can be derived using GTM. As a result, GTM provides a general and unifying approach to the field of multibody dynamics.

In the next section, the basic terminology and concepts of GTM are introduced and explained. This is followed by a number of applications of GTM to physical systems that are essentially one-dimensional in nature, i.e. their response can be represented by simple scalar variables. In the subsequent section, GTM is extended to the kinematic and dynamic analysis of two-dimensional mechanical systems, for which vectorial variables are employed. A number of different examples are also presented. Some advanced topics are then introduced, including the application of GTM to mechatronic and three-dimensional multibody systems, and the development of highly-efficient recursive algorithms for robotic manipulators. This document concludes with a summary of GTM and the advantages of using it to model mechanical systems.

2 Basic Concepts of Graph-Theoretic Modelling

2.1 Linear Graph Terminology

A linear graph G is a collection of e edges which intersect only at *nodes* (or *vertices*). These edges can be represented pictorially by lines¹, and the v nodes at which they connect can be represented by points. The topology of a linear graph is defined when one specifies which edges are incident upon (connected to) each and every node. This topology can be written in a simple mathematical form using the *incidence matrix*, defined in Section 2.2.

The graph G is said to be connected if every node can be reached from any other node by traversing a sequence of edges (and nodes) called a *path*. If G is not connected, then it exists in *parts* where each part is a connected subgraph of G . Two paths are said to be distinct if the only nodes common to them are their end nodes. A *circuit* is defined as a subgraph of G such that there are exactly two distinct paths between every pair of nodes of the subgraph. A *tree* of a connected graph G is defined as any subgraph that: (1) is connected; (2) contains all the nodes of G ; and (3) has no circuits. An important feature of a tree is that it has exactly one path between every pair of nodes and has exactly $(b = v - 1)$ edges. A *cotree* is defined as the subgraph of G that remains after deleting the edges of a tree, and hence there are $(c = e - v + 1)$ edges in the cotree. The edges in the tree and its cotree are called *branches* and *chords*, respectively. A *fundamental circuit* is a circuit consisting of a single chord, and a unique set of branches; there is one f-circuit for each chord of the graph.

To illustrate some of these concepts, and to demonstrate the use of linear graphs in representing the topology of a physical system, consider the electrical network shown in Figure 1(a). It consists of a voltage source E_1 , four resistors R_{2-5} , and a current source I_6 . Figure 1(b) shows the linear graph which is topologically equivalent (isomorphic) to the electrical network. This graph is constructed by drawing a node for each point at which two elements connect, and by replacing these elements with directed edges on a one-to-one basis. The direction (arrow) assigned to each edge is arbitrary and serves only to provide a reference direction for the measurement of physical variables.

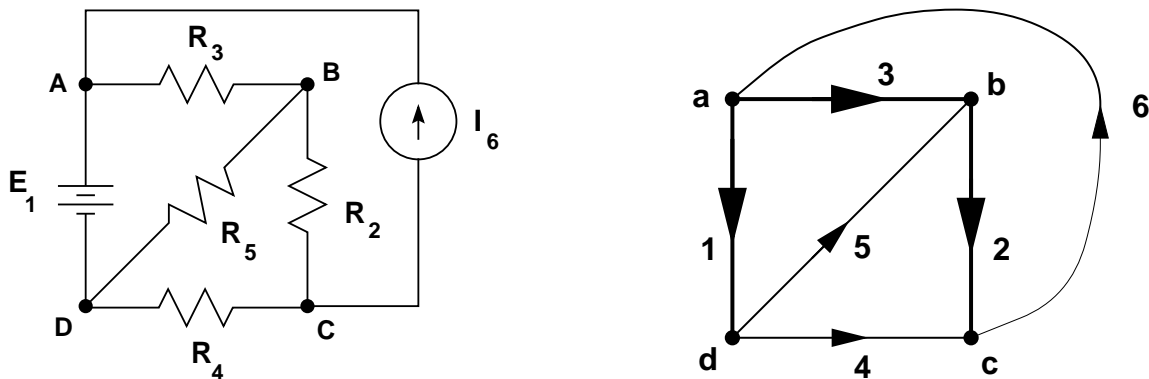


Figure 1: Electrical Network (a) and Isomorphic Linear Graph (b)

¹This pictorial representation by lines is the reason for the term *linear* graph — it has nothing to do with the nature of the equations generated by a graph-theoretic approach, which may be highly nonlinear as shown in the examples.

In this example, the edges 1, 2, and 3 have been selected as branches of the tree (and drawn as bold lines), while the remaining edges 4, 5, and 6 comprise the chords of the cotree. Note that there are many possible trees for a linear graph; for example, the graph in Figure 1(b) has 15 different trees (and 15 associated cotrees).

Another important subgraph is the *cutset*, defined as a subset of edges that, when removed, divides the connected graph into two parts, and no subset of this subset has this property. For the example in Figure 1(b), the subset 3, 4, 5, and 6 is a cutset since its removal leaves the graph in two parts where part 1 is edge 1 (and its nodes) and part 2 is edge 2 (and its nodes), and no subset of (3,4,5,6) has this property. A *fundamental cutset* is a cutset consisting of one branch, and a unique set of chords; there is one f-cutset for each branch in the linear graph.

2.2 Topological Equations

As shown in the previous section, linear graphs can be used to encapsulate the topology of a physical system. In this section, linear graph theory is used to create mathematical models of engineering systems. To accomplish this, it is necessary to introduce a set of physical quantities known as the *through* and *across variables*.

A through variable τ is associated with each edge of a linear graph, and corresponds to a quantity that would be measured by an instrument in series with the physical element represented by the edge. For an electrical network, a suitable through variable would be the current in each element. In a hydraulic network, it might be a flow rate, while in a mechanical system it could be a force or torque.

An across variable α is also associated with an edge of a graph, and corresponds to a quantity measured by an instrument placed in parallel with the physical element represented by the edge. This might be the voltage in an electrical element, pressure differential across a hydraulic component, or the displacement, velocity, or acceleration of a mechanical element. Note that derivatives and integrals of through and across variables are themselves through and across variables, respectively. It is also important to note that through and across variables can be scalars, vectors, or tensors of even higher order.

Finally, it is occasionally convenient to introduce an auxiliary set of variables known as the *nodal variables*. Essentially, a nodal variable α_n corresponds to an across variable measurement between a given node and a reference or “datum” node. In an electrical system, the nodal variables correspond to the voltage differences between the nodes and the datum node. The nodal variables in a hydraulic system would be pressures relative to the ground node, while in a mechanical system the nodal variables would correspond to the displacements, velocities, or accelerations measured by an observer in the datum node. Note that a linear graph with e elements and v nodes will have e through and across variables, and $v - 1$ nodal variables.

2.2.1 Vertex Postulate

The above definition of a through variable guarantees that *the sum of through variables at any node of a linear graph must equal zero when due account is taken of the orientation of edges incident upon that node*, regardless of the type of physical system represented by the graph. This general statement about through variable continuity is known as the “Vertex Postulate”.

Physically, the Vertex Postulate is equivalent to Kirchoff's Current Law if the system under study is an electrical circuit. In the case of a hydraulic network, this Postulate represents the conservation of mass at any junction in the system. For a mechanical system, the Vertex Postulate corresponds to the equilibrium of forces at any point.

The Vertex Postulate can be written in mathematical form for all v nodes by multiplying the $v \times e$ "incidence matrix" by the column matrix $\{\tau\}$ containing all e through variables, and setting the result to zero:

$$[I]\{\tau\} = \{0\} \quad (1)$$

which gives one scalar equation for each node in the system. Note that the topology of the system is embedded in the incidence matrix $[I]$, which is constructed directly from the linear graph as follows²:

$$I_{jk} = \begin{cases} 0 \\ +1 \\ -1 \end{cases} \text{ if edge } k \text{ is } \begin{cases} \text{not incident on} \\ \text{incident on and away from} \\ \text{incident on and towards} \end{cases} \text{ the } j^{\text{th}} \text{ node.}$$

As an example, the incidence matrix corresponding to the linear graph in Figure 1(b) is given by:

$$[I] = \begin{matrix} & & E_1 & R_2 & R_3 & R_4 & R_5 & I_6 \\ \begin{matrix} a \\ b \\ c \\ g \end{matrix} & \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} & & & & & & \end{matrix} \quad (2)$$

If one were to use this incidence matrix in equation (1), the corresponding column matrix of through variables would be:

$$\{\tau\} = [i_1, i_2, i_3, i_4, i_5, i_6]^T \quad (3)$$

where i_j is the current flowing through element j .

As shown in Section 2.4, there are several mathematical formulations or derivations of system equations, all of which begin with the creation of the incidence matrix.

2.2.2 Cutset Equations

It is important to note that the equations shown in (1) are not linearly independent, since the rank of $[I]$ is only $b = v - 1$. To avoid problems during the formulation of system equations, it is useful to replace the incidence matrix in equation (1) with the *reduced incidence matrix* $[A]$, which is obtained by deleting the row of $[I]$ corresponding to the datum node:

$$[A]\{\tau\} = \{0\} \quad (4)$$

which are known as the "cutset equations", since they represent the set of b linearly independent cutsets that isolate the remaining $v - 1$ nodes. For the electrical network example, $[A]$ would be obtained by deleting the last row of the incidence matrix in (2) corresponding to the datum node g .

²Note that several authors use the opposite of this convention; however, the final set of equations obtained through elementary row operations will be the same in either case.

Just like the Vertex Postulate, the cutset equations represent a balance of through variables at any node in the system. However, one might equally consider the balance of through variables for any portion of the system graph left isolated by a cutset — this subgraph can be more than just an individual node. If one expresses the balance of through variables for the subgraphs left isolated by the fundamental cutsets, an alternative set of b linearly independent *f-cutset equations* can be obtained.

Mathematically, the f-cutset equations are written in the form:

$$[A_f]\{\tau\} = \{0\} \quad (5)$$

where the f-cutset matrix $[A_f]$ is obtained by applying Gauss-Jordan elimination to $[A]$ to obtain the row-reduced echelon form:

$$[A_f] = \left[[1_b] \ [A_c] \right] \quad (6)$$

where $[1_b]$ is a $b \times b$ unit matrix, and $[A_c]$ is the remaining $b \times c$ submatrix. Note that the b leftmost columns of $[A]$ correspond to the branches in the tree, which define the fundamental cutsets.

The matrix $[A_f]$ can also be generated from a visual inspection of the f-cutsets for the linear graph. Specifically,

$$A_{f,jk} = \begin{cases} 0 \\ -1 \\ +1 \end{cases} \text{ if edge } k \text{ is } \begin{cases} \text{not in the cutset for} \\ \text{in the cutset and oriented against} \\ \text{in the cutset and oriented with} \end{cases} \text{ the } j^{\text{th}} \text{ branch.}$$

Recall that each fundamental cutset separates the graph into two distinct parts. As seen above, the orientation of the single branch which connects the two parts defines the reference direction for all the chords in the f-cutset. As an example, the f-cutset matrix $[A_f]$ for the system in Figure 1 takes the form:

$$[A_f] = \begin{bmatrix} 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 & -1 \end{bmatrix} \quad (7)$$

However, it is generally preferable to automate the generation of the f-cutset matrix (and equations) by using a computer to apply Gauss-Jordan elimination to the reduced incidence matrix, once it has been assembled from the linear graph.

2.2.3 Circuit Postulate and Equations

The second major postulate of linear graph theory states that *the sum of across variables around any circuit of a graph must equal zero when due account is taken of the direction of edges in the circuit*. This general statement of across variable compatibility is known as the *Circuit Postulate*, which holds for any physical system represented by a linear graph.

For an electrical circuit, the Circuit Postulate takes the familiar form of Kirchoff's Voltage Law. However, it also applies to hydraulic networks, in which case it guarantees that the sum of pressure differences around any closed loop must equal zero. For a mechanical system, the Circuit Postulate corresponds to the summation of vector displacements (or velocities or accelerations) around any closed kinematic chain.

If the Circuit Postulate is applied to the c fundamental circuits, where there is a unique f-circuit for each chord, a set of linearly independent equations can be generated. Mathematically, these can be written in the matrix form:

$$[B_f]\{\alpha\} = \{0\} \quad (8)$$

where the f-circuit matrix $[B_f]$ can be generated from a visual inspection of the f-circuits for the linear graph. Specifically,

$$B_{f_{jk}} = \begin{cases} 0 \\ -1 \\ +1 \end{cases} \text{ if edge } k \text{ is } \begin{cases} \text{not in the circuit for} \\ \text{in the circuit and oriented against} \\ \text{in the circuit and oriented with} \end{cases} \text{ the } j^{\text{th}} \text{ chord.}$$

Essentially, the single chord defines a positive direction for travelling around the f-circuit; if a branch in the circuit is oriented in this same reference direction, the corresponding entry in $[B_f]$ is +1.

Using this procedure to construct $[B_f]$, one obtains the following matrix:

$$[B_f] = \begin{bmatrix} [B_b] & [1_c] \end{bmatrix} \quad (9)$$

where $[1_c]$ is a $c \times c$ unit matrix corresponding to the unique chords in the f-circuits, and $[B_b]$ is the remaining $c \times b$ submatrix. As an example, the f-circuit matrix for the graph in Figure 1 is given by:

$$[B_f] = \begin{bmatrix} 1 & -1 & -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

and the corresponding column matrix of across variables is:

$$\{\alpha\} = [v_1, v_2, v_3, v_4, v_5, v_6]^T \quad (11)$$

where v_i is the voltage across element i .

2.3 Orthogonality

A more direct approach to calculating the f-circuit matrix is obtained from the important fact that the rows of $[A_f]$ and $[B_f]$ are orthogonal:

$$[A_f][B_f]^T = [0] \quad (12)$$

which represents a general conservation principle, applicable to any type of physical system represented by a linear graph. This Principle of Orthogonality [6] can be used to derive Tellegen's Theorem or the Principle of Virtual Work for mechanical systems. Substituting equations (6) and (9) into equation (12) and re-arranging, the following expression is obtained:

$$[B_b] = -[A_c]^T \quad (13)$$

which can be used to automatically generate the f-circuit equations once the f-cutset matrix has been computed.

Finally, in those situations when it is useful to employ the nodal variables, they can be related to the set of across variables by the nodal transformation equation:

$$\{\alpha\} = [A]^T \{\alpha_n\} \quad (14)$$

which can be derived [7] by writing the f-circuit equations for the original graph, supplemented by a ‘‘Lagrangian tree’’ consisting of edges drawn from each node to the common datum node.

2.4 Constitutive Equations

Together, the cutset and circuit equations constitute a set of $b+c = e$ linear equations in terms of the through and across variables. Since there are two of these variables for each edge in the graph, there is a total of $2e$ unknown quantities. To obtain a necessary and sufficient set of equations for solving for all $2e$ unknowns, the topological equations derived in the previous section are supplemented by the e constitutive or *terminal* equations that relate the through and across variables for each physical component.

Essentially, the terminal equation represents the physical behaviour that characterizes a component of a particular type, and can only be determined through experimentation. By carefully measuring the through and across variables for a given element under a variety of conditions, a functional relationship can usually be established. This relationship might be conveniently expressed in one of the following forms:

$$\alpha = \alpha(t) \implies \text{across driver} \quad (15)$$

$$\tau = \tau(t) \implies \text{through driver} \quad (16)$$

$$\alpha = \mathcal{F}(\tau) \implies \text{resistive element} \quad (17)$$

$$\tau = \mathcal{G}(\alpha) \implies \text{conductive element} \quad (18)$$

$$\mathcal{H}(\alpha, \tau, t) = 0 \implies \text{hybrid element} \quad (19)$$

Note that, unlike the topological cutset and circuit equations, the terminal equations for a component may be highly nonlinear differential-algebraic equations.

Electrical Components. For electrical networks, the terminal equations that relate the voltages v across and currents i through a number of common linear components are given in Table 1. The symbols R , C , and L represent the resistance, capacitance, and inductance of the corresponding elements. A more comprehensive set of terminal equations for both linear and nonlinear electrical components is given in [7]. Note that a positive current i flows in the same direction as the edge, while a positive v represents a voltage *drop* across the edge.

Hydraulic Components. For hydraulic systems, the across and through variables for an element are the pressure drop P and volumetric flow rate q , respectively, with the product Pq representing hydraulic power. Alternatively, hydraulic head or elevation can be used as an across variable in place of pressure. Table 2 lists the terminal equations for a variety of hydraulic components.

A constant head source is modelled as an across or pressure driver. In some cases, an overhead tank of large capacity may be considered as a constant head pressure driver. If changes in its head are not negligible, then a simple reservoir is represented by the capacitor-like form shown in Table 2, where C is a function of the ratio of cross-sectional area to specific gravity.

Element	Terminal Equation
Voltage Source	$v = v(t)$
Current Source	$i = i(t)$
Resistor	$v = R i$
Capacitor	$i = C \frac{dv}{dt}$
Inductor	$v = L \frac{di}{dt}$

Table 1: Terminal Equations for Common Electrical Components

Element	Terminal Equation
Pressure Driver	$P = P(t)$
Simple Reservoir	$q = C \frac{dP}{dt}$
Flow Driver	$q = Q_n(t)$
Hydraulic Resistance	$P = R q^k$
Hydraulic Inertance	$P = L \frac{dq}{dt}$

Table 2: Terminal Equations for Common Hydraulic Components

A pump is modelled as a through or flow driver of known flow rate $Q_n(t)$, where n is a node. A consumption is also modelled as a flow driver. The arrow on the corresponding edge denotes the direction of flow.

A pipe is modelled as a hydraulic resistance. The resistance R is a function of the pipe length l and roughness coefficient f (based on the material and age of the pipe), and an inverse function of the diameter d . The exponent k in Table 2 is a constant, where usually $1 \leq k \leq 2$. The relationship of P to q is linear ($k = 1$) in the laminar flow region (Reynold’s number $< 21,000$), and nonlinear ($k \neq 1$) in the turbulent region.

Mechanical Components. As shown in Table 3, one-dimensional mechanical systems may be either translational or rotational in nature. Translational components might be used to model a vibrating platform, while the rotational components would be included in a model of a rotating shaft system.

Note that for either physical domain, rotational or translational, the rigid body terminal equations are written in the d’Alembert form. This is because the force (torque) *through* the body is equal and opposite to the resultant force (torque) acting on the body. For each translational component, there is an analogous rotational component. In either case, it is necessary to establish a positive coordinate axis for the measurement of displacements.

Multi-Terminal Components. Finally, it should be noted that some components (e.g. electrical transformer or transistor, hydraulic reservoir with more than one exit) cannot be represented by a single edge because the component is connected to the rest of the system at more than 2 locations. If v_m nodes are required to represent the terminals of a *multi-terminal component*, it can be shown that the component can be represented by $v_m - 1$ edges connecting these nodes in a tree-like

<i>Translational</i>		<i>Rotational</i>	
Element Type	Equation	Element Type	Equation
Rigid Body	$F = -m\ddot{x}$	Rigid Body	$T = -I\ddot{\theta}$
Position Driver	$x = x(t)$	Rotation Driver	$\theta = \theta(t)$
Spring	$F = -k(x - l_0)$	Spring	$T = -K(\theta - \theta_0)$
Damper	$F = -d\dot{x}$	Damper	$T = -D\dot{\theta}$
Applied Force	$F = F(t)$	Applied Torque	$T = T(t)$

Table 3: Terminal Equations for One-Dimensional Mechanical Components

topology. The terminal equations for this multi-terminal component are then expressed in terms of the $v_m - 1$ through and across variables.

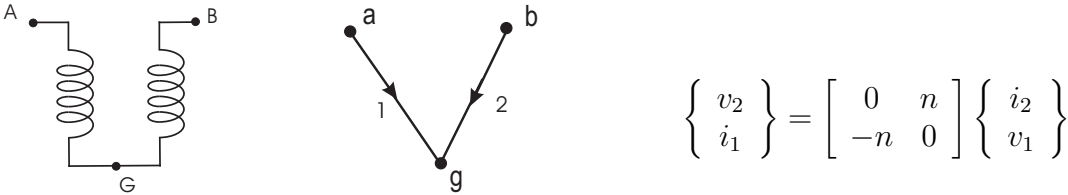


Figure 2: Ideal Transformer, Graph, and Terminal Equations

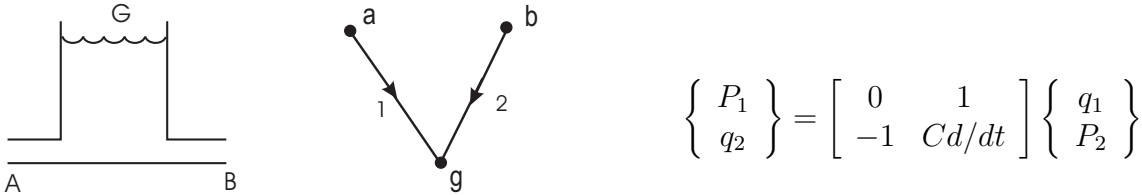


Figure 3: Hydraulic Reservoir, Graph, and Terminal Equations

An electrical transformer and a hydraulic reservoir with two outlets are examples of three-terminal components, and are shown with their terminal equations and graphs in Figures 2 and 3. In the case of the ideal transformer, one can show that the products i_1v_1 and i_2v_2 are equal, thereby establishing that power is conserved. Note that n represents the turn ratio, which is a parameter of the transformer. The linear graph model of the hydraulic reservoir with two outlets provides one differential equation for the pressure in the tank and accounts for the two outflows, which can be verified by writing out the equations in matrix form.

2.5 Formulation and Solution of System Equations

As seen in the previous section, a total of $2e$ equations are obtained when the cutset, circuit, and terminal equations are assembled for a given physical system. These equations form a necessary

and sufficient set for solving for the $2e$ through and across variables. If the b nodal variables are introduced for expediency (e.g. one or more terminal equations are most easily expressed in terms of nodal variables), then the number of unknowns increases to $2e + b$. However, the e nodal transformation equations (14) can be used to supplement the b vertex equations (4) and e terminal equations, thereby providing a sufficient set in what is known as a *nodal formulation*. Although a larger number of equations have to be solved from a nodal formulation, it does have the advantage that a tree is not required to generate the system equations.

On the other hand, the selection of a tree can be used to substantially reduce the number of equations to be solved. In a *branch-chord* formulation of the system equations, equations (5) and (6) are used to express the through variables for branches in terms of the chord through variables:

$$\{\tau_b\} = -[A_c]\{\tau_c\} \quad (20)$$

This *chord transformation* equation can then be used to eliminate the branch through variables from the set of system equations. In a similar fashion, equations (8) and (9) can be used to express the chord across variables in terms of the across variables for tree elements:

$$\{\alpha_c\} = -[B_b]\{\alpha_b\} \quad (21)$$

which are known as the *branch transformation* equations. By substituting equations (20) and (21) into the system terminal equations, a set of e equations in terms of only e unknown variables is obtained. These unknown variables are the b branch across variables and the c chord through variables, which can effectively be reduced further in number by selecting across drivers into the tree and through drivers into the cotree.

Another alternative for automatically generating the system equations is one of the *tableau methods* of formulation [7]. In these approaches, the terminal equations are assumed to take a variety of standard forms, which are assembled with the cutset and circuit equations using a “stamping” procedure that is similar to the finite element assembly process. The result is a relatively compact set of equations that are ready for subsequent analytical or numerical solution.

The actual solution method used will depend upon the time-domain nature of the analysis, and the functional form of the terminal equation. If a time-invariant or steady-state analysis is required, then the system equations will be algebraic in nature. Furthermore, they will be linear if all terminal equations are linear, thereby admitting a closed-form solution to be found. If one or more terminal equations are nonlinear, then the system equations will also be nonlinear and will not in general admit an analytical solution. In this case, an iterative method such as the Newton-Raphson procedure can be used to numerically generate an approximate solution.

If the response of the system is time-varying, then the governing equations will usually comprise a set of differential-algebraic equations (DAEs). If they are linear (i.e. if all terminal equations are linear), then the set of DAEs can be reduced to a smaller set of ordinary differential equations (ODEs), which can be solved using Laplace transforms or any other equivalent method for linear systems of ODEs. If the DAEs are nonlinear in nature, then numerical methods of analysis must be used to generate an approximate solution for the time response of the system.

3 One-Dimensional Physical Systems

3.1 Electrical Network

Shown in Figure 4(a) is a simple electrical network consisting of a voltage source $E_1(t)$, two resistors R_{2-3} , and a capacitor C_4 . This electrical network example is taken from the User's Manual [8] for the software package **MS-1**.

The corresponding system graph, shown in Figure 4(b), has an edge for each of these four components, and three nodes that represent their points of connection. From this graph, the incidence matrix is assembled:

$$[I] = \begin{matrix} & E_1 & R_2 & R_3 & C_4 \\ \begin{matrix} a \\ b \\ g \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 1 \\ -1 & 0 & -1 & -1 \end{bmatrix} \end{matrix} \quad (22)$$

from which the reduced incidence matrix is obtained by deleting the row corresponding to the ground node g :

$$[A] = \begin{matrix} & E_1 & R_2 & R_3 & C_4 \\ \begin{matrix} a \\ b \end{matrix} & \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 1 \end{bmatrix} \end{matrix} \quad (23)$$

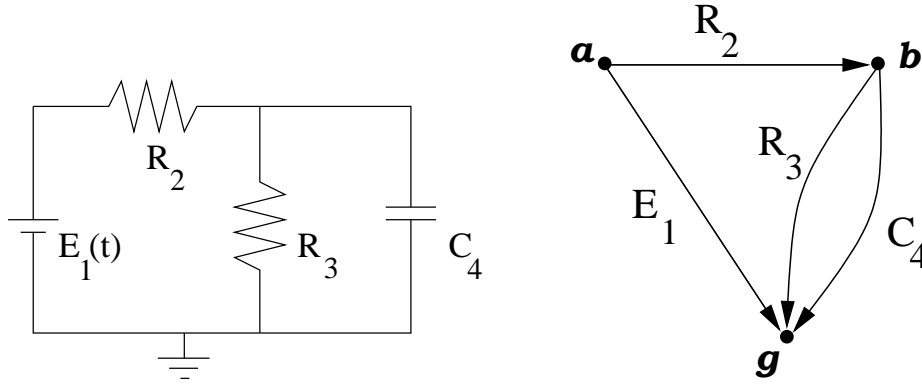


Figure 4: Electrical Network (a) and System Graph (b)

At this point, a nodal formulation can be used to assemble the following group of expressions:

$$\{v\} = [A]^T \{v_n\} \quad (24)$$

$$[A]\{i\} = \{0\} \quad (25)$$

where the nodal variables correspond to the voltages at a and b , relative to the datum node g :

$$\{v_n\} = [v_a, v_b]^T$$

and the terminal equations relating the across variables (voltages) $\{v\}$ to the through variables (currents) $\{i\}$ are assumed to be of the linear form shown in Table 1:

$$v_1 = E_1(t) \quad (26)$$

$$v_2 = R_2 i_2 \quad (27)$$

$$v_3 = R_3 i_3 \quad (28)$$

$$i_4 = C_4 \dot{v}_4 \quad (29)$$

subject to the initial condition $v_4(0) = v_{40}$. Equations (24-29) constitute a set of 10 linear differential-algebraic equations that can be solved for the 8 through and across variables, and the 2 unknown nodal variables

The number of equations to be solved simultaneously can be reduced by selecting a tree and using a branch-chord formulation, as described in Section 2.4. Choosing E_1 and C_4 as branches results in the following fundamental cutset matrix:

$$[A_f] = \begin{matrix} & E_1 & C_4 & R_3 & R_2 \\ a & \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix} & & & \\ b & & & & \end{matrix} = \begin{bmatrix} [1_b] & [A_c] \end{bmatrix} \quad (30)$$

which can be used to write the chord transformation equations (20) for this specific system:

$$\begin{Bmatrix} i_1 \\ i_4 \end{Bmatrix} = - \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix} \begin{Bmatrix} i_3 \\ i_2 \end{Bmatrix} \quad (31)$$

Similarly, making use of equation (13), the branch transformation equations (21) can be written explicitly as:

$$\begin{Bmatrix} v_3 \\ v_2 \end{Bmatrix} = - \begin{bmatrix} 0 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_4 \end{Bmatrix} \quad (32)$$

Substituting equations (31) and (32) into the terminal equations (26-29),

$$\mathcal{H}(\{v_b\}, \{i_c\}, t) = \begin{Bmatrix} v_1 - E_1(t) \\ (v_1 - v_4) - R_2 i_2 \\ v_4 - R_3 i_3 \\ (i_2 - i_3) - C_4 \dot{v}_4 \end{Bmatrix} = 0 \quad (33)$$

a set of only 4 equations to be solved for the 2 tree voltages and the 2 cotree currents, after which the remaining through and across variables are obtained by back-substitution into equations (31) and (32).

Note that the selection of the voltage source into the tree effectively reduces by one the number of simultaneous equations to be solved. In addition, the linear nature of the resistor elements allows their currents to be expressed analytically in terms of the remaining unknown variable, v_4 :

$$i_2 = \frac{E_1(t) - v_4}{R_2} \quad (34)$$

$$i_3 = \frac{v_4}{R_3} \quad (35)$$

which are substituted into the last of equations (33) to obtain:

$$C_4 \dot{v}_4 + \left(\frac{1}{R_3} + \frac{1}{R_2} \right) v_4 = \frac{E_1(t)}{R_2} \quad (36)$$

a single, linear, ordinary differential equation that can be solved for $v_4(t)$, subject to the given initial condition, from which all other variables are found by back-substitution.

Before leaving this example, it is important to note that reduction of the number of branch-chord equations to less than e is made possible by an intelligent selection of tree and cotree elements, and by exploitation of the particular nature of their terminal equations.

3.2 Hydraulic Network

A nonlinear water distribution network is an example of a hydraulic system containing pipes, an overhead storage tank, a pump, and consumption at a node, as shown in Figure 5(a). We will assume that there are no elevation differences between nodes of the system. Because of nonlinearities in pipe characteristics, the system yields nonlinear differential-algebraic equations.

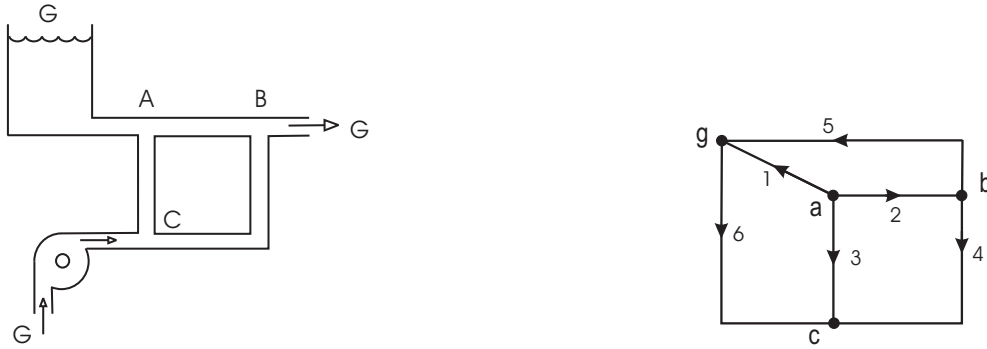


Figure 5: Pipe Network (a) and System Graph (b)

The linear graph for the system in Figure 5(a) is shown in Figure 5(b). The top of the tank is assumed to be open to the atmosphere, and considered the same as the datum or reference node g ; the edge representing the tank is drawn from the tank (node a) to g . We note that the tanks, pumps, and consumptions are generally connected from a node to the reference node, whereas a pipe may be connected between any pair of nodes. The system in Figure 5 does not have across drivers or inertance components. A nodal formulation method, with additional substitutions, will be used for this example.

We begin with the reduced incidence matrix and the vertex equations, where we order the edges starting with the the tank, followed by the pipe edges, and then the pumps and consumptions:

$$\begin{matrix} a \\ b \\ c \end{matrix} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & -1 & 0 & -1 \end{bmatrix} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{Bmatrix} = \{0\} \quad (37)$$

The terminal equations are:

$$q_1 = C_1 \frac{dP_1}{dt} \quad (38)$$

$$q_i = R_i P_i^{1/k}, \quad i = 2, 3, 4 \quad (39)$$

$$q_5 = Q_b, \text{ consumption} \quad (40)$$

$$q_6 = Q_c, \text{ pump} \quad (41)$$

Substituting the terminal equations for each component into (37), we obtain the following. The nonlinear pipe equations are represented by *, where $q = R * P$ is $q = RP^{1/k}$. The tank gives rise to a differential equation, which is kept separately:

$$\begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix} \begin{Bmatrix} R_2 * P_2 \\ R_3 * P_3 \\ R_4 * P_4 \end{Bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \{q_1\} = \begin{Bmatrix} 0 \\ -Q_b \\ Q_c \end{Bmatrix} \quad (42)$$

and $C_1 \dot{P}_1 = q_1$.

From the nodal transformation equations, we obtain the following subset, where P_a , P_b , and P_c are the nodal pressures:

$$\begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{Bmatrix} P_a \\ P_b \\ P_c \end{Bmatrix} \quad (43)$$

Substituting for the pipe across variables $\{P\}$ in terms of nodal pressures,

$$\begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{bmatrix} \begin{Bmatrix} R_2 * \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{Bmatrix} P_a \\ P_b \\ P_c \end{Bmatrix} \\ R_3 * \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{Bmatrix} P_a \\ P_b \\ P_c \end{Bmatrix} \\ R_4 * \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \begin{Bmatrix} P_a \\ P_b \\ P_c \end{Bmatrix} \end{Bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \{q_1\} = \begin{Bmatrix} 0 \\ -Q_b \\ Q_c \end{Bmatrix} \quad (44)$$

and $C_1 \dot{P}_a = q_1$.

Explicitly,

$$R_2 (P_a - P_b)^{1/k} + R_3 (P_a - P_c)^{1/k} + q_1 = 0 \quad (45)$$

$$-R_2 (P_a - P_b)^{1/k} + R_4 (P_b - P_c)^{1/k} = -Q_b \quad (46)$$

$$-R_3 (P_a - P_c)^{1/k} - R_4 (P_b - P_c)^{1/k} = Q_c \quad (47)$$

and $C_1 \dot{P}_a = q_1$, with the initial pressure $P_a(0)$ specified.

As can be seen, the above system of equations gives rise to a system of nonlinear differential-algebraic equations. The above equations represent four equations in four unknowns P_a , P_b , P_c , and q_1 , with given initial condition on the tank pressure, P_a .

The solution is obtained by using an iterative procedure, such as Newton's method, where, using the known initial value $P_a(0)$ and estimated values for the other three unknowns, the algebraic system of equations is solved; then using the value of q_1 and $P_a(0)$, the differential equation is integrated for a time step, giving a value for P_a at the next time instant, and the process is repeated for the remaining simulation period.

In the event that the pipe terminal equations are linear (i.e. $k = 1$), the mixed-nodal tableau equations in the Laplace domain are:

$$\begin{bmatrix} R_2 + R_3 & -R_2 & -R_3 & 1 \\ -R_2 & R_2 + R_4 & -R_4 & 0 \\ -R_3 & -R_4 & R_3 + R_4 & 0 \\ C_1 s & 0 & 0 & -1 \end{bmatrix} \begin{Bmatrix} P_a \\ P_b \\ P_c \\ q_1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -Q_b \\ Q_c \\ C_1 P_1(0) \end{Bmatrix} \quad (48)$$

where s is the Laplace variable.

3.3 Mechanical System

Shown in Figure 6 is a frictionless one-dimensional system of unconstrained masses, linear springs and dampers, and applied forces. The displacements of components are measured from the static equilibrium position of the system in the absence of the external forces. This example has been taken from Wittenburg [9], who generates the system equations of motion using an ad hoc mixture of graph-theoretic procedures (for kinematic relationships) and Newton's Second Law (to obtain dynamic equations). Wittenburg then shows how this procedure would have to be modified if the system were "free-floating", i.e. if none of the masses were connected to the ground by a spring or damper.

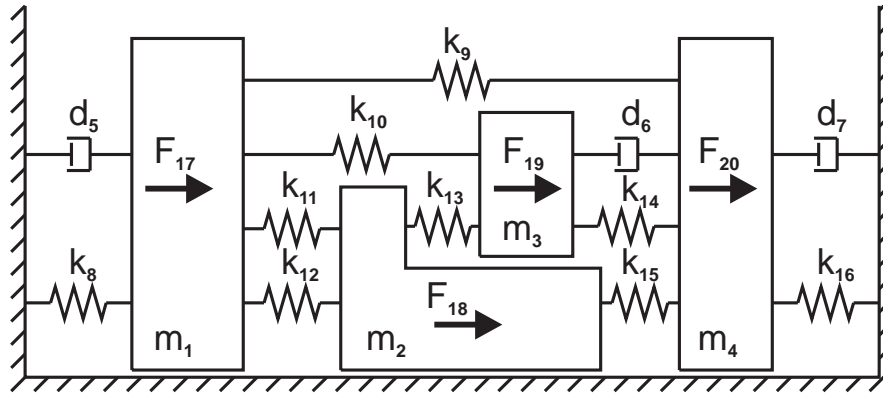


Figure 6: One-Dimensional Vibrating System

Wittenburg represents the connectivity of the various elements by separate "spring" and "damper" graphs; analogous "mass" and "force" graphs are not employed, due to the implicit assumption that these latter elements always originate from the ground node. This is clearly a limiting assumption for force elements, since hydraulic actuators are frequently connected between two bodies, neither of which is the ground. Wittenburg goes on to define a third "coordinate" graph in order to specify the variables to appear in the final equations of motion: the absolute displacements of masses m_1 and m_3 and the relative displacements of springs k_{13} and k_{15} .

To accomplish the same result using formal graph-theoretic methods, the system is represented by the linear graph shown in Figure 7. Note that the three separate graphs defined by Wittenburg are

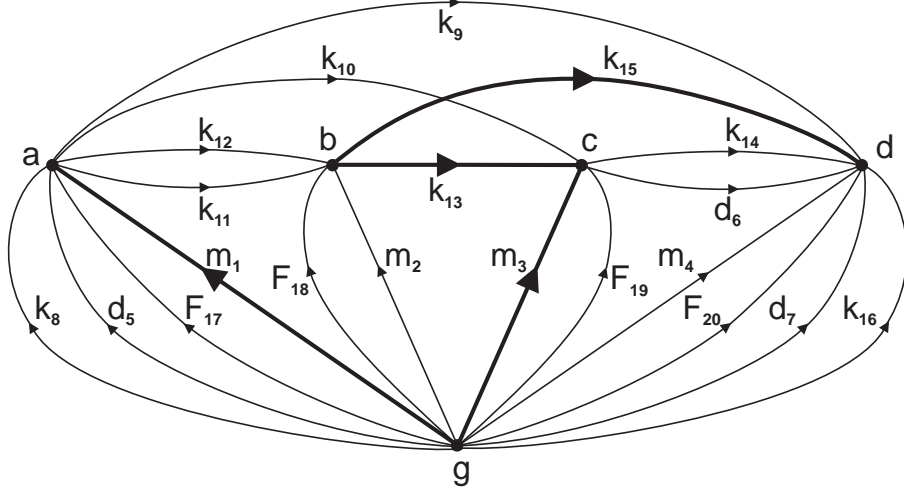


Figure 7: Linear Graph of Vibrating System

encapsulated in the single non-planar graph shown here. Using this linear graph, a hybrid branch-chord formulation will be used to generate scalar matrix expressions for the equations of motion. To systematically generate these equations in terms of the coordinates selected by Wittenburg, the corresponding elements m_1 , m_3 , k_{13} , and k_{15} are selected into the tree, which is shown in bold in the figure. The resulting set of “branch coordinates” is:

$$\{x\}_b = [x_1, x_3, x_{13}, x_{15}]^T \quad (49)$$

3.3.1 Hybrid Branch-Chord Formulation

Although the mathematical details presented below are somewhat complex, the steps in this formulation are relatively simple: the branch transformation equations are substituted into the chord terminal equations which, along with the branch terminal equations, are then substituted into the chord transformation equations. The result is the set of equations of motion for the system.

The mathematical derivation will be presented in the reverse order, starting with the chord transformation equations for all masses in the tree:

$$\{F_m\}_b = -[A_m]_c \{F\}_c \quad (50)$$

where $\{F_m\}_b$ are the forces in the tree masses, $[A_m]_c$ are the rows of the fundamental cutset submatrix $[A_c]$ corresponding to these masses, and $\{F\}_c$ contains the forces in all cotree elements. Substituting the mass terminal equations,

$$-[m]_b \{\ddot{x}_m\}_b = -[A_m]_c \{F\}_c \quad (51)$$

where $[m]_b$ is a diagonal matrix containing the masses of the two rigid bodies in the tree, and $\{\ddot{x}_m\}_b$ contains their accelerations.

Similar chord transformation equations can be generated for the translational springs, dampers, and forces in the tree:

$$-[d]_b \{\dot{x}_d\}_b = -[A_d]_c \{F\}_c \quad (52)$$

$$-[k]_b\{x_k\}_b = -[A_k]_c\{F\}_c \quad (53)$$

$$\{F_f\}_b = -[A_f]_c\{F\}_c \quad (54)$$

noting that $[A_d]_c$ and $[A_f]_c$ will be zero for this example, since no dampers or forces were selected into the tree. Together, equations (51-54) constitute a set of b ordinary differential equations, where b is both the number of elements in the tree (four for this example) and, for a system of one-dimensional unconstrained masses, the number of degrees of freedom.

By substituting the terminal equations for cotree elements into $\{F\}_c$, one can see that the number of variables in this set of ODEs is greater than four:

$$\{F\}_c = \begin{Bmatrix} -[m]_c\{\ddot{x}_m\}_c \\ -[d]_c\{\dot{x}_d\}_c \\ -[k]_c\{x_k\}_c \\ \{F_f\}_c \end{Bmatrix} \quad (55)$$

where $\{\ddot{x}_m\}_c$, $\{\dot{x}_d\}_c$, $\{x_k\}_c$, and $\{F_f\}_c$ respectively correspond to masses, dampers, springs, and forces in the cotree. To eliminate these additional across variables from the set of ODEs, the branch transformation equations are applied.

Starting with the masses in the cotree,

$$\{\ddot{x}_m\}_c = -[B_m]_b\{\ddot{x}\}_b \quad (56)$$

where $[B_m]_b$ contains the rows of the fundamental circuit sub-matrix $[B_b]$ corresponding to the cotree masses, and $\{\ddot{x}\}_b$ contains the accelerations of all tree elements. Similarly for the cotree dampers and springs,

$$\{\dot{x}_d\}_c = -[B_d]_b\{\dot{x}\}_b \quad (57)$$

$$\{x_k\}_c = -[B_k]_b\{x\}_b \quad (58)$$

Substituting equations (55-58) into equations (51-54), one obtains a set of four ODEs that can be written in the standard form:

$$[M]\{\ddot{x}\}_b + [D]\{\dot{x}\}_b + [K]\{x\}_b = \{F\} \quad (59)$$

where $[M]$, $[D]$, and $[K]$ are the system mass, damping, and stiffness matrices, respectively. The explicit mathematical expressions for these matrices, which are functions of the topological sub-matrices and component parameters, are rather large and complex.

Once generated, the linear ODEs (59) can be solved analytically or numerically for the motions of the selected four branches, after which the motions of all chords can be obtained through back-substitution into the branch transformation equations (56-58). At this point, the dynamic analysis is completed.

3.3.2 Alternative Formulation

Before leaving this example though, it is enlightening to formulate the equations in a slightly different fashion, in order to get expressions for the final set of ODEs that are both explicit and concise.

Instead of ordering the fundamental cutset matrix (6) so that the leftmost columns correspond to elements in the tree, one can partition $[A_f]$ as follows:

$$[A_f] = \left[[A_m][A_d][A_k][A_f] \right] \quad (60)$$

where $[A_m]$ corresponds to all mass elements, placed in the leftmost columns, $[A_d]$ corresponds to all dampers, $[A_k]$ to all springs, and $[A_f]$ to all forces. If the k th element is in the tree, then the corresponding column contains all 0's except for a single 1 in the k th row. To illustrate, the expressions for $[A_m]$ and $[A_d]$ for this example are:

$$[A_m] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (61)$$

$$[A_d] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \quad (62)$$

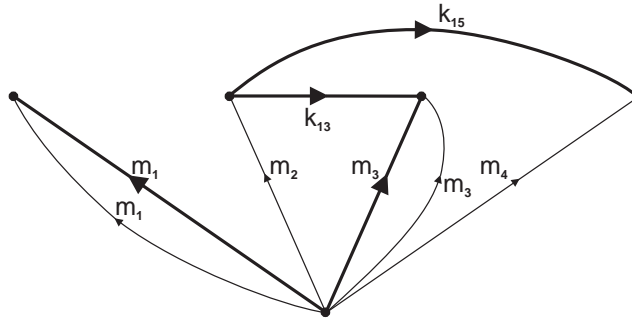


Figure 8: Union of Tree and Mass Elements

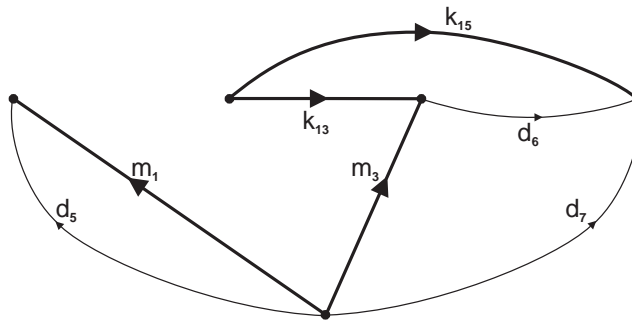


Figure 9: Union of Tree and Damper Elements

It is instructive to note that these sub-matrices can be generated by examining the sub-graphs of Figure 7 corresponding to the union of all tree elements with all mass elements, dampers, springs, and forces. As an example, the sub-matrix $[A_m]$ is precisely the fundamental cutset sub-matrix $[A_c]$ for the linear graph shown in Figure 8. Note that in order to relate the variables for all mass elements to those for all tree elements, it is necessary to duplicate elements m_1 and m_3 , since they appear in both of these sets. For the sub-graph shown in Figure 9, from which $[A_d]$ is obtained, it is not necessary to duplicate any damper elements since none were selected into the tree.

With this alternative partitioning of $[A_f]$ for the system graph, the fundamental cutset equations (5) take the form:

$$[A_m]\{F_m\} + [A_d]\{F_d\} + [A_k]\{F_k\} + [A_f]\{F_f\} = \{0\} \quad (63)$$

where $\{F_m\}$ corresponds to the forces in all masses, $\{F_d\}$ contains all damper forces, and similarly for $\{F_k\}$ and $\{F_f\}$. Substituting the terminal equations for these column matrices of through variables,

$$[A_m][m]\{\ddot{x}_m\} + [A_d][d]\{\dot{x}_d\} + [A_k][k]\{x_k\} = [A_f]\{F_f\} \quad (64)$$

where $\{\ddot{x}_m\}$ corresponds to all mass accelerations, $\{\dot{x}_d\}$ to all spring velocities, and $\{x_k\}$ to all spring displacements (measured from the static equilibrium position). If the four mass elements had been selected to form the tree, then $[A_m]$ would be a unit matrix and the accelerations in equation (64) could be decoupled using the fact that $[m]$ is a diagonal mass matrix.

To eliminate the excess variables appearing in equation (64), it is necessary to relate the displacements of all mass, spring, and damper elements to the displacements $\{x\}_b$ of elements in the tree. For mass elements, this is directly obtained from the branch transformation equations for the sub-graph shown in Figure 8:

$$\{\ddot{x}_m\} = [A_m]^T \{\ddot{x}\}_b \quad (65)$$

noting that the orthogonality relationship (13) has been employed. Similarly, the branch transformation equations for the graph in Figure 9 provides the relationship for damper elements:

$$\{\dot{x}_d\} = [A_d]^T \{\dot{x}\}_b \quad (66)$$

and finally, for spring elements:

$$\{x_k\} = [A_k]^T \{x\}_b \quad (67)$$

Substituting equations (65-67) into equation (64), one obtains the final set of governing ODEs in the concise and insightful form:

$$[A_m][m][A_m]^T \{\ddot{x}\}_b + [A_d][d][A_d]^T \{\dot{x}\}_b + [A_k][k][A_k]^T \{x\}_b = [A_f]\{F_f\} \quad (68)$$

in complete agreement with equation (67) from Wittenburg [9]. Unlike Wittenburg's derivation though, this set of equations was obtained using a consistent graph-theoretic approach. This same approach can be used without modification to formulate the equations for a free-floating version of this one-dimensional unconstrained system; in such a case, the tree would have to incorporate at least one mass element, the coordinate for which would appear in the final equations of motion. Furthermore, only the single linear graph shown in Figure 7 is needed to derive the sub-matrices $[A_m]$, $[A_d]$, $[A_k]$, and $[A_f]$; the sub-graphs in Figures 8 and 9 are presented for pedagogical reasons only. The relationship between these topological matrices and the system mass, damping, and stiffness matrices, which depend directly upon the selection of elements into the tree, is clearly seen from a comparison of the two equivalent sets of motion equations (59) and (68).

4 Two-Dimensional Mechanical Systems

In this Section, the basic graph-theoretic modelling concepts presented in Section 2 are applied to the kinematic and dynamic analysis of planar mechanical systems. Two linear graphs, one for translational motions and one for rotational motions, are used to represent the multibody system topology. The terminal equations for the multi-dimensional components are presented in concise vectorial form, as are the topological equations. An efficient formulation procedure, that uses orthogonal projection techniques to eliminate non-working constraint reactions, is then described. The Section concludes with a number of different examples that demonstrate the efficacy and simplicity of a graph-theoretic approach, when compared to conventional methods for multibody dynamics [11]–[17].

4.1 Linear Graph Representation

In order to apply formal graph-theoretic methods to a multibody system, it is necessary to create a linear graph that encapsulates the system topology. As was shown for one-dimensional systems, this is accomplished by identifying the significant points in the physical system, and representing them by the nodes of a graph. For a multi-dimensional mechanical system, these vertices correspond to centres of mass for rigid bodies, a datum node that represents the origin of an inertial reference frame, and points on a body (other than the mass centre) to which forces are applied or joints are connected. These latter nodes have no counterpart in a one-dimensional system graph.

Edges are then drawn between these vertices so that each edge corresponds to a physical element in the original system. Thus, a rigid body would be represented by an edge starting at the datum node and ending at the mass centre node, while an external force would also start at the datum node and end at the node representing the point of application. The location of this point relative to the body centre of mass is given by a body-fixed vector, which is represented by a “rigid-arm” element between the mass centre node and the point. Finally, kinematic joints or spring-damper-actuators between two bodies are represented by edges connecting the end points of these physical components.

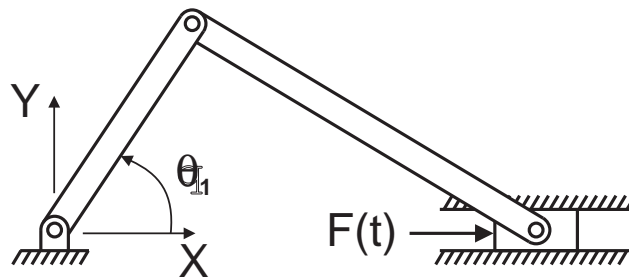


Figure 10: Slider-Crank Mechanism

Applying these guidelines to the slider-crank mechanism shown in Figure 10, the linear graph of Figure 11 is obtained. Edges m_1 , m_2 , and m_3 represent the three rigid bodies in this closed-loop

mechanism, the rigid-arm elements r_5 to r_8 correspond to body-fixed position vectors that define the location of pin joints on the crank and connecting arm, and h_9 to h_{11} represent these three joints. The edge s_{12} corresponds to the prismatic joint between the slider body and the ground, while R_4 represents a ground-fixed vector from the origin to a point on the axis of sliding. Finally, edge F_{13} corresponds to an external force applied to the slider body. Note that the three rigid bodies are drawn in dotted lines on top of the linear graph, to emphasize the similarity between the graph topology and the structure of the physical system.

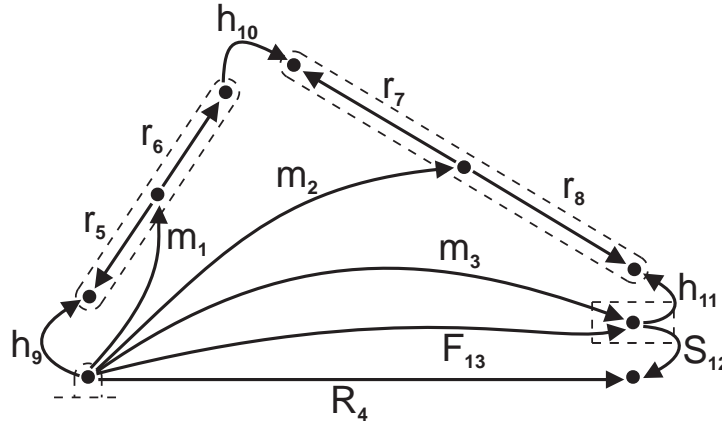


Figure 11: Linear Graph of Slider-Crank

Chasles’s theorem states that the general motion of a rigid body can be decomposed into independent translational and rotational motions. Thus, the through variables for a multibody system will consist of both forces and torques, while the across variables will include both translational and rotational displacements. In addition, joints connected to a rigid body may constrain either its translational or rotational motion, or both. The linear graph shown in Figure 11 represents only the translational motion of elements in the system; the topological equations obtained from this graph will be in terms of forces and translational displacements (or velocities or accelerations) only. Thus, this linear graph will be referred to as the “translational” graph, or T-graph. A separate graph, shown in Figure 12, is therefore needed to represent the rotational motions of the slider-crank components. From this “rotational graph”, or R-graph, a second set of topological equations will be generated, in terms of torques and rotational displacements.

Upon examination, it is obvious that the rotational graph is simpler than its translational counterpart. This is due to the fact that rotational quantities are properties of an entire body, and not of any single point on the body. For example, angular velocity (or displacement or acceleration) is a property of a rigid body or reference frame, and has no meaning for an isolated point. Similarly, the dynamic effects of an external torque is independent of where the torque is applied to a rigid body. Thus, all points on a rigid body are shrunk to a single node representing the entire body, thereby eliminating elements like body-fixed and ground-fixed vectors from the R-graph. The four nodes in Figure 12 represent the three moving bodies, and the inertial reference frame attached to the ground body. The only elements that appear in the R-graph are the three inertia elements m_{1-3} and the four joints h_{9-11} and s_{12} connecting these three rigid bodies. In addition, the driving force F_{13} does not appear in the R-graph, since it is a purely translational quantity.

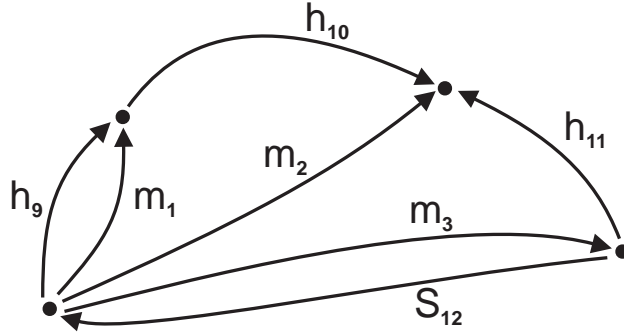


Figure 12: Rotational Graph of Slider-Crank

An interesting observation is the fact that the R-graph can be automatically generated from the T-graph by eliminating edges corresponding to body-fixed and ground-fixed vectors, and collapsing the end nodes of these edges to form a single node representing the body. In addition, purely translational quantities, such as forces or translational springs and dampers, are deleted while torques and other rotational elements (springs, dampers, actuators) are added to the R-graph. In this manner, Figure 12 is systematically obtained from Figure 11, which is the only graph required to generate the complete set of topological equations³. In the remainder of this section though, both translational and rotational graphs will be presented for two reasons: firstly because it is desirable to select different elements into the trees for the two graphs, and secondly for pedagogical purposes.

Just as scalar quantities provide a suitable description of electrical or hydraulic networks, physical vectors provide a convenient mathematical description of the through and across variables associated with multi-dimensional mechanical systems. In particular, the through variable corresponding to an edge in the T-graph is concisely represented by a force vector \underline{F} , while the across variable is represented by the displacement vector \underline{r} , or any of its derivatives (velocity \underline{v} or acceleration \underline{a}). For an edge in the R-graph, the corresponding through variable is a torque vector \underline{T} , while the across variable corresponds to the rotational displacement ($\underline{\theta}$), which can be represented by a vector quantity for two-dimensional mechanical systems.

4.2 Terminal Equations

Associated with every edge is one or more terminal equations that define the physical characteristics of the corresponding element. Consistent with formal graph-theoretic methods, these terminal equations are written in terms of the system through and across variables, defined above. As an example, a terminal equation for edge h_9 is:

$$\underline{r} = 0 \tag{69}$$

where \underline{r} is the translational displacement corresponding to the revolute joint h_9 . This terminal equation (69) ensures that the two points on the ground and crank connected by h_9 remain coincident at all times.

³In the earliest research in this area, Andrews [10] incorporated rotational quantities into the T-graph to form a single “Vector-Network Diagram”.

As another example, the terminal equation for a rigid-arm element can be written vectorially as:

$$\underline{r} = \underline{r}(\underline{\theta}) \quad (70)$$

where $\underline{\theta}$ is the angular displacement of the rigid body to which the arm is attached. In other words, the displacement vector for a rigid-arm varies as the carrier body rotates. In scalar form, equation (70) represents a rotation transformation, a fact that is made more explicit by writing (70) in matrix form:

$$\{r\} = [R(\theta)]\{r\}' \quad (71)$$

where $\{r\}'$ contains the two constant components of \underline{r} in a local body-fixed frame, $\{r\}$ contains the components of \underline{r} resolved in a global frame, and $[R]$ is the rotation transformation matrix between these two frames, defined as:

$$[R] = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \quad (72)$$

For an ideal prismatic joint, one terminal equation is:

$$\underline{r} = s\hat{u}(\underline{\theta}) \quad (73)$$

where s is the magnitude of the displacement, which must be parallel to the axis of sliding defined by the unit vector \hat{u} . Similar to the rigid-arm displacement, the unit vector \hat{u} is a function of the orientation angle ($\underline{\theta}$) of the body on which the slider axis is defined. A second terminal equation for a prismatic joint is:

$$\underline{F} \cdot \hat{u} = 0 \quad (74)$$

which states that the constraint force \underline{F} is normal to the axis of sliding.

Element	Translation	Rotation
Rigid Body (m)	$\underline{F}_1 = -m_1 \underline{a}_1$	$\underline{T}_1 = -I_1 \underline{\alpha}_1 - \sum \underline{r} \times \underline{F}$
Position Driver (R)	$\underline{r}_2 = \underline{r}_2(t)$	
Rotation Driver (θ)		$\underline{\theta}_3 = \underline{\theta}_3(t)$
Rigid Arm (r)	$\underline{r}_4 = \underline{r}_4(\underline{\theta}_1)$	
(body-fixed vector)	$\underline{v}_4 = \underline{\omega}_1 \times \underline{r}_4$ $\underline{a}_4 = \underline{\alpha}_1 \times \underline{r}_4 - \omega_1^2 \underline{r}_4$	
Force Actuator (F)	$\underline{F}_5 = \underline{F}_5(\underline{r}_5, \underline{v}_5, t)$	
Torque Actuator (T)		$\underline{T}_6 = \underline{T}_6(\underline{\theta}_6, \underline{\omega}_6, t)$
Revolute Joint (h)	$\underline{r}_7 = 0$	$\underline{T}_7 = 0$
Prismatic Joint (s)	$\underline{r}_8 = s_8 \hat{u}_8$ $\underline{F}_8 \cdot \hat{u}_8 = 0$	$\underline{\omega}_8 = 0$ $\underline{\alpha}_8 = 0$

Table 4: Terminal Equations for Planar Mechanical Components

As a final example, the terminal equation governing the translational motion of a rigid body is given by the d'Alembert form:

$$\underline{F} = -m\underline{a} \quad (75)$$

which states that the force “through” a rigid body is equal and opposite to the resultant of external forces on the body. Similarly, the terminal equation governing the rotational motion of a rigid body can be written:

$$\underline{T} = -I\underline{\alpha} - \sum \underline{r}_r \times \underline{F}_r - \sum \underline{r}_s \times \underline{F}_s \quad (76)$$

where \underline{r}_r and \underline{r}_s represent the respective displacements of rigid-arms and prismatic joints connected to the rigid body, and \underline{F}_r and \underline{F}_s are the corresponding forces in these elements. Essentially, the two summations appearing in the rotational terminal equation (76) represent torques about the centre of mass resulting from forces applied at other points on the body. This model of a rigid body represents a departure from conventional graph-theoretic approaches [10, 18, 19] that use additional “dependent torque” elements to represent each cross product term in the two summations. The approach taken here results in much simpler rotational graphs that are similar in form to those employed by other authors [20, 21].

The terminal equations for an individual component can only be determined from empirical observation, and may be highly nonlinear in nature. Furthermore, they may be written in terms of translational or rotational quantities, or a combination of both. Shown in Table 4 are the terminal equations for a variety of planar mechanical components, all of which are included in the models of multibody systems presented later. For a more complete discussion of these and other terminal equations, see [18, 19].

4.3 Topological Equations

Having defined linear graphs to represent the multibody system, standard graph-theoretic methods can now be used to generate the linear topological relationships between the sets of through and across variables. Clearly, translational and rotational quantities with different physical units cannot be added directly together; hence, separate cutset and circuit equations are generated for the two different linear graphs.

Once a “T-tree” has been selected for the translational graph, the fundamental cutset equations (5) can be automatically generated:

$$[A_f]\{\underline{F}\} = \{0\} \quad (77)$$

referred to as the “T-cutset equations”, where $[A_f]$ is the fundamental cutset matrix for the T-graph. The corresponding chord transformation equations (20) take the form:

$$\{\underline{F}\}_b = -[A_c]\{\underline{F}\}_c \quad (78)$$

which express the forces in branches as a linear combination of the forces in cotree elements.

Physically, the T-cutset equations represent the conditions for dynamic equilibrium. As an example, consider the T-graph in Figure 11, into whose cotree all of the joint elements and the force driver are placed. From a visual inspection of the graph, the chord transformation equation for branch m_2 is:

$$\underline{F}_2 = -\underline{F}_{10} - \underline{F}_{11} = -m_2\underline{a}_2 \quad (79)$$

which corresponds to Newton’s Second Law, written for a free-body diagram of body 2. Using a graph-theoretic approach, no free-body diagram is needed.

The “T-circuit equations” can also be automatically generated as:

$$[B_f]\{\underline{r}\} = \{0\} \quad (80)$$

where $[B_f]$ is the fundamental circuit matrix for the T-graph. The corresponding set of branch transformation equations are:

$$\{\underline{r}\}_c = -[B_b]\{\underline{r}\}_b \quad (81)$$

where $\{\underline{r}\}_c$ are the chord displacements and $\{\underline{r}\}_b$ are the branch displacements. In either equation (80) or (81), these displacements can be replaced by translational velocities $\{\underline{v}\}$ or accelerations $\{\underline{a}\}$.

Physically, the T-circuit equations represent the kinematic relationships between the various physical components. As an example, and using the same tree selection as above, the T-circuit equation for chord h_{10} is obtained from a visual inspection of the T-graph in Figure 11:

$$\underline{r}_{10} - \underline{r}_7 - \underline{r}_2 + \underline{r}_1 + \underline{r}_6 = 0 \quad (82)$$

which corresponds to the conditions required for vector loop closure of these translational displacements.

Analogous dynamic and kinematic relationships are obtained from the cutset and circuit equations for the R-graph. Specifically, the ‘‘R-cutset’’ equations take the form:

$$[C_f]\{\underline{T}\} = 0 \quad (83)$$

where $[C_f]$ is the fundamental cutset matrix for the R-graph. The corresponding chord transformation equations are:

$$\{\underline{T}\}_b = -[C_c]\{\underline{T}\}_c \quad (84)$$

where $[C_c]$ is the rotational counterpart to $[A_c]$ for the T-graph.

Similarly, the ‘‘R-circuit’’ equations are written as:

$$[D_f]\{\underline{\theta}\} = 0 \quad (85)$$

where $[D_f]$ is the fundamental circuit matrix for the R-graph. The corresponding branch transformation equations are:

$$\{\underline{\theta}\}_c = -[D_b]\{\underline{\theta}\}_b \quad (86)$$

where $[D_b]$ is the rotational counterpart to $[B_b]$ for the T-graph. It can be calculated directly from $[C_c]$ using the orthogonality relationship (13):

$$[D_b] = -[C_c]^T \quad (87)$$

4.4 Tree and Coordinate Selection

Together, the cutset, circuit, and terminal equations provide a necessary and sufficient set of equations for determining the time response of a mechanical system [18, 19]. Thus, the selection of trees for the translational and rotational graphs does not affect the underlying mathematical model; in fact, a tree is not even needed in a nodal formulation of the governing equations, as discussed in Section 2.5. However, the selection of trees can greatly reduce the number of motion equations that have to be solved simultaneously, especially if some care is taken in selecting the branches of these trees.

In graph-theoretic terminology, the tree across and cotree through variables are known as “primary variables”, while the tree through and cotree across variables are known 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 (20,21). Using these linear equations therefore, the secondary variables can be eliminated from the set of governing equations. The result is a smaller number of equations expressed solely in terms of across variables for tree elements — the *branch coordinates* — 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.

One can then make the observation that the number of equations of motion remaining will depend directly on the number of branch coordinates that have been used. Thus, the number of equations can be further reduced by selecting into the trees *those elements for which a minimum number of across variables are unknown*. The result will be a smaller number of branch coordinates and therefore, a smaller number of equations of motion.

Using this simple criterion, one can examine the terminal equations in Table 4 to establish a preferred order for selecting elements into the translational tree. Clearly, it is desirable to include position drivers and revolute joints into the tree, since their across variables are completely known functions of time. Similarly, it is preferable to select prismatic joints as branches over rigid body elements, since the former introduces only one unknown variable s into the set of branch coordinates, while the latter introduces the two unknown x and y displacements of the body. Finally, it is advantageous to put force actuators into the cotree, since the corresponding through variables appearing in the motion equations would be known functions.

Less obvious is the fact that rigid-arm elements should be selected first into the T-tree, even though they are implicitly defined in terms of the rotation of the carrier body. However, since all the rigid-arms on a given body are functions of the same orientation angle, *at most* only one variable needs to be added to the set of branch coordinates. Thus, if one arm on a body is included in the tree, then all arms on the body should be included. The alternative is to include no rigid-arms, in which case the mass element would be needed to complete the tree. As mentioned above, this would introduce two additional variables into the set of branch coordinates; thus it is preferable to select all rigid-arm elements into the tree. Furthermore, fewer additional branches will be required to complete the tree since there are generally more rigid-arm elements than mass elements.

Based upon these arguments, the components listed in Table 4 should be selected into the T-tree in the following order, until the tree is complete:

1. Rigid arm elements.
2. Position drivers.
3. Revolute joints.
4. Prismatic joints.
5. Rigid bodies.
6. Force actuators.

Using the same criterion for reducing the number of equations of motion, one can also determine a preferred order for selecting elements into the rotational tree. Again, rotation drivers and prismatic joints are selected first, since the corresponding rotational across variables are known

functions of time. Then, one should select any rigid body elements corresponding to the rigid-arms in the translational tree, unless the rigid body rotation has already been defined by a rotation driver or a prismatic joint. Furthermore, the selection of rigid body rotations ahead of revolute joint coordinates leads to a simplified form of the equations of motion, as pointed out by Huston [22]. Finally, it is advantageous to put torque actuators into the cotree, since their through variables are given functions.

Thus, the components from Table 4 should be selected into the rotational tree in the following order:

1. Rotation drivers.
2. Prismatic joints.
3. Rigid bodies.
4. Revolute joints.
5. Torque actuators.

Comparing the two different tree selection schemes, one can see that it is generally better to **not** select the same components into the trees of the translational and rotational graphs. However, most previous graph-theoretic formulations [10, 19, 23], with the exception of Li and Andrews [18], do exactly this. Some approaches [20, 21] make use of a single graph to represent the mechanical system, and a single tree to generate equations for rotation and translation. Although Li and Andrews do employ two different trees, their selection scheme is less than optimal from the point of view of equation reduction. As an example, prismatic joints are selected into the translational tree ahead of revolute joints. This leads to a larger number of branch coordinates, and therefore equations, for the reasons outlined above. Furthermore, Li and Andrews' classification scheme is very restrictive in that no prismatic joints are allowed in the translational cotree, and no revolute joints are allowed in the rotational tree. The only restriction on the selection of trees in this current work is that all rigid-arm elements must be in the translational tree. As long as this one requirement is met, the graph-theoretic procedure described in the next section will automatically generate the motion equations corresponding to any tree selection.

4.5 Graph-Theoretic Formulation Procedure

Once the topology and parameters for a mechanical system have been defined, the translational and rotational graphs can be generated automatically. The only other required information is the identification of trees for the two graphs; in the absence of such identification, the tree selection criteria given in the previous section can be employed. At this point, the equations governing the motion of the mechanical system can be automatically formulated by following the four simple steps of the procedure outlined below. This procedure represents a new approach [24] to formulating the motion equations for multibody systems, and is obtained by combining graph-theoretic methods with a variation of the "projection methods" presented by Scott [25] and Blajer [26]. Essentially, this procedure represents a generalization of the joint coordinate formulation developed by Baciuc and Kesavan [20].

Step 1: Projection of Cutset Equations onto Motion Space

To formulate dynamic equations for the mechanical system, the cutset equations for all rigid bodies, passive joints, and forces/torques in the tree are generated from the graphs. These cutset equations are then projected onto the “motion space” of the corresponding component, which is the space of motions allowed by that component (a generalization of the “relative motion space” concept [20] for passive kinematic constraints). Clearly, the across variables for a given component span its own motion space. As an example, the motion space of a planar rigid body is described by the corresponding across variables x , y , and θ , while the motion space of a prismatic joint is defined by \hat{u} , the unit vector parallel to the slider axis. Thus, the cutset equations for a rigid body are projected onto the x , y , and z axis, where z is normal to the plane of motion, while the cutset equations for a prismatic joint are projected onto the slider axis.

For kinematic constraint elements (joints), one can also define a “reaction space”, which corresponds to the space spanned by the constraint forces and torques for that element. For revolute joints, the reaction space corresponds to the xy plane of motion, while the reaction space of a prismatic joint is defined by the unit vectors \hat{n} and \hat{k} , both of which are perpendicular to the motion space defined by \hat{u} . In fact, for any passive frictionless joint, the reaction space will be orthogonal to the motion space, since no work is done by that joint. This is not the case for active kinematic constraints.

The motion space for the entire system is simply the union of the motion spaces of the individual components selected into the tree; thus, the system motion space is spanned by the set of branch coordinates. *The effect of projecting the dynamic cutset equations onto this motion space is to eliminate the reaction forces and torques corresponding to passive kinematic constraints selected into the tree.* After substitution of the terminal equations from Table 4 into these projected cutsets, the result is a set of n differential equations, where n is the number of branch coordinates. In the general case, these branch coordinates are not independent; hence, they do not form a basis for the system motion space. Therefore, these n differential equations are in terms of both the tree and cotree across variables, as well as the reaction forces and torques for cotree kinematic constraints. For an open-loop mechanical system, an independent set of branch coordinates is obtained by simply selecting all kinematic constraints into the trees; in this case, the system motion space is equivalent to the “tangent space” defined by Scott [25], and all constraint reactions are automatically eliminated from the dynamic equations.

Note that explicit expressions for the cotree constraint reactions can be generated, if desired, by projecting the same cutset equations onto the reaction space corresponding to the cotree element.

Step 2: Projection of Circuit Equations onto Reaction Space

For a mechanical system containing closed kinematic chains, one can generally define branch coordinates that are fewer in number than a set of joint coordinates for the same system. Nevertheless, these n branch coordinates will still be dependent variables, related by m nonlinear algebraic equations of constraint. These are obtained by first generating the circuit equations for active and passive kinematic constraints placed into the cotrees. *By projecting these circuit equations onto the corresponding reaction space, the across variables for cotree constraint elements are automatically eliminated.* The resulting constraint equations are in terms of the branch coordinates, as well as the across variables for cotree elements other than kinematic constraints. Note that an algebraic equation is obtained for each reaction force or torque appearing in the dynamic equations from Step 1.

Also note that explicit expressions for the cotree constraint across variables can be generated

by projecting these same circuit equations onto the corresponding motion space. These may be required to compute slider displacements and velocities appearing in the mass terminal equation (76), or may simply be of interest to the analyst.

Step 3: Elimination of Secondary Variables

The next step in the procedure is to use the branch and chord transformations to eliminate any secondary variables that appear in the projected cutset and circuit equations. It is important that these variables be eliminated in the following order:

3(a). Rigid-arm forces: These appear in rotational cutset equations as a result of the mass terminal equations. These tree through variables can be expressed in terms of cotree through variables using the chord transformation equations (78) for the translational graph. Upon substitution of terminal equations, the rigid-arm forces are replaced by primary variables, and secondary variables representing cotree across variables.

3(b). Cotree translational across variables: These correspond to rigid bodies and force actuators in the cotree of the translational graph. Using the translational branch transformation equations (81) and the set of corresponding terminal equations, these variables can be replaced by expressions involving branch coordinates and across variables for rigid bodies in the cotree of the rotational graph.

3(c). Cotree rotational across variables: These correspond to rigid bodies and kinematic constraints in the rotational cotree, and can be expressed directly in terms of branch coordinates using the branch transformation equations (86) for the rotational graph.

By the end of Steps 3(a)-(c), all secondary variables have been eliminated from the projected cutset and circuit equations, which are solely in terms of the n branch coordinates and m cotree constraint reactions.

Step 4: Assembly of Motion Equations

For a kinematic analysis, Step 1 can be skipped since the cutset equations represent the dynamic equations that would be obtained by applying the Newton-Euler equations to selected sub-sections of the system. Assuming that all constraints are holonomic, the result of Steps 2 and 3 is the set of m nonlinear algebraic equations:

$$\{\Phi(q, t)\} = 0 \quad (88)$$

where $\{q\}$ is the set of n branch coordinates. Since $m = n$ for a well-posed kinematic problem, the values of $\{q\}$ at any time t can be found by solving equation (88) using an appropriate numerical method.

If a subsequent velocity analysis is required, then the projected circuit equations of Step 2 are written in terms of velocities, and not displacements. After substituting the necessary terminal equations and applying Step 3, one obtains the linear algebraic equations:

$$\{\Phi\}_q \{\dot{q}\} = -\{\Phi\}_t \quad (89)$$

where $\{\Phi\}_q$ is the Jacobian matrix of the constraint equations (88), and $\{\Phi\}_t$ is the partial derivative of these same equations with respect to time. Once the values of $\{q\}$ have been obtained at a given time, the linear equations (89) can be solved for the velocities $\{\dot{q}\}$.

If an acceleration analysis is also required, the projected circuit equations are written in terms of accelerations, and Step 3 is used to eliminate variables other than $\{q\}$ or its derivatives. The

result is another set of linear equations:

$$\{\Phi\}_q\{\ddot{q}\} = -(\{\Phi\}_q\{\dot{q}\})_q\{\dot{q}\} - 2\{\Phi\}_{qt}\{\dot{q}\} - \{\Phi\}_{tt} \quad (90)$$

which can be solved for the accelerations $\{\ddot{q}\}$ once $\{q\}$ and $\{\dot{q}\}$ have been obtained from the previous analyses.

Finally, if a dynamic analysis is required, the constraint equations (88) are insufficient in number to solve for $\{q(t)\}$. In this case, m is less than n , the difference being equal to the number of degrees of freedom of the system, and the constraint equations are supplemented by the differential equations produced by Steps 1 and 3:

$$[M]\{\ddot{q}\} + \{\Phi\}_q^T\{\lambda\} = \{Q(q, \dot{q}, t)\} \quad (91)$$

where $[M]$ is a symmetric positive-definite mass matrix, $\{\lambda\}$ is a set of m Lagrange multipliers corresponding to the reaction forces and torques in the cotree kinematic joints, and $\{Q\}$ contains external loads and quadratic velocity terms. Together, equations (88) and (91) constitute a set of $n + m$ differential-algebraic equations (DAEs) that can be solved for the n coordinates $\{q(t)\}$ and m Lagrange multipliers $\{\lambda(t)\}$ using a suitable numerical method.

At this point, one can generate the motion equations resulting from an absolute coordinate formulation [11, 14] by selecting all mass elements into both trees; the branch coordinates are then identical to the set of absolute coordinates. In this case, the projected cutset equations represent the Newton-Euler equations for each individual rigid body, with the reaction forces and torques for all kinematic constraints appearing explicitly. Similarly, the projected circuit equations correspond to the constraint equations obtained from an examination of the joints and drivers, all of which are in the cotree. Combined, these equations constitute a relatively large set of DAEs for which sparse methods of numerical solution may be appropriate.

To obtain the equations of motion corresponding to a joint coordinate formulation [16, 20, 21], one joint after another is added to both the translational and rotational trees, until they are complete. The projected circuit equations then represent the conditions for “loop closure”, while the constraint reactions in cotree (“cut”) joints appear in the projected cutset equations. The combined set of DAEs are generally fewer in number than those obtained using absolute coordinates, and can be solved quite efficiently using recursive techniques.

For open-loop mechanical systems, the set of DAEs given by equations (88) and (91) can be reduced to a “minimal” set of ordinary differential equations (ODEs) equal in number to the degrees of freedom of the system. This is accomplished by simply selecting all joints and position drivers into the T-tree and all prismatic joints and rotation drivers into the R-tree. With this tree selection, no constraint equations ($m = 0$) are generated by Step 2; thus, the n branch coordinates are independent generalized coordinates, and the n ODEs obtained from Steps 1 and 3 are sufficient to determine the time response of the system.

For systems with closed kinematic chains and a mixture of revolute and prismatic joints, the DAEs obtained using the trees recommended in Section 3 are generally fewer in number than those obtained using joint coordinates, as is shown in the examples that follow. These equations can be further reduced analytically to a minimal number of ODEs only through the use of complicated geometrical transformations [27, 28]. It is Nikravesh’s contention [29] that the extra effort required for this last level of equation reduction offsets the computational savings obtained from having a smaller set of equations.

Computer Implementation

Due to its systematic nature, the previous formulation was encoded with relative ease [30] into a sequence of Maple procedures. Using conventional graph-theoretic methods, the cutset and circuit equations are automatically generated from the given system topology and for the tree selected by the user. The terminal equations shown in Table 4 are contained in a library of modelling components that can easily be updated to include new components. From the projected cutset and circuit equations, the set of motion equations governing the kinematic or dynamic response of the given system is automatically assembled in a symbolic form that provides insight into their structure. One advantage of using a commercial symbolic package is that the velocity and acceleration equations (89-90) can be directly obtained by symbolically differentiating the constraint equations (88) with respect to time; this greatly reduces the amount of programming effort required. Although there are other significant advantages of a symbolic approach [30], a general solution to the nonlinear equations of motion can only be obtained using numerical methods. The ability of Maple to write these equations in either C++ or Fortran format facilitates their subsequent numerical solution.

A second computer implementation [31] of this graph-theoretic formulation was created using a purely numerical approach based on object-oriented programming (OOP). Although the advantages of a symbolic implementation are not realized, an OOP approach is able to exploit the analogy that exists between physical components and object “classes”, resulting in very modular C++ code that was relatively quick to develop. The software (**MeSyA**) is easily maintained, and new components can be added by simply developing new object classes. A graphical user interface for defining the mechanical system and animating its computed response was developed using AutoCAD.

4.6 Examples

4.6.1 Dynamic Equations for Double Pendulum

Shown in Figure 13 is a double pendulum consisting of two rigid bodies connected in series to the ground by revolute joints. Assuming that the initial conditions are specified for this open-loop system, one can determine the motion of the double pendulum in a uniform gravitational field once the governing equations have been obtained. The graph-theoretic formulation procedure presented in Section 4.5 will be used to accomplish this equation generation.

The T-graph for the double pendulum, shown in Figure 14, consists of 2 rigid body elements m_{1-2} , 4 rigid-arm elements r_{3-4} , 2 revolute joints h_{7-8} , and the 2 gravitational forces W_{9-10} . Although the rigid-arm element r_6 is not strictly needed, it has been included for the purposes of tracking the tip of the outer body.

The R-graph for the double pendulum, shown in Figure 15, is obtained directly from the T-graph by collapsing the end nodes of rigid-arm elements and deleting the two gravitational forces.

For this example, the preferred tree selection scheme presented in Section 4.4 will be used. Thus, rigid-arm elements r_{3-6} and revolute joints h_{7-8} are selected into the T-tree, while the R-tree is comprised of the two mass elements $m_{1,2}$. The set of branch coordinates resulting from this tree selection is:

$$\{q\} = [\theta_1, \theta_2]^T$$

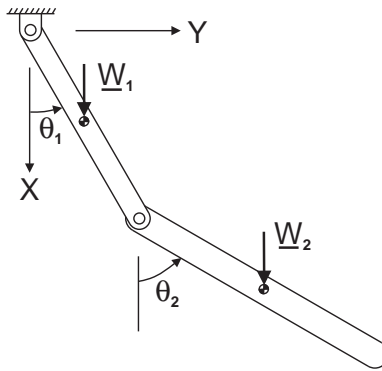


Figure 13: Double Pendulum

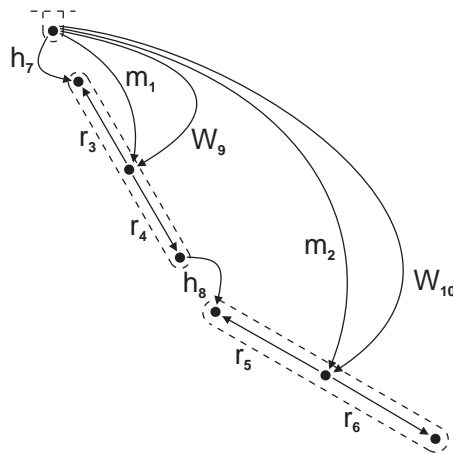


Figure 14: Linear Graph for Double Pendulum

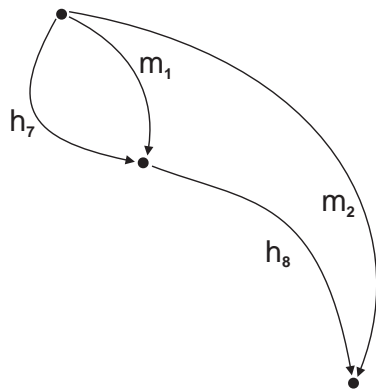


Figure 15: Rotational Graph for Double Pendulum

i.e. the set of “absolute angle coordinates” favoured by Huston [22]. Since the open-loop system has exactly 2 degrees of freedom, these branch coordinates will be independent variables not subject to any constraint equations. Thus, the final set of motion equations will consist of only 2 ODEs.

Step 1: There are no masses in the T-tree, and the T-cutsets for h_7 and h_8 will vanish when projected onto the z axis, representing the rotational motion space of these joints. Thus the only dynamic equations result from the projection of the R-cutset equations for m_1 and m_2 onto the z axis of rotation:

$$(\underline{T}_1 + \underline{T}_7 - \underline{T}_8 = 0) \cdot \hat{k} \quad (92)$$

$$(\underline{T}_2 + \underline{T}_8 = 0) \cdot \hat{k} \quad (93)$$

Upon substitution of the terminal equations from Table 4, these equations become:

$$(-I_1 \underline{\alpha}_1 - \underline{r}_3 \times \underline{F}_3 - \underline{r}_4 \times \underline{F}_4 = 0) \cdot \hat{k} \quad (94)$$

$$(-I_2 \underline{\alpha}_2 - \underline{r}_5 \times \underline{F}_5 - \underline{r}_6 \times \underline{F}_6 = 0) \cdot \hat{k} \quad (95)$$

Step 2: The only cotree joints are h_7 and h_8 in the R-cotree. However, the corresponding R-circuit equations will vanish when projected onto the xy reaction space for these joints. Thus, no constraint equations result from this Step, as expected.

Step 3: For the four rigid-arm elements in the T-tree, the following chord transformation equations are obtained:

$$\underline{F}_3 = \underline{F}_1 + \underline{F}_9 + \underline{F}_2 + \underline{F}_{10} \quad (96)$$

$$\underline{F}_4 = -\underline{F}_2 - \underline{F}_{10} \quad (97)$$

$$\underline{F}_5 = \underline{F}_2 + \underline{F}_{10} \quad (98)$$

$$\underline{F}_6 = 0 \quad (99)$$

or, after substitution of the terminal equations for force variables:

$$\underline{F}_3 = -m_1 \underline{a}_1 + \underline{W}_1 - m_2 \underline{a}_2 + \underline{W}_2 \quad (100)$$

$$\underline{F}_4 = m_2 \underline{a}_2 - \underline{W}_2 \quad (101)$$

$$\underline{F}_5 = -m_2 \underline{a}_2 + \underline{W}_2 \quad (102)$$

which are used to eliminate \underline{F}_{3-5} from the projected cutset equations (94-95).

Generating the branch transformation equations for bodies m_1 and m_2 in the T-cotree:

$$\underline{a}_1 = -\underline{a}_3 + \underline{a}_7 = 0 \quad (103)$$

$$\underline{a}_2 = -\underline{a}_5 + \underline{a}_8 + \underline{a}_4 - \underline{a}_3 + \underline{a}_7 = 0 \quad (104)$$

and after substitution of the corresponding terminal equations:

$$\underline{a}_1 = -\underline{\alpha}_1 \times \underline{r}_3 + \omega_1^2 \underline{r}_3 \quad (105)$$

$$\underline{a}_2 = -\underline{\alpha}_2 \times \underline{r}_5 + \omega_2^2 \underline{r}_5 + \underline{\alpha}_1 \times \underline{r}_4 - \omega_1^2 \underline{r}_4 - \underline{\alpha}_1 \times \underline{r}_3 + \omega_1^2 \underline{r}_3 \quad (106)$$

which are used to eliminate \underline{a}_1 and \underline{a}_2 from the chord transformation equations (100-102).

Step 4: Substituting equations (100-102) and (105-106) into the projected cutset equations (94-95), one obtains the following set of 2 ordinary differential equations of motion:

$$\begin{bmatrix} I_{11} & I_{12} \\ I_{21} & I_{22} \end{bmatrix} \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} = \{Q(q, \dot{q}, W_1, W_2)\} \quad (107)$$

where $\{Q\}$ is a complex function of the gravitational forces and quadratic velocity terms, while the entries in the mass matrix are:

$$I_{11} = I_1 + m_1 r_3^2 + m_2 (r_3^2 + r_4^2 - 2r_3 \cdot r_4) \quad (108)$$

$$I_{12} = m_2 r_5 \cdot (r_3 - r_4) = I_{21} \quad (109)$$

$$I_{22} = I_2 + m_2 r_5^2 \quad (110)$$

Physically, the term I_{22} represents the moment of inertia of the second body about the second pin joint h_8 , while I_{11} is the moment of inertia about the first pin joint h_7 of the “augmented body” [21] corresponding to m_1 . As predicted by Huston [22], the mass matrix appearing in the final motion equations (107) is simpler in form than that obtained using a joint coordinate formulation. This leads to faster computer simulations of the double pendulum response, obtained by numerically integrating the equations of motion for some given set of initial conditions.

4.6.2 Kinematic Equations for Quick-Return Mechanism

Shown in Figure 16 is a quick-return mechanism, the function of which is to transform a uniform rotational speed ω of the crank BC into a translation of the slider at E to the left, followed by a relatively rapid return to the right. The link lengths and joint locations given by Haug [14] have been used in this example, as well as a crank speed of 4π rad/s.

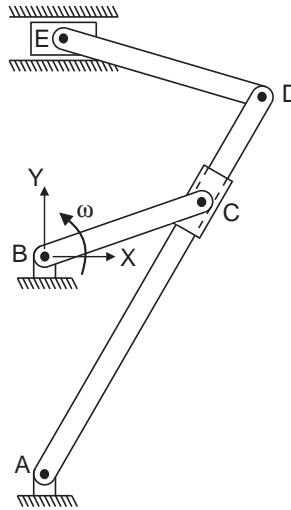


Figure 16: Quick-Return Mechanism

The translational and rotational graphs of the quick-return mechanism are shown in Figures 17 and 18, respectively; the individual bodies are drawn in dotted lines and the edges comprising the two trees are shown in bold. The ground-fixed position vectors R_{1-2} and body-fixed rigid-arms r_{3-8} are selected into the translational tree, as are the edges h_{14-18} representing the 5 pin joints. The slider joint elements s_{19-20} are selected into the rotational tree, as are the mass elements m_9 and m_{12} and the rotation driver ω_{21} . Note that m_9 and m_{12} have been selected instead of m_{10} and m_{13} , since the displacements of the rigid-arm elements r_{4-6} in the translational tree are implicit functions of θ_9 and θ_{12} ; by directly choosing θ_9 and θ_{12} into the set of branch coordinates, the number of subsequent substitutions to eliminate secondary variables are kept to a minimum.

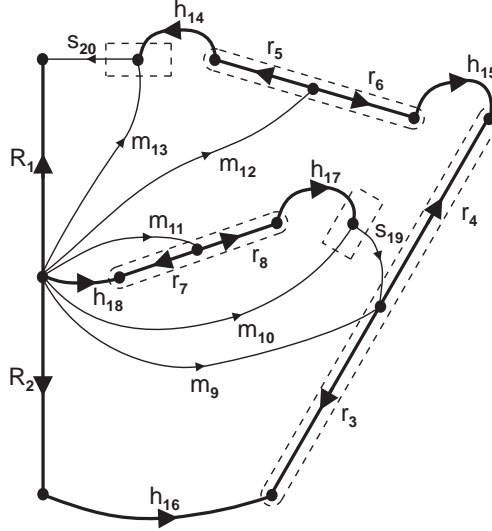


Figure 17: Translational Graph of Quick-Return

Note that the *only* branch coordinates resulting from this particular tree selection are θ_9 and θ_{12} , in terms of which the final set of kinematic equations will be automatically generated. In contrast, the use of absolute coordinates would result in 14 algebraic equations of constraint (2 being generated for each of the 7 joints), supplemented by the driving equation for the crank. Even the more efficient set of joint coordinates would require the driving equation plus 4 loop closure equations in order to solve for 5 coordinates. This is more than twice the number of equations that are generated by the graph-theoretic procedure.

Step 2: Since we are not interested in the dynamic response of the mechanism, generation of the cutset equations (Step 1) is skipped and the formulation procedure begins by projecting the T-circuit equations for the prismatic joints in the T-cotree onto their reaction spaces:

$$[r_{19} + r_3 - r_{16} - r_2 + r_{18} - r_7 + r_8 + r_{17}] \cdot \hat{n}_{19} = 0 \quad (111)$$

$$[r_{20} - r_1 + r_2 + r_{16} - r_3 + r_4 - r_{15} - r_6 + r_5 + r_{14}] \cdot \hat{n}_{20} = 0 \quad (112)$$

where \hat{n}_{19} and \hat{n}_{20} are the unit vectors normal to the axes of sliders s_{19} and s_{20} , respectively. Assuming that the axis of s_{19} is defined relative to body m_9 , the components of the unit vector \hat{n}_{19} are functions of the branch coordinate θ_9 , while \hat{n}_{20} is simply the unit vector in the negative Y direction, $-\hat{j}$.

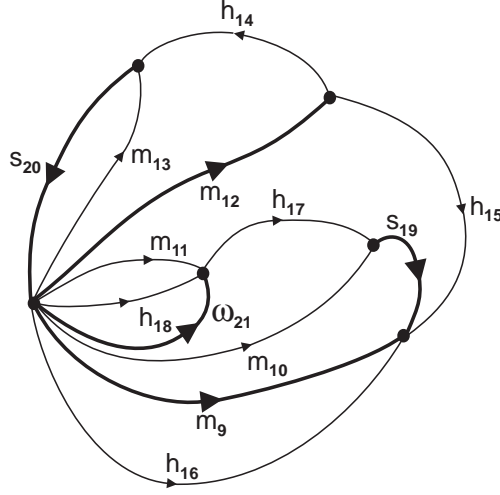


Figure 18: Rotational Graph of Quick-Return

Substituting the terminal equations from Table 4 into equations (111-112), one can write these equations in the functional form:

$$[r_3(\theta_9) - \underline{R}_2 + r_8(\theta_{11}) - r_7(\theta_{11})] \cdot \hat{n}_{19}(\theta_9) = 0 \quad (113)$$

$$[\underline{R}_2 - \underline{R}_1 + r_4(\theta_9) - r_3(\theta_9) + r_5(\theta_{12}) - r_6(\theta_{12})] \cdot \hat{n}_{20} = 0 \quad (114)$$

or, in the scalar form that is automatically generated by the Maple implementation [30] of the formulation procedure:

$$(l_7 + l_8) \sin \theta_9 \cos \theta_{11} - (l_7 + l_8) \cos \theta_9 \sin \theta_{11} - l_2 \cos \theta_9 = 0 \quad (115)$$

$$-l_1 - l_2 + (l_3 + l_4) \sin \theta_9 + (l_5 + l_6) \cos \theta_{12} = 0 \quad (116)$$

where l_i is the length of position vector r_i .

Step 3: The next step in the formulation procedure is to eliminate the secondary variable θ_{11} by generating the rotational (R) circuit equation for rigid body m_{11} in the R-cotree:

$$\theta_{11} = \theta_{21}(t) = 0.4356 + 4\pi t \quad (117)$$

which is used to replace θ_{11} in equations (115-116) with the specified driving function. An approximate solution of these 2 nonlinear equations for θ_9 and θ_{12} was then generated for 2 complete crank cycles using the Matlab procedure *fsolve*. These numerical results were then back-substituted into the analytical expression for the horizontal displacement of the slider s_{20} , obtained by projecting the corresponding T-circuit equation onto the slider axis defined by the unit vector \hat{u}_{20} :

$$[r_{20} - r_1 + r_2 + r_{16} - r_3 + r_4 - r_{15} - r_6 + r_5 + r_{14}] \cdot \hat{u}_{20} = 0 \quad (118)$$

which is reduced by the Maple procedures to simply:

$$s_{20} = (l_3 + l_4) \cos \theta_9 - (l_5 + l_6) \sin \theta_{12} \quad (119)$$

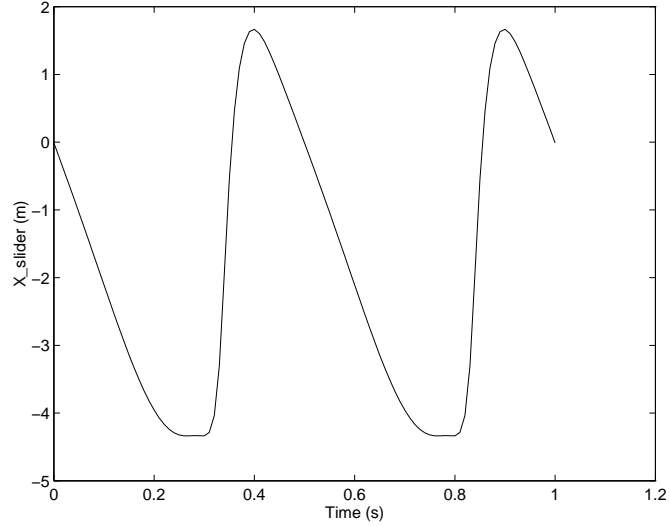


Figure 19: Kinematic Response of Quick-Return

This horizontal displacement has been plotted in Figure 19, which clearly shows the forward “working” stroke of the slider being followed by a short dwell period and a relatively quick return stroke.

It is interesting to note that equation (115) involves only one of the two unknown coordinates, θ_9 . Thus, this decoupled equation can be solved on its own for $\theta_9(t)$, after which equation (116) can be solved for $\theta_{12}(t)$ through simple back-substitution. For this example then, the number of constraint equations that need to be solved simultaneously has been reduced to the number of degrees of freedom. Furthermore, one can obtain closed-form expressions for θ_{11} and θ_{12} as explicit functions of θ_9 by analytically solving equations (115-116). These expressions can then be used in a dynamic analysis to reduce the equations of motion from a set of 5 DAEs to a single ODE. However, the complexity of the closed-form expressions, which contain inverse trigonometric functions and square roots, leads to a very large and very cumbersome ODE. Further discussion of the dynamic equations for the quick-return mechanism can be found in [30].

4.6.3 Dynamic Equations for Slider-Crank Mechanism

In this last example, the equations of motion will be derived for the slider-crank mechanism shown in Figure 10. To keep the number of these equations as small as possible, the tree of the translational graph shown in Figure 11 should consist of the position vector elements R_4 and r_{5-8} , as well as edges h_{9-11} representing the 3 revolute joints. With the same goal in mind, the prismatic joint s_{12} is selected into the tree of the rotational graph shown in Figure 12, along with the 2 rigid body elements m_{1-2} . The unknown branch coordinates corresponding to this tree selection are θ_1 and θ_2 , which constitute the set of variables in which the equations of motion will be generated.

Step 1: Projecting the R-cutset equations for m_1 and m_2 onto their motion space (Z -axis) yields:

$$[\underline{T}_1 + \underline{T}_9 - \underline{T}_{10}] \cdot \hat{k} = 0 \quad (120)$$

$$[\underline{T}_2 + \underline{T}_{10} + \underline{T}_{11}] \cdot \hat{k} = 0 \quad (121)$$

where \hat{k} is the unit vector in the Z direction. Substituting the terminal equations from Table 1 produces:

$$[-I_1 \underline{\alpha}_1 - \underline{r}_5 \times \underline{F}_5 - \underline{r}_6 \times \underline{F}_6] \cdot \hat{k} = 0 \quad (122)$$

$$[-I_2 \underline{\alpha}_2 - \underline{r}_7 \times \underline{F}_7 - \underline{r}_8 \times \underline{F}_8] \cdot \hat{k} = 0 \quad (123)$$

No other differential equations are generated, since the cutset equations for all other elements are orthogonal to their motion spaces.

Step 2: Projecting the T-circuit equation for the prismatic joint s_{12} onto its reaction space:

$$[\underline{r}_{12} - \underline{r}_4 + \underline{r}_9 - \underline{r}_5 + \underline{r}_6 + \underline{r}_{10} - \underline{r}_7 + \underline{r}_8 - \underline{r}_{11}] \cdot \hat{n}_{12} = 0 \quad (124)$$

where $\hat{n}_{12} = -\hat{j}$ defines the direction normal to the axis of slider joint s_{12} . Substituting the terminal equations from Table 4 into equation (124) and evaluating the dot products, one obtains the scalar equation:

$$\Phi(\theta_1, \theta_2) = (l_7 + l_8) \cos \theta_2 - (l_5 + l_6) \sin \theta_1 = 0 \quad (125)$$

which is the single constraint equation for the 2 coordinates being used to represent the configuration of this 1 degree-of-freedom system. Had a kinematic analysis been required, this single equation would be sufficient to solve for the time response of the system since either θ_1 or θ_2 would be replaced by some given function of time.

Step 3: Although the constraint equation (125) is in its final form, the secondary variables appearing in the dynamic equations (122-123) must be eliminated, starting with the rigid-arm forces. Generating the T-cutset equations for these elements, and substituting the terminal equations, one obtains:

$$\underline{F}_5 = -m_1 \underline{a}_1 - m_2 \underline{a}_2 - m_3 \underline{a}_3 + F_{13} \hat{i} + F_{12} \hat{j} \quad (126)$$

$$\underline{F}_6 = m_2 \underline{a}_2 + m_3 \underline{a}_3 - F_{13} \hat{i} - F_{12} \hat{j} \quad (127)$$

$$\underline{F}_7 = -m_2 \underline{a}_2 - m_3 \underline{a}_3 + F_{13} \hat{i} + F_{12} \hat{j} \quad (128)$$

$$\underline{F}_8 = m_3 \underline{a}_3 - F_{13} \hat{i} - F_{12} \hat{j} \quad (129)$$

The translational accelerations \underline{a}_{1-3} are then obtained by generating the T-circuit equations for m_{1-3} , into which the terminal equations for accelerations are substituted:

$$\underline{a}_1 = -\underline{\alpha}_1 \times \underline{r}_5 + \omega_1^2 \underline{r}_5 \quad (130)$$

$$\underline{a}_2 = -\underline{\alpha}_2 \times \underline{r}_7 + \omega_2^2 \underline{r}_7 + \underline{\alpha}_1 \times (\underline{r}_6 - \underline{r}_5) - \omega_1^2 (\underline{r}_6 - \underline{r}_5) \quad (131)$$

$$\underline{a}_3 = \underline{\alpha}_2 \times (\underline{r}_8 - \underline{r}_7) - \omega_2^2 (\underline{r}_8 - \underline{r}_7) + \underline{\alpha}_1 \times (\underline{r}_6 - \underline{r}_5) - \omega_1^2 (\underline{r}_6 - \underline{r}_5) \quad (132)$$

Equations (130-132) are used to eliminate \underline{a}_{1-3} from equations (126-129), which in turn are used to eliminate \underline{F}_{5-8} from the dynamic equations (122-123). It is clear that, in a numerical implementation of this formulation procedure, these substitutions could be executed very efficiently by exploiting the recursive nature of equations (126-132).

Step 4: After eliminating the secondary variables and simplifying the resulting scalar expressions, equations (122-123) are assembled into the matrix form:

$$[M] \begin{Bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{Bmatrix} + \{\Phi\}_q^T F_{12} = \{Q\} \quad (133)$$

where F_{12} is the unknown magnitude of the reaction force in the cotree prismatic joint s_{12} , and the entries in the mass matrix $[M]$ are:

$$M_{11} = I_1 + m_1 l_5^2 + (m_2 + m_3)(l_5 + l_6)^2 \quad (134)$$

$$M_{12} = (m_2 l_7 + m_3(l_7 + l_8))(l_5 + l_6)(c_1 s_2 - s_1 c_2) = M_{21} \quad (135)$$

$$M_{22} = I_2 + m_2 l_7^2 + m_3(l_7 + l_8)^2 \quad (136)$$

where $c_i = \cos \theta_i$ and $s_i = \sin \theta_i$. Note that M_{11} is the moment of inertia about the axis of pin joint h_9 of the “augmented body” [21] corresponding to m_1 , while M_{22} is the moment of inertia of the augmented body corresponding to m_2 about the axis of pin joint h_{10} . As expected, the coefficient matrix $\{\Phi\}_q$ of the unknown constraint force is the Jacobian of the constraint equation (125):

$$\{\Phi\}_q = \frac{\partial \Phi}{\partial \{q\}} = [-(l_5 + l_6)c_1 \quad -(l_7 + l_8)s_2] \quad (137)$$

Finally, the entries in the column matrix $\{Q\}$ are:

$$Q_1 = -F_{13}(l_5 + l_6)s_1 - (m_2 \omega_2^2 l_7 + m_3 \omega_2^2 (l_7 + l_8))(l_5 + l_6)(c_1 c_2 + s_1 s_2) \quad (138)$$

$$Q_2 = F_{13}(l_7 + l_8)c_2 + (m_2 \omega_1^2 l_7 + m_3 \omega_1^2 (l_7 + l_8))(l_5 + l_6)(c_1 c_2 + s_1 s_2) \quad (139)$$

To obtain the time response of the mechanism for this selection of branch coordinates, the 3 DAEs consisting of equations (125) and (133) must be solved using an appropriate numerical method for the 2 coordinates and the 1 constraint force. In contrast, use of absolute coordinates would have resulted in a much larger system of 9 differential equations (3 for each body) and 8 algebraic equations of constraint (2 for each joint), for a total of 17 DAEs. The more efficient set of joint coordinates would result in 3 differential equations and 2 algebraic equations for loop closure, for a total of 5 DAEs. Thus, the use of branch coordinates and the graph-theoretic formulation procedure presented in Section 4.5 results in a significant reduction of the number of dynamic equations of motion for the slider-crank mechanism. The same conclusion can be drawn for a wide variety of common mechanical systems containing one or more closed kinematic chains.

4.6.4 Dynamic Analysis of Squeezer Mechanism

The squeezer mechanism (a proposed design for an early impact printer) shown in Figure 20 is a popular dynamic analysis problem often used for benchmarking multibody analysis programs [5]. This system has one degree of freedom and consists of seven moving bodies, 11 pin joints, 1 translational spring, and a single applied torque driving the system.

The corresponding translational graph is made up of 36 elements and 26 nodes. The elements consist of bodies m_1 to m_7 ; pin joints h_8 to h_{17} ; translational spring k_{18} ; ground-fixed vectors R_{20} to R_{22} ; and body-fixed vectors r_{23} to r_{37} . The rotational graph has 18 elements and 8 nodes and can be obtained from the translational graph by excluding the body-fixed and ground-fixed vectors, and the translational spring. The driving torque T_{19} , which was excluded from the translational graph, is included in the rotational graph. For the sake of brevity, the two linear graphs and the physical parameters of the system components are not presented here; for complete details, see [32].

Starting from rest, the motion of the system was simulated using the **MeSyA** program for a period of 0.03 seconds, using a fixed integration time step of 0.00005 seconds and the following

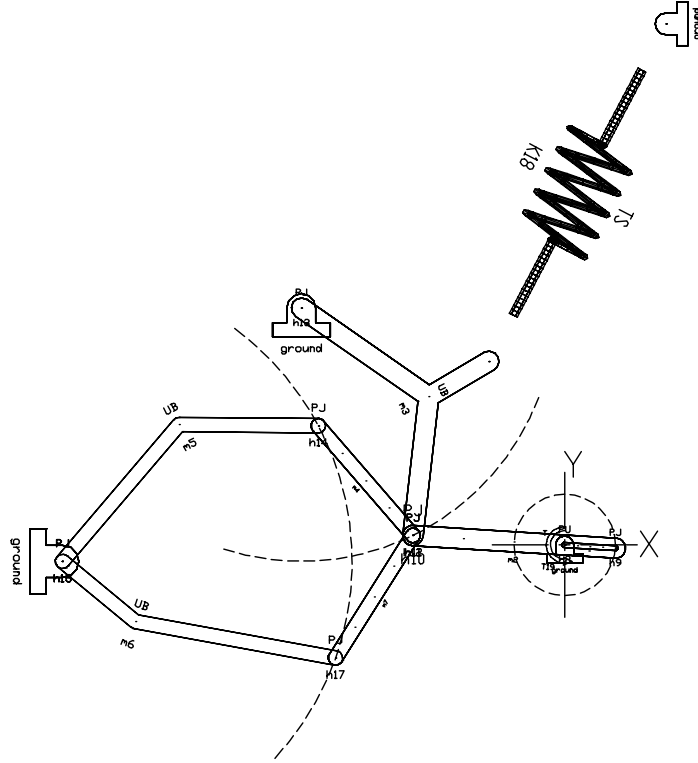


Figure 20: Squeezer Mechanism

initial conditions for the driving link:

$$\theta_1 = -0.062020 \text{ rad}; \dot{\theta}_1 = 0.0 \text{ rad/s.}$$

The time history profile of the driving link angular velocity is plotted in Figure 21. As expected, the overall velocity is seen to increase over the duration of the simulation, since the driving torque is adding energy to the undamped system. The spring provides either an accelerating or decelerating effect, depending on the angular position of the driving link.

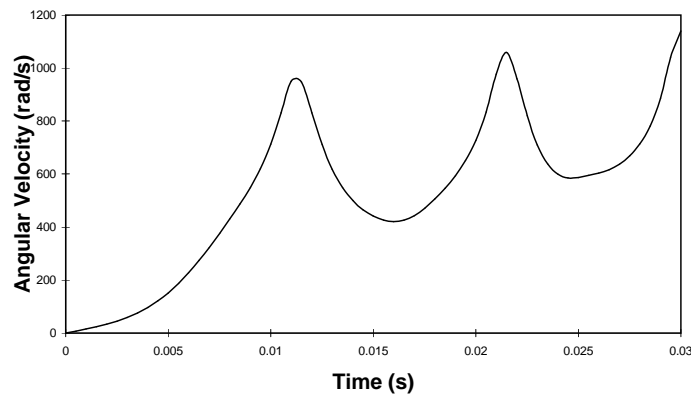


Figure 21: Squeezer Mechanism Driving Link Angular Velocity Profile

Table 5 shows a comparison of the number of DAEs and the computer simulation times obtained using three different tree selections. AC refers to the use of absolute coordinates, JC refers

to joint coordinates, and AAC refers to absolute angular coordinates, which result from the selection of mass elements into the R-tree. The relative execution times have been normalized with respect to the AC tree selection. As expected, this table demonstrates the improvement in simulation time due to the reduced number of equations obtained using the tree selection criteria of Section 4.4. It is also worth noting that a reduction in the number of constraint equations that have to be integrated will have an impact on the accuracy of simulation results due to the reduced amount of constraint violation incurred over the duration of a simulation.

Tree Type	Number of Equations	Normalized Execution Time
AC	41	1.00
JC	13	0.76
AAC	13	0.73

Table 5: Comparison of Results for Squeezer Mechanism

References

- [1] J.A. Bondy and U.S.R. Murty, **Graph Theory with Applications**, North-Holland, New York, 1976.
- [2] N.L. Biggs, E.K. Lloyd, and R.J. Wilson, **Graph Theory: 1736-1936**, Oxford University Press, 1976.
- [3] H.E. Koenig, Y. Tokad, and H.K. Kesavan, **Analysis of Discrete Physical Systems**, McGraw-Hill, 1967.
- [4] M. Chandrashekar, G. Savage, S. Birkett, and J. McPhee, **Graph-Theoretic Modelling: Four Decades of Development**, Technical Report, Systems Design Engineering, University of Waterloo, Canada, 1995.
- [5] W. Schiehlen, editor, **Multibody Systems Handbook**, Springer-Verlag, 1990.
- [6] G. Andrews, **Dynamics Using Vector-Network Techniques**, Technical Report, Mechanical Engineering, University of Waterloo, Canada, 1977.
- [7] M. Chandrashekar and G. Savage, **Engineering Systems: Analysis, Design, and Control**, Course Notes, Systems Design Engineering, University of Waterloo, Canada, 1993.
- [8] F. Lorenz, **Pluridisciplinary System Modelling**, Lorenz Simulation s.a., Belgium, 1995.
- [9] J. Wittenburg, *Topological Description of Articulated Systems*, **Computer-Aided Analysis of Rigid and Flexible Mechanical Systems**, edited by M. Pereira and J. Ambrósio, Kluwer Academic Publishers, pp.159-196, 1994.
- [10] G.C. Andrews, *The Vector-Network Model: A New Approach to Vector Dynamics*, **Mechanism and Machine Theory**, vol.10, pp.57-75, 1975.
- [11] P.E. Nikravesh, **Computer-Aided Analysis of Mechanical Systems**, Prentice-Hall, 1988.
- [12] R.E. Roberson and R. Schwertassek, **Dynamics of Multibody Systems**, Springer-Verlag, 1988.
- [13] A.A. Shabana, **Dynamics of Multibody Systems**, Wiley, 1989.
- [14] E.J. Haug, **Computer-Aided Kinematics and Dynamics of Mechanical Systems**, Allyn and Bacon, 1989.
- [15] R.L. Huston, **Multibody Dynamics**, Butterworth-Heinemann, 1990.
- [16] F.M.L. Amirouche, **Computational Methods in Multibody Dynamics**, Prentice-Hall, 1992.
- [17] J. García de Jalón and E. Bayo, **Kinematic and Dynamic Simulation of Multibody Systems**, Springer-Verlag, 1994.

- [18] T.W. Li and G.C. Andrews, *Application of the Vector-Network Method to Constrained Mechanical Systems*, **Journal of Mechanisms, Transmissions, and Automation in Design**, vol.108, pp.471-480, 1986.
- [19] G.C. Andrews, M.J. Richard, and R.J. Anderson, *A General Vector-Network Formulation for Dynamic Systems with Kinematic Constraints*, **Mechanism and Machine Theory**, vol.23, pp.243-256, 1988.
- [20] G. Baciuc and H.K. Kesavan, *Graph-Theoretic Modeling of Particle-Mass and Constrained Rigid Body Systems*, **Mechanism and Machine Theory**, vol.30, pp.953-967, 1995.
- [21] J. Wittenburg, **Dynamics of Systems of Rigid Bodies**, B.G. Teubner, Stuttgart, 1977.
- [22] R.L. Huston, Y.S. Liu, and C. Liu, *Use of Absolute Coordinates in Computational Multibody Dynamics*, **Computers and Structures**, vol.52, no.1, pp.17-25, 1994.
- [23] J.J. McPhee, *A Unified Graph-Theoretic Approach to Formulating Multibody Dynamics Equations in Absolute or Joint Coordinates*, **Journal of the Franklin Institute**, vol.334B, no.3, pp.431-445, 1997.
- [24] 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, no.6, pp.805-823, 1998.
- [25] D. Scott, *Can a Projection Method of Obtaining Equations of Motion Compete with Lagrange's Equations?*, **American Journal of Physics**, vol.56, no.5, pp.451-456, 1988.
- [26] W. Blajer, *A Projection Method Approach to Constrained Dynamic Analysis*, **Journal of Applied Mechanics**, vol.59, pp.643-649, 1992.
- [27] A. Kecskeméthy and M. Hiller, *An Object-Oriented Approach for an Effective Formulation of Multibody Dynamics*, **Computer Methods in Applied Mechanics and Engineering**, vol.115, pp.287-314, 1994.
- [28] P.H. Todd, *A Constructive Variational Geometry Based Mechanism Design Software Package*, **Mechanical Design and Synthesis**, ASME DE-Vol.46, pp.267-273, 1992.
- [29] P.E. Nikravesh and G. Gim, *Systematic Construction of the Equations of Motion for Multibody Systems Containing Closed Kinematic Loops*, **Advances in Design Automation**, ASME DE-Vol.19-3, pp.27-33, 1989.
- [30] J.J. McPhee and C.E. Wells, *Automated Symbolic Analysis of Mechanical System Dynamics*, **MapleTech**, vol.3, no.1, pp.48-56, 1996.
- [31] O.M. Oshinowo and J.J. McPhee, *Object-Oriented Implementation of A Graph-Theoretic Formulation for Planar Multibody Dynamics*, **International Journal for Numerical Methods in Engineering**, vol.40, pp.73-90, 1997.

- [32] O.M. Oshinowo, **Automated Analysis of Planar Mechanical Systems with User-Defined Coordinates: A Graph-Theoretic Approach**, M.A.Sc. Thesis, University of Waterloo, Canada, 1996.
- [33] J.J. McPhee, *Modelling of Spatial Mechanical Systems using Linear Graph Theory and "Branch Coordinates"*, presented at 6th Conference on Nonlinear Vibrations, Stability, and Dynamics of Structures, Virginia Polytechnic Institute, U.S.A., 1996.
- [34] R. Schwertassek and A. Eichberger, *Recursive Generation of Multibody System Equations in Terms of Graph-Theoretic Concepts*, **Mechanics of Structures and Machines**, vol.17, no.2, pp.197-218, 1989.
- [35] L. Chan, *A Recursive Graph-Theoretic Method for Analyzing Planar Robotic Manipulators*, **Internal Report**, University of Waterloo, Canada, 1996.
- [36] B.J. Muegge, **Graph-Theoretic Modelling and Simulation of Planar Mechatronic Systems**, M.A.Sc. Thesis, University of Waterloo, Canada, 1996.
- [37] J.J. McPhee, *On the Use of Linear Graph Theory in Multibody System Dynamics*, **Nonlinear Dynamics**, vol.9, pp.73-90, 1996.
- [38] J.J. McPhee, M.G. Ishac, and G.C. Andrews, *Wittenburg's Formulation of Multibody Dynamics Equations from a Graph-Theoretic Perspective*, **Mechanism and Machine Theory**, vol.31, no.2, pp.201-213, 1996.
- [39] A.E. Baumal, J.J. McPhee, and P.H. Calamai, *Design Optimization of Active Vehicle Suspension Systems using Genetic Algorithms*, **Computer Methods in Applied Mechanics and Engineering**, vol.163, pp.87-94, 1998.