Unified Modelling Theories for the Dynamics of Multidisciplinary Multibody Systems

John McPhee

Systems Design Engineering, University of Waterloo, Canada
mcphee@real.uwaterloo.ca

1 Introduction

A major goal of research in multibody dynamics is to develop formulations that automatically generate and solve the governing equations of motion for a system of rigid and flexible bodies, given only a description of the system. Much progress has been made in multibody dynamics research over the last few decades and nowadays, there are several commercially-successful computer programs (e.g. Adams, Dads) that automatically analyze the dynamics of multibody mechanical systems.

However, these same programs, and the theoretical formulations on which they are based, are not capable of modelling general multidisciplinary applications in which a multibody system is coupled to other physical domains, e.g. electrical or pneumatic. There are numerous important applications of multidisciplinary multibody systems, including vehicles with active suspensions and traction control, mechatronic systems, and micro-electromechanical systems (MEMS). The design of these multidisciplinary applications would be greatly facilitated by algorithms that could automate their dynamic analysis.

There are two distinct approaches that have been proposed for modelling and simulating the dynamics of multidisciplinary multibody systems. The first is based on coupled simulations, or "co-simulation", in which two separate simulation programs or subroutines are coupled numerically. The advantage of co-simulation is that one can use existing programs that are very well-developed for their particular domain. However, numerical stability problems may arise during a co-simulation [20] and, more importantly, there is no underlying mathematical framework that one could use to generate analytical models of these multidisciplinary applications.

The second approach is to apply a unified systems theory to the dynamic modelling of multidisciplinary multibody systems. This paper focuses on this second approach for several reasons:

- a unified theory leads to analytical models that promote physical insight
- a unified theory can be applied manually or implemented in a computer algorithm
- the computer implementation can be symbolic or numeric

J.A.C. Ambrósio (ed.), Advances in Computational Multibody Systems, 125–154.

- symbolic models are very appropriate for real-time simulation
- symbolic models facilitate design optimization and sensitivity calculations
- symbolic models are easily communicated between colleagues and to students
- a unified theory can be extended to new situations and domains

Two main systems theories dominate the literature: linear graph theory and bond graph theory. Note that linear graph theory is sometimes abbreviated to "graph theory", especially in the literature on electrical circuits. A few authors [17, 22] have also proposed the principle of virtual work as the basis of a third unified theory, but these authors are forced to adopt elements of graph theory in their formulations, e.g. when generating topological equations for electrical subsystems. Thus, virtual work on its own does not constitute a complete and independent systems theory that can be used to automate the dynamic analysis of complex systems.

The goal of this paper is to present the modelling of multidisciplinary multibody systems using bond graph theory and linear graph theory, and to investigate their relative advantages and disadvantages.

2 Representative Problems

The features of modelling with bond graphs and linear graph theory will be demonstrated by means of four example problems, ranging from quite simple to very complex. Three of the examples are multidisciplinary applications, with components from the mechanical and electrical domains, and the last three examples contain multi-dimensional multibody subsystems.

2.1 Condenser Microphone

A simple model of a condenser microphone [3, 4] is shown in Figure 1. A voltage source E_4 is connected in series with a resistor R_1, capacitor C_2, and inductor L_3. The resistor and inductor are modelled by standard linear constitutive equations [10], but the upper plate of the capacitor is free to move, requiring extra consideration. This upper plate represents the mass (m_5), stiffness (k_6), and damping (d_7) of the mechanical portion of the microphone. For completeness, the gravitational force on the upper plate has also been included in the model.

Due to the electrical attraction of the two plates, a voltage v_2 across the capacitor results in an attractive force F_2 between the two plates given by [3]:

$$F_2 = -\frac{1}{2}\frac{dC_2}{dx}v_2^2 \tag{1}$$

where the capacitance C_2 is a function of the plate separation x; it decreases as x increases, i.e. $dC_2/dx < 0$, giving a positive force of attraction in equation (1).

The second constitutive equation for the capacitor is:

$$i_2 = C_2\frac{dv_2}{dt} + \frac{dC_2}{dx}\frac{dx}{dt}v_2 \tag{2}$$

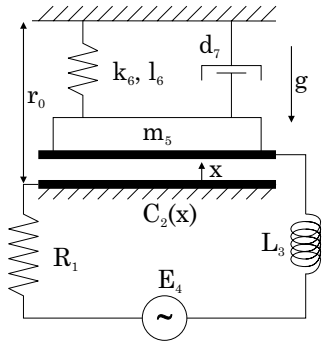


Fig. 1. Simple model of condenser microphone

where i_2 is the current through the capacitor; the second term on the right-hand side is induced by the relative motion of the two plates. Note that there are usually two constitutive equations for transducers, such as this moving-plate capacitor, that couple one physical domain (electrical) to a second (mechanical). It is through these transducers that energy can flow between the two domains.

In this condensor microphone, vibrations of the plate result in an electrical current that can be measured and amplified, if necessary. The goal of this example is to generate the system equations that relate the motion of the plate to the current.

2.2 Inverted double pendulum

This example, taken from Karnopp et al [5], is a planar mechanical multibody system. As shown in Figure 2, it consists of a horizontally-translating mass m, upon which an inverse double pendulum is mounted. The two links of this pendulum have lengths l_1 and l_2, and the masses m_1 and m_2 are assumed to be concentrated at the tips of the links. Gravity acts vertically downwards.

The purpose of this example is to demonstrate some features of multi-dimensional multibody systems, and the application of bond graphs and linear graph theory to these systems.

To automate the dynamic analysis of a multibody system, one must develop a formulation that can express all kinematic quantities in terms of a general set of

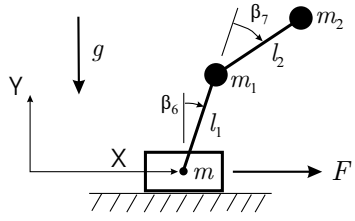


Fig. 2. Inverted double pendulum

coordinates, and generate the dynamic equations in terms of these coordinates. By far, the two most popular sets are absolute (Cartesian) coordinates and relative (joint) coordinates, although other possibilities do exist [23].

Absolute coordinates lead to relatively simple computer implementations, which is why they are used in commercially-successful packages such as Adams, Dads, and Working Model. These coordinates, which represent the position and orientation of every body in the system, will not be independent if there are any holonomic joints in the system. In that case, the n coordinates $\mathbf{q}$ are related by m nonlinear algebraic equations:

$$\mathbf{\Phi}(\mathbf{q}, t) = \mathbf{0} \tag{3}$$

where $n - m = f$, the degrees of freedom of the system. The dynamic equations are easily generated from free-body diagrams of each body, with the constraint reactions represented by Lagrange multipliers $\boldsymbol{\lambda}$:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{F} \tag{4}$$

where $\mathbf{M}$ is the constant $n \times n$ mass matrix, $\mathbf{\Phi}_{\mathbf{q}}$ is the Jacobian matrix of the constraint equations (3), and $\mathbf{F}$ contains external forces and quadratic velocity terms. Equations (3) and (4) constitute a set of $n + m$ differential-algebraic equations (DAEs) that can be solved for $\mathbf{q}(t)$ and $\boldsymbol{\lambda}(t)$.

Joint coordinates are more difficult to implement in an automated formulation, because one must pay more attention to topological processing [24]. However, the reward is fewer equations to solve than those expressed in absolute coordinates. For open-loop systems, i.e. those having no closed kinematic chains, the joint coordinates are independent and equal in number to the degrees of freedom f. There are no kinematic constraint equations (3) to satisfy, and a minimal set of f ordinary differential equations (ODEs) is obtained for the dynamics:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{F} \tag{5}$$

where the $f \times f$ mass matrix $\mathbf{M}$ is now a function of $\mathbf{q}$.

Since the inverted double pendulum is an example of an open-loop system, it is best modelled by joint coordinates.

2.3 Robot Manipulator

In this example, an experimental two-link robot manipulator is modelled. An overview of the experimental system is shown in Figure 3. It consists of two DC motors, a shoulder motor and an elbow motor, and two links. The interchangeable links may be rigid or flexible; in this example, we model the manipulator with rigid links. The manipulator is supported by air bearings on a large glass surface so as to minimize friction and gravitational effects. The shoulder motor may also be fixed to the glass surface by a vacuum, which is the case considered in this example. A variety of sensors are used to track the motion of the manipulator and a moving payload. These

sensors provide feedback to a microcomputer that is responsible for generating control signals to the two motors. More details of the WatFlex experimental manipulator[1] can be found in [21].

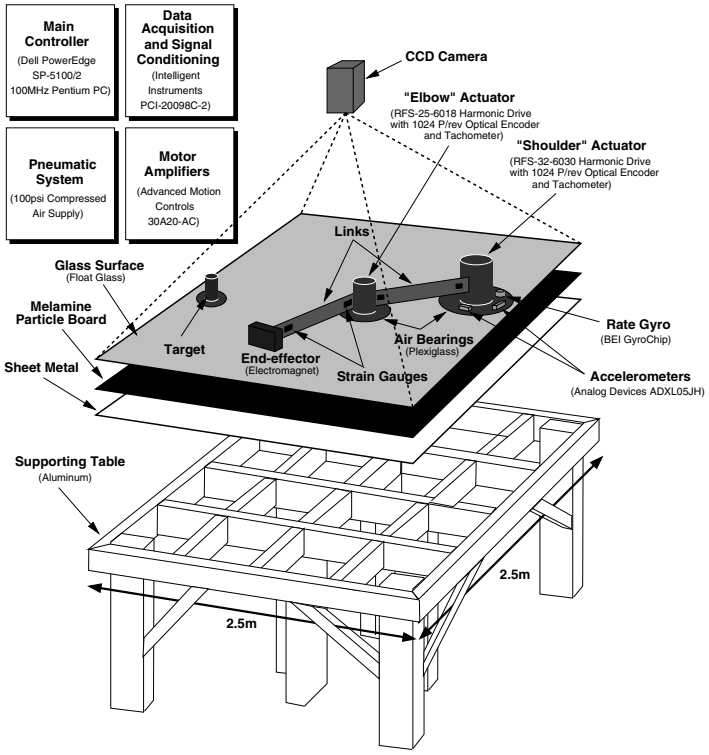


Fig. 3. Watflex robot manipulator

Shown in Figure 4 is a model of the two-link manipulator. We have chosen to include gravity (g) in the model, by re-orienting the robot such that the links rotate about horizontal joint axes. The first joint angle is designated θ_1, while the second is θ_{1-2}. Therefore, the absolute rotation θ_2 of the second link is the sum of the two joint angles.

The objective is to control the robot to follow a prescribed joint trajectory, specifically a rotation of both joints by 90 degrees in 4 seconds at constant angular speeds. These desired joint angles are input to a PD-controller that computes the differences between the desired and actual angles. The controller then supplies the two DC motors with voltages that are proportional to these differences (errors), and the time derivatives of these errors (i.e. the differences between desired and actual angular speeds).

[1] http://real.uwaterloo.ca/~watflex

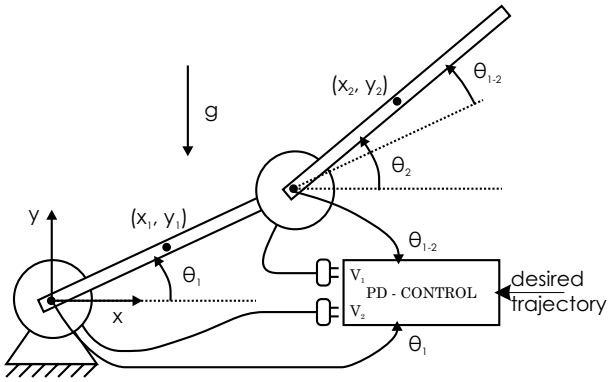


Fig. 4. PD-controlled robot manipulator

Similar to the moving-plate capacitor in the previous example, there are two constitutive equations for each DC motor in the model:

$$v = K_v \frac{d\theta}{dt} + Ri + L\frac{di}{dt} \tag{6}$$

$$T = K_T i - B\frac{d\theta}{dt} \tag{7}$$

where K_v is the voltage (back emf) constant, L and R are the armature inductance and resistance, respectively, K_T is the torque constant, and B is the coefficient of viscous friction in the bearings. The masses and inertias of the stator and rotor are assumed to be lumped with the mechanical component to which they are affixed.

2.4 Parking Gate System

In this fourth example, another electromechanical multibody system is considered. This system is used to raise and lower a flexible barrier, which is typically used to control access to a parking lot. The flexible barrier is 3 m long, and is rigidly connected to link P_2O_2 of the Watt-II six-bar mechanism [8] shown in Figure 5. A spring is used to counter-balance the weights of the moving links and flexible barrier, and an asynchronous 3-phase induction motor is used to drive the input link of the six-bar mechanism. More details of this system may be found in [13].

As shown in Figure 6, the induction motor is in a star-star configuration with a short-circuited rotor. Mutual inductance effects arise within each circuit, and also between the stator and rotor components — which is what converts the electrical currents into a driving torque.

The position of the rotor θ influences these mutual inductance effects, as can be seen from the constitutive equation for the motor torque T:

$$T = \frac{1}{2p}\sum_{j=1}^{p}\sum_{k=1}^{p}\frac{\partial}{\partial\theta}\left(\mathbf{M}_{jk} i_j i_k\right) \tag{8}$$

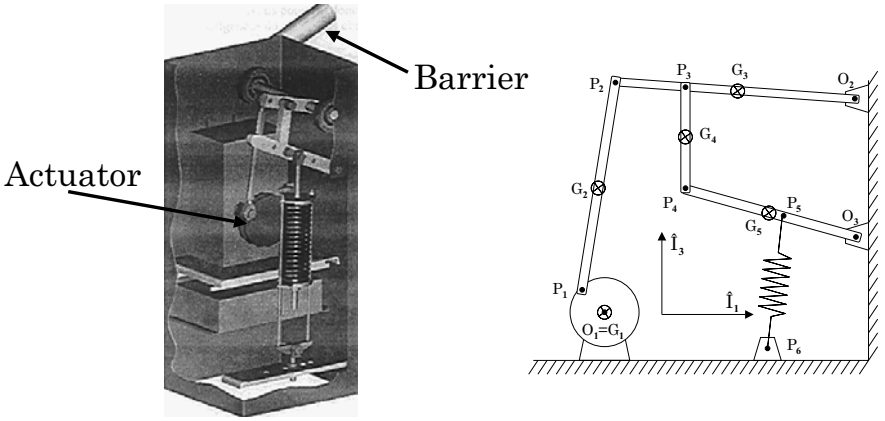


Fig. 5. Parking gate system

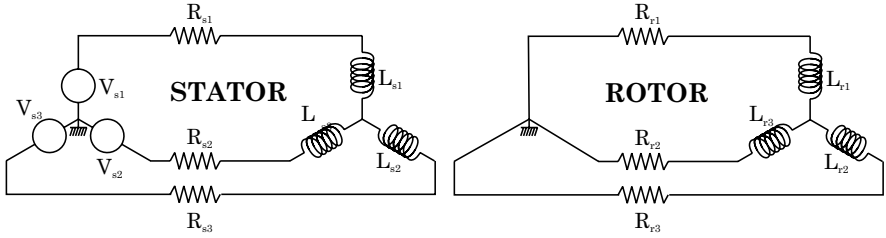


Fig. 6. Equivalent circuit for 3-phase actuator

where $p = 3$ is the number of pairs of poles, i_j is the current in inductor j, the index j ranges over the three inductors (L_{s1}, L_{s2}, L_{s3}) in the stator, k ranges over the three rotor inductors (L_{r1}, L_{r2}, L_{r3}), and $\mathbf{M}_{jk}$ is the entry in the jth row and kth column of the mutual inductance matrix defined by:

$$\mathbf{M} = M_{sr} \begin{pmatrix} \cos(\theta_{em}) & \cos\left(\theta_{em} + \frac{2\pi}{3}\right) & \cos\left(\theta_{em} + \frac{4\pi}{3}\right) \\ \cos\left(\theta_{em} + \frac{4\pi}{3}\right) & \cos(\theta_{em}) & \cos\left(\theta_{em} + \frac{2\pi}{3}\right) \\ \cos\left(\theta_{em} + \frac{2\pi}{3}\right) & \cos\left(\theta_{em} + \frac{4\pi}{3}\right) & \cos(\theta_{em}) \end{pmatrix} \tag{9}$$

where θ_{em} is the electromechanical position given by $\theta_{em} = p\theta$, and M_{sr} is the external mutual inductance between the stator and rotor.

The voltage v_i induced in each stator and rotor inductance is given by the second constitutive equation:

$$v_i = \sum_{j=1}^{p} \mathbf{L}_{ij} \frac{di_j}{dt} - \sum_{k=1}^{p} \frac{d}{dt} (\mathbf{M}_{ik} i_k) \tag{10}$$

where $\mathbf{L}_{ij}$ is the entry in the ith row and jth column of the inductance matrix of the stator or rotor in which inductor i resides:

$$\mathbf{L_s} = \begin{pmatrix} L_s & M_s & M_s \\ M_s & L_s & M_s \\ M_s & M_s & L_s \end{pmatrix} \qquad \mathbf{L_r} = \begin{pmatrix} L_r & M_r & M_r \\ M_r & L_r & M_r \\ M_r & M_r & L_r \end{pmatrix} \tag{11}$$

where L_s and L_r are the stator and rotor self-inductances, respectively, and M_s and M_r are the internal mutual inductances of the stator and rotor. In equation (10), the index j ranges over the inductors that are fixed with respect to the given inductor i, while the index k ranges over the inductors that are in relative motion. Thus, the first summation includes self-inductance terms, while the second summation represents the mutual inductance between stator and rotor components, which depends on the relative angle θ.

The flexible barrier is initially in a horizontal position, with no currents in the stator and rotor. A dynamic analysis and simulation is required to determine the response of the parking gate system to a sinusoidal input voltage (220V, 50Hz) over a period of 2 seconds.

3 Unified Modelling Theories

A perusal through monographs on "system dynamics" [5, 11] reveals that two unified modelling theories have become firmly established over the last few decades.

The first, linear graph theory, was invented by Leonhard Euler in 1736 to solve the famous Königsberg bridge problem [1]. Kirchoff applied graph theory to his analysis of electrical networks in the 1850s, and the generality of the graph theory approach in all physical domains was established by Trent [32] in 1955. The application of linear graph theory to physical system modelling is now well-established [6, 10, 11]. For multidisciplinary applications, the coupling between the different physical domains is not explicitly shown by the graph, but is embedded in the constitutive equations for the coupling elements (transducers).

The second unified modelling theory, bond graph theory, was invented by Henry Paynter [9] as an alternative notation in which energy flows between different physical domains are explicitly represented in the graph. Bond graphs were slow to be accepted by the engineering community, "mostly because of rather hazy mathematical underpinnings" [26]. Birkett and Roe [14], among others, have published a series of papers to address the lack of a combinatorial foundation for bond graphs. Nowadays, bond graphs are very well-known [2, 5, 12] and applied to multidisciplinary problems in both academia and industry.

Some authors [26] have stated that bond graphs are a special case of oriented linear graphs, while others [14] have concluded that bond graphs and linear graphs are distinct special cases of matroids. Regardless of the relationship between bond graph theory and linear graph theory, their representation of and application to multidisciplinary problems is very different in practice, especially for multibody systems.

In the following, a brief overview of bond graphs and linear graph theory is given, and their differences are highlighted through their application to the four example problems.

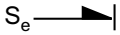
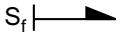
3.1 Bond Graph Theory

To develop a bond graph or linear graph model of a physical system, it is first decomposed into a finite number of discrete components. In bond graph theory, one distinguishes between "one-port" elements that connect to other components at two locations, and "two-port" elements that have four connection points. A one-port element is represented graphically by a single stroke, or bond, while a two-port element is depicted by two bonds. It is through these bonds that power flows through the system model; the two-port elements can provide an explicit representation of the energy transfer between different physical domains in a multidisciplinary application.

Associated with each bond is an effort (e) and flow (f) variable, the product of which gives power. For electrical components, the effort and flow corresponds to voltage and current, respectively. For mechanical systems, the effort and flow are usually taken as force and velocity, respectively, in what is known as the "force-effort" analogy [12]. In the less popular "force-flow" analogy, forces are flows and velocities are efforts.

By defining these generalized variables, and components with generalized constitutive equations in terms of these variables, one can develop bond graphs that represent multiple physical domains. The set of generalized one-port components includes resistances (R), inertias (I), capacitances (C), effort sources (S_e), and flow sources (S_f). A resistance, through which energy is lost from the system, can model an electrical resistor or a mechanical viscous damper. Similarly, an inertia can model an electrical inductor or a mechanical inertia, in the force-effort analogy. In the force-flow analogy, an inertia represents a mechanical spring. A summary of the bond graphs and constitutive equations for generalized one-port components is shown in Table 1

Table 1. Bond graphs and constitutive equations for generalized one-port components

Component	Bond graph	Equation
Effort source		$e = e(t)$
Flow source		$f = f(t)$
Resistance		$e = e(f)$
Capacitance		$f = C\frac{de}{dt}$
Inertia		$e = I\frac{df}{dt}$

Note that the bond graphs shown in Table 1 have a half-arrow associated with them. This is to identify the direction used to measure positive power flow, a convention established by the modeller for each system. Usually, one tries to predict positive power flows, starting from energy sources and flowing to loads. Thus, the arrow is directed away from the effort and flow sources in Table 1. Since resistances dissipate

power, usually in the form of heat, the arrow is directed towards the R component. Finally, power may flow to and from the capacitances and inertias; in Table 1, power flowing into these components (i.e. being stored) is chosen as positive.

Also shown in Table 1 are vertical bars at one end of each bond. This bar is known as a "causal" stroke, and is used to assign causality to a model, i.e. which of the efforts and flows are causes, and which are effects. The convention is that the causal stroke indicates the direction of effort, with flow acting in the reverse direction. Thus, the causal stroke is on the right side of the effort source in Table 1, and on the left side of the flow source. With linear R elements, there is no preferred causality; one may interpret the effort as causing the flow, or vice-versa. With linear inertias, the flow is proportional to the time integral of effort. The causal direction shown in Table 1 for the I element implies that effort is input to the element, and flow is the output. Thus, flow is being calculated from effort, which requires an integration. This is known as "integral causality"; the converse is derivative causality [12].

The assignment of causality to a bond graph model is used to facilitate the computation of the system equations. Causality conflicts can also reveal a fundamental modelling problem. By obtaining integral causality for a bond graph model, the governing equations will take the form of ordinary differential equations that can be readily solved using numerical integration methods. Derivative causality is to be avoided, due to the error-prone nature of computations involving numerical differentiation.

There are two basic types of two-port elements: transformers and gyrators [12]. In a transformer, the efforts in the two bonds are proportional, with the ratio known as the transformer modulus. Since transformers are energy-conserving elements, i.e. the power in one bond equals the power in the other, the flows in the two bonds must be equal to the reciprocal ratio. For gyrators, the effort in one bond is proportional to the flow in the second, and vice-versa for power conservation. The proportionality constant is the gyrator modulus. In modulated transformers and gyrators, the moduli are time-varying inputs.

From the topology of the physical system being modelled, the bond graphs of the discrete components are combined using 0 and 1 junctions, where 0 represents a parallel connection, and 1 represents a series connection. For all components connected by a 0 junction, all efforts are equal and all flows must sum to zero, taking into account the direction assigned to the half-arrows. For all components connected by a 1 junction, all flows are equal and all efforts sum to zero. Thus, the 0 and 1 junctions provide the topological equations for a given system.

By combining the topological equations with the constitutive equations, one has a necessary and sufficient set of equations to generate a complete system model. The bond graph approach is unified, since it can easily handle multidisciplinary applications, and very systematic. Thus, it is quite amenable to computer implementation and several computer programs have been developed using bond graph theory as their basis.

Full details of bond graph theory may be found in [5], [12], and [14]. In the following, the application of bond graph theory to the four examples is presented and discussed.

3.1.1 Bond Graph Model of Microphone

Shown in Figure 7 is a bond graph representation of the condenser microphone shown in Figure 1. A single bond graph provides a unified representation of the electrical and mechanical domains making up this multidisciplinary application. Note the use of the colon (:) notation in the figure to show the parameters associated with a generalized component model. Also, the effort and flow variables are shown explicitly for some of the bonds, with effort above the bond and the flow variable below.

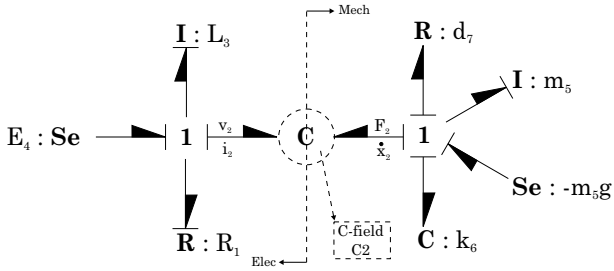


Fig. 7. Bond graph model of condenser microphone

The electrical circuit is modelled on the left using a resistance (R_1), capacitance (C_2), an inertia (L_3), and an effort source for the voltage driver (E_4). These four electrical components are connected in series, i.e. by a 1 junction. The assigned power flow directions are consistent with those shown in Table 1.

The mechanical subsystem is modelled on the right using a resistance for the damper (d_7), a capacitance for the spring (k_6), an inertia for the mass (m_5), and an effort source for the weight. The components in the physical system are connected in parallel; however, we are using the conventional force-effort analogy which requires that the bond graph components be connected by a series 1 junction. This is somewhat counter-intuitive, but it is needed to obtain the correct topological equations for the force-effort analogy. Using the force-flow analogy, parallel and series mechanical connections would be represented by parallel 0 and series 1 junctions, respectively. Regardless of whether one uses a force-effort or force-flow analogy, the bond graph does not bear much resemblance to the physical system.

The bond graphs for the mechanical and electrical domains are explicitly coupled by the capacitance element. In this case, the moving-plate capacitor is modelled as a "C-field" two-port component with the mechanical effort $e_2 = F_2$ and electrical flow $f_2 = i_2$ defined by the constitutive equations (1) and (2), respectively.

The system equations are derived very systematically from the bond graph. One starts with the topological equations for efforts from the 1 junctions:

$$v_4 - v_1 - v_3 - v_2 = 0 \tag{12}$$

$$F_2 + F_5 + F_6 + F_7 - F_g = 0 \tag{13}$$

where v_i and F_i is the voltage or force for component i, and F_g is the weight. The corresponding topological equations for flows are:

$$i_1 = i_3 = i_4 = i_2 \tag{14}$$

$$\dot{x}_2 = \dot{x}_6 = \dot{x}_7 = \dot{x}_g = \dot{x_5} = \dot{x} \tag{15}$$

Combining the topological equations (12-15) with the constitutive equations for components, and re-arranging, one obtains:

$$C_2'\dot{x}\left(-R_1 i_2 - L_3\frac{di_2}{dt} + E_4(t)\right) + C_2\left(-R_1\frac{di_2}{dt} - L_3\frac{d^2 i_2}{dt^2} + \dot{E}_4\right) = i_2 \tag{16}$$

$$m_5\ddot{x} + d_7\dot{x} + k_6 x + m_5 g = \frac{1}{2}C_2'\left(-R_1 i_2 - L_3\frac{di_2}{dt} + E_4(t)\right)^2 \tag{17}$$

where $C_2' \equiv \frac{dC_2(x)}{dx}$. These two ODEs are equivalent to those derived by hand in [17], and can be solved for the capacitor current $i_2(t)$ and plate separation $x(t)$.

3.1.2 Bond Graph Model of Inverted Double Pendulum

When one switches from one-dimensional systems to multi-dimensional mechanical systems, the bond graph representation becomes considerably more complex. Consider the bond graph model of the inverted double pendulum, taken from [5] and shown in Figure 8. This bond graph requires some explanation since it bears little resemblance to the physical system in Figure 2.

The bond graph model consists mainly of one-port elements, plus two multi-port modulated transformers (MTFs). Although the system has only 3 degrees of freedom, it is modelled by 7 absolute speeds: $\dot{x}$ for the sliding mass, and $\dot{x}, \dot{y}, \dot{\theta}$ for each of the two links. Associated with each of these speeds is an inertia in the bond graph, e.g. the moment of inertia J_1 corresponding to $\dot{\theta}_1$. Note that $\theta_1 = \beta_6$, the coordinate for the first revolute joint in Figure 2, and $\theta_2 = \beta_6 + \beta_7$. Gravity and the applied force F are modelled as effort sources in the $-y$ and x directions, respectively.

Clearly, there are a lot of bonds making up the model of this relatively simple mechanical system. This is due to the fact that bond graphs were designed to operate on scalar variables. Hence, there is one bond for each of the 7 absolute speeds. One can combine the speeds associated with a given body into a single "multibond graph" [31] to obtain a simpler graph representation. However, the speeds are still represented by a column matrix of scalar variables, and not as a frame-invariant tensor, e.g. a Gibbs vector. Thus, the modeller must keep track of local reference frames and introduce manually-derived rotation transformations into the graph, as needed.

Note also that there are no components to explicitly represent the joints that connect the rigid bodies and constrain their absolute speeds. Thus, the 4 kinematic constraint equations (3) that express the interdependency of the 7 absolute coordinates must also be derived manually, with 2 equations obtained from each revolute joint.

The constraint forces that arise in the joints are converted into their corresponding Lagrange multipliers by the modulated transformers. Thus, these MTF elements

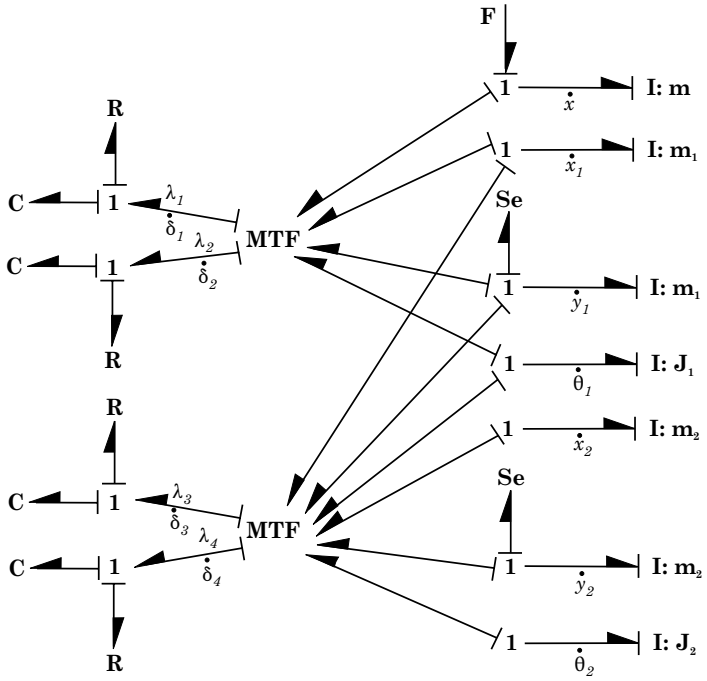


Fig. 8. Bond graph model of inverted double pendulum [5]

represent the rows of the Jacobian matrices (transposed) from equation (4); the top MTF in the bond graph corresponds to the Jacobian of the two kinematic constraints from the lower revolute joint in Figure 2, while the bottom MTF represents the Jacobian of the two constraint equations from the upper revolute joint.

Bond graphs were designed to operate on speeds, so that power is obtained from the product of efforts and flows. However, multibody system equations are generally expressed in terms of displacements, which are needed to calculate kinematic constraint equations, their Jacobians, and kinematic transformations between frames. These displacements are not represented as effort or flow variables in the bond graph, and must be obtained from numerical integration of the absolute speeds.

The use of absolute coordinates results in large systems of DAEs corresponding to equations (3) and (4). To avoid having to solve DAEs, resistance and capacitance elements are added to the bond graph to model damping and stiffness in the joints [33]. The translational speeds $\dot{\delta}$ across the joint, which are zero for an ideal joint model, are the complementary variables to the Lagrange multipliers. By modelling the joints in this fashion, the constraint equations are only needed to compute the Jacobians for the MTF elements, and the DAEs are converted to stiff ODEs. For the inverted double pendulum modelled with 7 absolute speeds, the 11 DAEs are reduced to 7 ODEs. Numerical integration of the latter may require special solvers for stiff systems, and will not guarantee that the 4 constraint equations are satisfied.

Tiernego and Bos [31] use multibond graphs to analyze the dynamics of open-loop multibody systems and, by applying well-known velocity transformations from absolute speeds to joint speeds [18], they are able to generate a minimal set of ODEs corresponding to equation (5). However, their procedure cannot handle the closed kinematic chains that typically arise in mechanical systems, and it seems that much of their analysis (e.g. derivation of velocity transformations) must also be performed manually before the bond graph is constructed.

Favre and Scavarda [15] present an extension of the work by Tiernego and Bos to systems with closed kinematic chains, but it seems that manual derivations of kinematic transformations are still required (no equations are presented for their two examples). The current state of bond graph modelling of multibody systems is best summarized by Karnopp et al [5]: "It is true that low-order, linear systems can be simulated with virtually no effort from the user. But complex nonlinear systems do require significant user input".

3.1.3 Bond Graph Model of Robot

Consider the bond graph representation, shown in Figure 9, of the two-link PD-controlled robot manipulator from Figure 4. The bond graph is quite complex, making it difficult to interpret, but it does constitute a single unified representation of this multidisciplinary application.

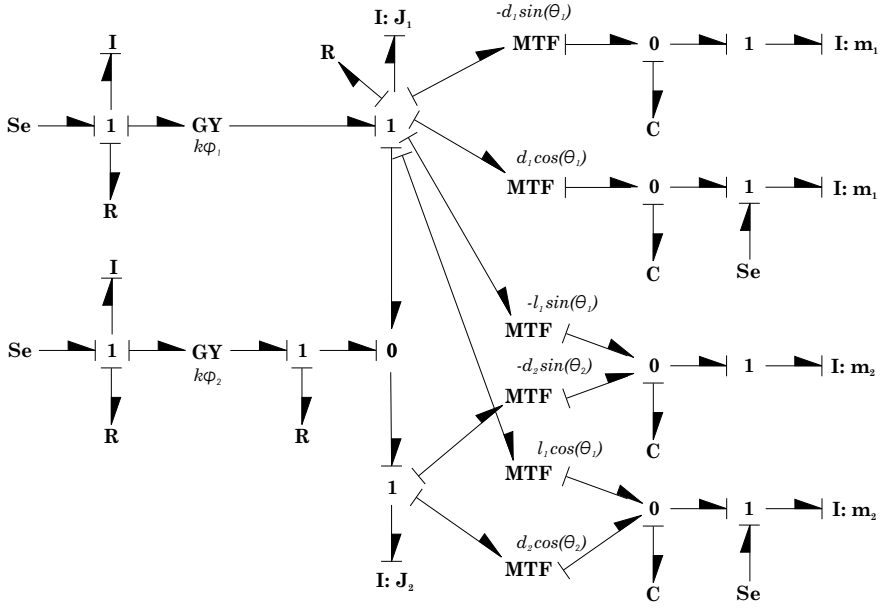


Fig. 9. Bond graph model of robot manipulator

The two DC-motors are modelled on the left of the bond graph. The applied voltage (coming from the PD-controller) is represented as an effort source, while the I and R elements correspond to the armature inductances and resistances, respectively. These elements are connected in series (1 junction) with a gyrator (GY), that couples the electrical domain to the mechanical domain. Substituting the component equations into the sum of efforts for the 1 junction gives the constitutive equation (6) for the DC-motor model.

The mechanical portion of this bond graph is somewhat different from that shown in Figure 8 for the inverted double pendulum. The mechanical subsystem is still represented by absolute speeds, with inertias corresponding to each of these 6 speeds. The joints are again modelled by stiffness (C) elements, but damping across the joint is not included. The modulated transformers are now used to convert between the Cartesian $\dot{x}, \dot{y}$ speeds and the absolute rotational speeds $\dot{\theta}_1$ and $\dot{\theta}_2$, i.e. velocity transformations. These transformations are manually derived and shown in the bond graph as parameters for the MTF two-port elements. The absolute rotational speeds are easily converted into joint angular speeds using 0 and 1 junctions. The rotational damping in the DC motor, seen in equation (7), is modelled by a resistance acting on the joint speed.

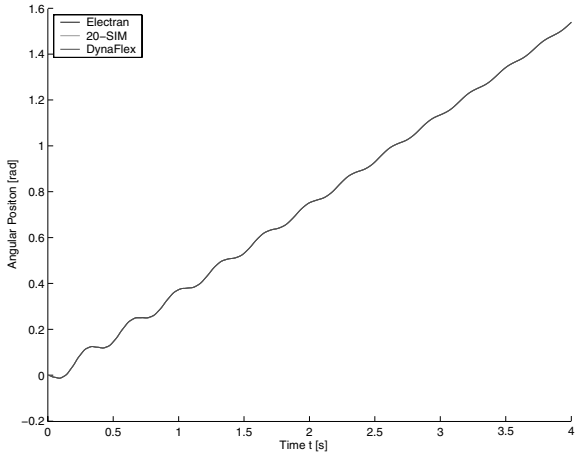


Fig. 10. Angular displacement of first joint versus time

To evaluate the performance of the PD-controller, the dynamic response of the robot is simulated by numerically integrating the 6 ODEs for the absolute coordinates. This was accomplished using the commercial software package 20-sim[2], with which the system equations can be generated from the bond graph model and numerically integrated. The 20-sim results for the angular displacement and speed of the first joint are shown in Figures 10 and 11, respectively. Also shown are the re-

[2] www.20sim.com

sults from two other software packages: DynaFlex, based on linear graph theory, and Robotran/Electran [13, 16], based on the virtual work principle and Kirchoff's laws.

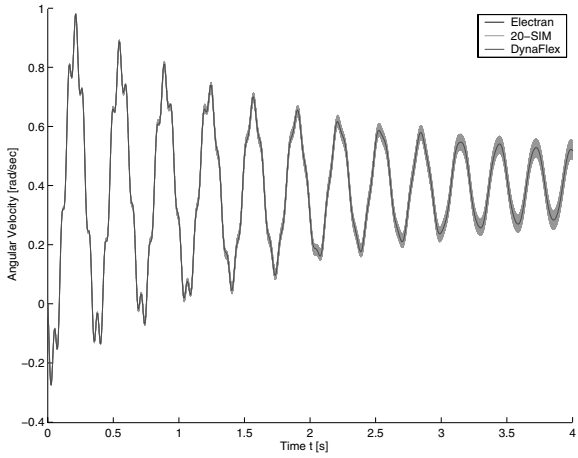


Fig. 11. Angular speed of first joint versus time

Because the robot is started from rest and immediately commanded to move at a constant speed of 90/4 deg/s (0.39 rad/s), large voltages are supplied by the PD controller to accelerate the system. This sets up an oscillation in the system response that is still evident at the end of the simulation. The plot of joint speed shows that the response is slowly converging to 0.39 rad/s, and that the bond graph results from 20-sim are in good agreement with those from the two other programs. However, additional high-frequency oscillations are visible in 20-sim velocity plot, resulting from the use of stiff springs to model the revolute joints. In contrast, the Electran and DynaFlex programs make direct use of rigid joint models, and their responses do not exhibit this high-frequency oscillation. As a result, they are able to simulate this problem faster than 20-sim.

3.1.4 Bond Graph Model of Gate System

Shown in Figure 12 is a bond graph model, created by Sass et al [27], of the 3-phase induction motor used to actuate the parking gate system in Figure 5. The inductive effects and the electromechanical coupling are modelled by a mixed IC-field for which the corresponding inductance matrices $\mathbf{L_s}$, $\mathbf{L_r}$, and $\mathbf{M}$ are defined by equations (9) and (11). The star-star connection shown in Figure 6 is modelled by the two 0 junctions that force the sum of the three currents to be zero. This star-star connection results in two constraints that can be seen in the bond graph as two derivative causalities for the I part of the IC-field.

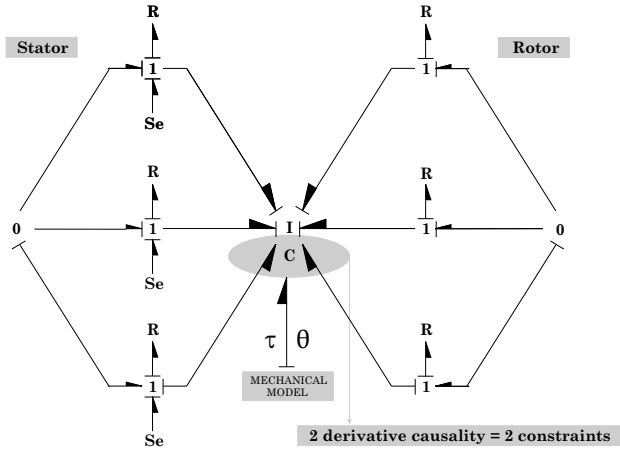


Fig. 12. Bond graph model of 3-phase actuator

A multibond graph of the 6-bar mechanism is shown in Figure 13. Using the approach of Favre and Scavarda [15] for this mechanism, every vector is assumed to be resolved in the inertial reference frame, and the center of mass is taken as the point of reference for each body. Thus, the bond graph takes the shape of a diamond for each body, as can be seen in the figure. This bond graph contains 5 moving bodies (for simplicity, the flexible beam is not shown in the graph) and the loop constraints are imposed by means of the zero-velocity flow source ($S_f = 0$). This method for opening and closing kinematic chains is only valid for revolute joints connected to the ground. Closing the loops elsewhere would require the manual calculation of the relative velocities of the two points connected by the revolute joint. Other joint types, e.g. prismatic or universal, would be even more problematic.

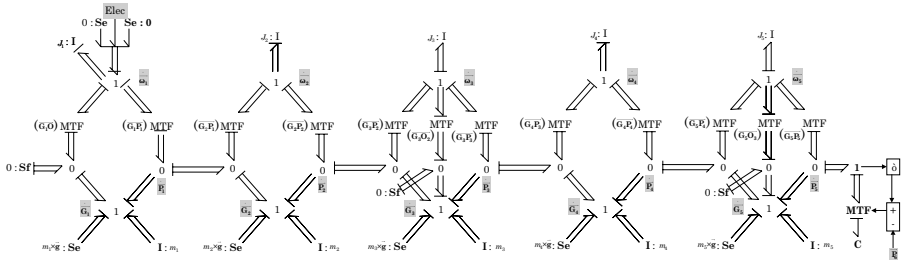


Fig. 13. Bond graph model of parking gate mechanism

One can see that the causality assignment and derivation of equations is not straightforward for a multidisciplinary system of this complexity. Furthermore, the manual derivation of kinematic transformations is very tedious for a system containing several bodies, especially if some of them are flexible. As a result of these

difficulties, a numerical simulation of the system response using bond graphs and the 20-sim software was abandoned.

3.2 Linear Graph Theory

Linear graph theory is a branch of mathematics devoted to the study of system topology. It has been combined with the characteristics of physical components to obtain a unified systems theory for modelling multidisciplinary applications. The term "graph-theoretic modelling" (GTM) is often used to denote this systems theory. In a nutshell, a system model is obtained by combining topological relationships from linear graph theory with the constitutive equations for individual components. This systems theory is very methodical and well-suited to computer implementation.

To model a physical system, individual components are identified and their constitutive equations are determined. In general, these constitutive relationships are obtained from experimental measurements of the component's "through" and "across" variables; through variables (τ) are measured by an instrument in series with the component, while across variables (α) are obtained from an instrument in parallel. For electrical systems, the through and across variables are current and voltage, respectively. For mechanical systems, force and displacement (or its derivatives) play the role of through and across variables, respectively. Note that through and across variables may be tensors of any order, including scalars and vectors.

One can see that the through and across variables correspond (approximately) to the flow and effort variables, respectively, from bond graph theory. However, the definition of through and across variables as experimental measurements naturally results in the force-flow analogy for mechanical systems. Furthermore, mechanical displacements are represented explicitly, and not as numerical integrals of velocities. Finally, the GTM approach is not restricted to scalar variables.

Once the constitutive equations are determined, the component models are combined in the topology defined by the structure of the physical system. A linear graph, consisting of lines (edges) and circles (nodes or vertices), is used to represent the system topology. The edges represent the individual components, whereas nodes represent the points of their interconnection. From this graph, linear topological equations are systematically obtained in terms of the through and across variables for all components. The system model is simply the combination of these topological equations with the individual constitutive equations.

Graph-theoretic modelling has been applied to a wide variety of disciplines and multidisciplinary applications, including electrical and mechanical systems [6, 10], electromechanical multibody systems [28], and electrohydraulic multibody systems [25]. The essential features of graph-theoretic modelling are presented in the following by means of the four example problems.

3.2.1 Linear Graph Model of Microphone

For the condenser microphone from Figure 1, a linear graph representation is shown in Figure 14. Edges R_1, C_2, L_3, and E_4 represent the resistor, capacitor, inductor,

and voltage source, respectively. Note that the linear graph resembles the physical system, which is an advantage when it comes to modelling using this approach. Directions are assigned to each edge to establish a positive convention for measuring the through and across variables, similar to setting the polarity on a measuring instrument. The constitutive equations for electrical components are expressed in terms of the scalar variables, current (i) and voltage (v). For the purpose of this example, we assume standard linear relationships for these components, e.g. $v_1 = R_1 i_1$ and $v_3 = L_3 \frac{di_3}{dt}$, but the constitutive equations may be highly nonlinear in general.

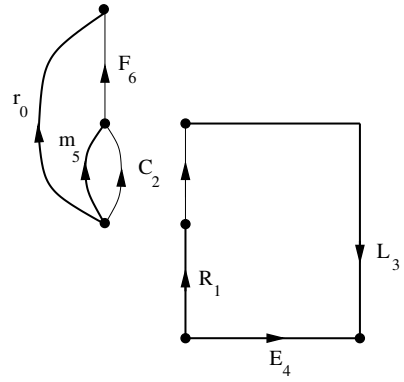


Fig. 14. Linear graph model of condenser microphone

Also shown in Figure 14 is the linear graph of the mechanical part of the condenser microphone. The edge m_5 represents the inertia and weight of the moving mass; the edge begins at a ground-fixed (inertial) reference node and terminates at the center of mass. Its constitutive equation is given by the combination of gravity with the d'Alembert form of Newton's Second Law: $\mathbf{F}_5 = m_5\mathbf{g} - m_5\boldsymbol{a}_5$, where the vector force $\mathbf{F}_5$ depends on gravity $\mathbf{g} = -g\hat{\boldsymbol{\imath}}$, the vector acceleration $\boldsymbol{a}_5 = \ddot{x}\,\hat{\boldsymbol{\imath}}$, and the upwards unit vector $\hat{\boldsymbol{\imath}}$ (parallel to x). The edge F_6 represents the combined effects of the spring and damper components (these could easily be split into separate edges for the spring and damper, if desired). Its vector constitutive equation is $\mathbf{F}_6 = -k_6(|\mathbf{r}_6| - l_6)\hat{\boldsymbol{r}}_6 - d_7(\mathbf{v}_6 \cdot \hat{\boldsymbol{r}}_6)\hat{\boldsymbol{r}}_6$, where l_6 is the undeformed spring length, k_6 and d_7 are the stiffness and damping coefficients, $\mathbf{v}_6$ is the relative velocity of the endpoints, and $\hat{\boldsymbol{r}}_6 = \mathbf{r}_6/|\mathbf{r}_6|$ is the unit vector parallel to the component. Finally, the edge r_0 locates the point where the spring-damper is attached to the ground: $\mathbf{r}_0 = l_0\,\hat{\boldsymbol{\imath}}$.

Note that the graph consists of two parts, one for each physical domain. This is always the case for linear graph models of multidisciplinary applications. The parts of the graph are not coupled explicitly, but by the constitutive equations for transducer components, which have an edge in each of the coupled domains. In this example, the electrical and mechanical domains are coupled by the moving-plate capacitor, which is characterized by two constitutive equations: the scalar equation

(2) for the electrical edge and the vector equation $\mathbf{F}_2 = -F_2\hat{\boldsymbol{\imath}}$ for the mechanical domain, where the attractive force F_2 is defined by equation (1) and acts in the $-X$ direction.

For each of the two parts of the linear graph, mechanical and electrical, we can generate sets of topological equations that relate the through and across variables. This can be done manually by inspection of the graph, or by applying matrix operations to an "incidence matrix" that encapsulates the topology of the physical system. For a linear graph with e edges and v vertices, entry I_{jk} of the $e \times v$ incidence matrix $\mathbf{I}$ is [0, -1, or +1] if edge k is [not incident upon, incident and away from, or incident and towards] the vertex j.

The Vertex Postulate [6] then allows us to write:

$$\mathbf{I}\,\boldsymbol{\tau} = 0 \tag{18}$$

where $\boldsymbol{\tau}$ is the column matrix of through variables for all edges. For electrical systems, the Vertex Postulate reduces to Kirchoff's Current Law at every node. For mechanical systems, the Vertex Postulate gives v equations for dynamic equilibrium.

Starting from the Vertex Postulate, two very useful sets of topological equations, the "cutset" and "circuit" equations, can be systematically derived by selecting a tree and applying elementary matrix operations to $\mathbf{I}$. For electrical networks, the circuit equations correspond to Kirchoff's Voltage Law around a closed circuit, while the cutset equations are linear combinations of the vertex equations for all the nodes in a given subgraph.

A tree is a set of $v - 1$ edges ("branches") that connects all of the vertices but does not contain any closed loops. A very attractive feature of linear graph theory is that *by selecting a tree, one can control the primary variables appearing in the final system*: they are the across variables $\boldsymbol{\alpha}_b$ for branch elements, and the through variables $\boldsymbol{\tau}_c$ for cotree elements ("chords"). This is accomplished by:

- re-writing the cutset equations as the chord transformations $\boldsymbol{\tau}_b = -\mathbf{A}_c\boldsymbol{\tau}_c$, where $\boldsymbol{\tau}_b$ are the branch through variables and $\mathbf{A}_c$ is obtained from elementary row operations on $\mathbf{I}$
- by re-writing the circuit equations as the branch transformations $\boldsymbol{\alpha}_c = -\mathbf{B}_b\boldsymbol{\alpha}_b$, where $\boldsymbol{\alpha}_c$ are the cotree across variables

The Principle of Orthogonality, which represents a very generalized energy conservation principle, guarantees that $\mathbf{B}_b = -\mathbf{A}_c^T$.

By selecting edges R_1, L_3, and E_4 into the tree for the electrical sub-graph in Figure 14, one gets the chord transformations:

$$\begin{Bmatrix} i_1 \\ i_3 \\ i_4 \end{Bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} i_2 \tag{19}$$

and the single branch transformation:

$$v_2 = -\begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}^T \begin{Bmatrix} v_1 \\ v_3 \\ v_4 \end{Bmatrix} \tag{20}$$

Assuming that there is one constitutive equation for each of the v elements, substituting the branch and chord transformations into these constitutive equations will result in v system equations in terms of the v primary variables.

It is possible to reduce the equations to an even smaller set by exploiting the linear nature of the electrical constitutive equations. One approach is to generate one equation for each capacitor and inductor, and to use the remaining constitutive equations and branch/chord transformations to express all other variables in terms of the capacitor voltages and inductor currents. This approach was successfully implemented by Muegge [7]. For the electrical portion of the linear graph shown in Figure 14, one would get two first-order ODEs in terms of v_2 and i_3.

Another approach is to express all variables in terms of the currents associated with chords, or the voltages associated with branches. The former is called the current formulation, while the latter is named the voltage formulation; both were implemented in the Maple symbolic programming language by Scherrer and McPhee [28]. By selecting the tree appropriately, one can significantly reduce the final number of system equations.

For the example shown in Figure 14 with the capacitor C_2 selected into the cotree, the current formulation will give a single second-order ODE in terms of the corresponding current i_2.

This is accomplished by substituting the chord transformations in (19) into the constitutive equations for the branches, giving:

$$\begin{aligned} v_1 &= R_1 i_2 \\ v_3 &= L_3 \frac{di_2}{dt} \\ v_4 &= E_4(t) \end{aligned}$$

where $E_4(t)$ is the prescribed voltage source. Substituting these constitutive equations into the branch transformation (20) gives:

$$v_2 = -R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \tag{21}$$

which expresses the capacitor voltage in terms of its current. Substituting this equation into the electrical constitutive equation (2), the single ODE for the electrical domain is obtained:

$$C_2' \dot{x} \left(-R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \right) + C_2 \frac{d}{dt} \left(-R_1 i_2 - L_3 \frac{di_2}{dt} + E_4(t) \right) = i_2 \tag{22}$$

where $C_2' \equiv dC_2/dx$ and the primary electrical variable is the cotree current i_2.

For the mechanical domain, the fixed vector r_0 and mass m_5 are selected into the tree shown as bold edges in Figure 14, resulting in only one primary across variable $r_5 = x$. This is an independent coordinate for the 1-dof mechanical subsystem, and a single dynamic equation is obtained from the cutset equation for the mass, projected onto its motion space defined by $\hat{\mathbf{i}}$:

$$(\mathbf{F}_5 + \mathbf{F}_2 - \mathbf{F}_6 = 0) \cdot \hat{\mathbf{i}} \tag{23}$$

Substituting the mechanical constitutive equations into this expression, evaluating, and re-arranging,

$$-m_5\ddot{x} - m_5 g + \frac{1}{2}\frac{dC_2}{dx}v_2^2 + d_7\dot{x}_6 + k_6(x_6 - l_6) = 0 \tag{24}$$

Assuming that the spring is unstretched at $x = 0$, which implies that $l_6 = r_0$, one gets the branch transformations:

$$\begin{aligned} x_2 &= x \\ x_6 &= r_0 - x, \quad \dot{x}_6 = -\dot{x} \end{aligned}$$

which shows that the spring-damper shortens as x increases. Substituting these branch transformations and the capacitor voltage (21) into equation (24), one gets the single ODE for the mechanical domain:

$$m_5\ddot{x} + d_7\dot{x} + k_6 x + m_5 g = \frac{1}{2}\frac{dC_2}{dx}\left(-R_1 i_2 - L_3\frac{di_2}{dt} + E_4(t)\right)^2 \tag{25}$$

Together, equations (22) and (25) can be solved for the primary variables $i_2(t)$ and $x(t)$. Equations (22) and (25) are equivalent to those derived by hand in [17].

3.2.2 Linear Graph Model of Inverted Double Pendulum

The same basic concepts apply when one models a multi-dimensional mechanical (multibody) system using linear graph theory: the system model is obtained by combining the constitutive equations for individual components with the linear cutset and circuit equations resulting from their connectivity. Again, the selection of a tree determines the primary variables appearing in the system equations. The cutset and circuit equations retain a simple form because linear graph theory allows the use of vector modelling variables. However, the constitutive equations for some components will be nonlinear due to the finite rotations of bodies in the system. Furthermore, the physical interpretation of nodes and edges must be generalized.

To illustrate, consider the inverted double pendulum shown in Figure 2, and its linear graph representation in Figure 15. For the sake of clarity, the three bodies are superimposed on the graph with dashed lines. One can see that the topology of the physical system is closely mirrored by the structure of the linear graph.

Each node in the linear graph represents the position and orientation of a body-fixed reference frame, while the edges represent transformations between frames corresponding to physical components. For each element, there are now two sets of through and across variables: translational and rotational. Thus, there will be two sets of cutset and circuit equations, since these variables cannot be added together. Although the incidence matrix is the same for each, selecting different trees can be used to create different cutset and circuit equations for translation and rotation. This can be used to reduce the system equations to a set that is smaller in number than those generated by conventional multibody formalisms [23].

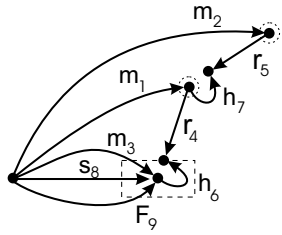


Fig. 15. Linear graph model of inverted double pendulum

In Figure 15, the edges m_1, m_2, and m_3 represent both the translational and rotational inertia of the three rigid bodies (slider and two links). These bodies are connected by the revolute joints h_6 and h_7, and by the prismatic joint s_8 between the slider and the ground. The "rigid arm" elements r_4 and r_5 define the position and orientation, relative to the center of mass frames on the bodies, of the body-fixed frames that define the connection points of these joints. Finally, the external force on the slider is modelled by the force element F_9, originating at the inertial frame (node) and terminating at the slider.

The constitutive equations for the multi-dimensional translation of rigid bodies (and spring-dampers) are the same as that shown in the previous section. However, a second equation relates the d'Alembert torque on the body to its rotational inertia. This equation corresponds to Euler's equations for rotational motion.

For the rigid-arm elements, e.g. r_4, the tip node does not rotate relative to the tail (center of mass) node; hence, the angular velocity (e.g. $\boldsymbol{\omega}_4$) is zero. However, the translational velocity of the rigid-arm is a nonlinear function of the angular velocity of the body on which it resides, e.g. $\mathbf{v}_4 = \boldsymbol{\omega}_1 \times \mathbf{r}_4$, which is a well-known result from rigid body kinematics. For the ideal joints, one always finds that the motion allowed by a joint, e.g. $\mathbf{r}_8 = x\hat{\boldsymbol{\imath}}$ where $\mathbf{r}_8$ is the translational displacement of the slider along X, is orthogonal to the reaction forces and torques that arise in the joint, e.g. $\mathbf{F}_8 = F_8\,\hat{\boldsymbol{\jmath}}$ and $\mathbf{T}_8 = T_8\hat{\boldsymbol{k}}$ where $\hat{\boldsymbol{\jmath}}$ and $\hat{\boldsymbol{k}}$ are unit vectors parallel to Y and Z, respectively. This is a result of the fact that ideal joints do no work, and can be used to eliminate joint reactions in the system dynamic equations.

The topological equations remain linear regardless of the nonlinearities in the constitutive equations. Furthermore, the selection of trees can again be used to define the primary variables $\mathbf{q}$ and $\boldsymbol{\lambda}$ in the final system equations, where the "branch coordinates" $\mathbf{q}$ are the unknown across variables for elements (branches) in the tree.

By selecting into the tree those components with known across variables, i.e. r_4 and r_5, the number n of branch coordinates (and system equations) is reduced. If the tree is completed by m_1, m_2, and m_3, then the final equations are in terms of the absolute coordinates for the three bodies. In that case, the system equations take the form of the DAEs (3) and (4).

If joints h_6, h_7, and s_8 are selected into the tree in place of m_1, m_2, and m_3, then one obtains equations in the joint coordinates β_6, β_7, and x; for this open-loop system, the governing equations would then take the form of the ODEs shown in

(5). Thus, linear graph theory provides a unification of traditional absolute and joint coordinate formulations.

Any joints left in the cotree, e.g. h_6 in the absolute coordinate formulation, will provide the reaction loads appearing in the Lagrange multipliers $\boldsymbol{\lambda}$. Furthermore, these cotree joints will also provide one kinematic constraint equation for each reaction load. These m constraint equations (3) express the relationships between the branch coordinates, which will not be independent if there are joints in the cotree; this is always the case for a system with closed kinematic chains. These constraint equations are always found by projecting the circuit equations for the cotree joints onto their reaction spaces.

To demonstrate, consider a mixed-coordinate formulation that results from selecting m_1, m_2, r_4, r_5, and s_8 into the tree for Figure 15. The corresponding branch coordinates $\mathbf{q} = [x_1, y_1, \theta_1, x_2, y_2, \theta_2, x]^T$ are identical to those used in the previous bond graph model of this system. (Note that the joint speed $\dot{x}$ is the same as the absolute speed used in the bond graph model, which neglected the other two absolute speeds for the sliding mass in an ad hoc manner). The two revolute joints left in the cotree, h_6 and h_7, each provide 2 Lagrange multipliers and 2 constraint equations to the system DAEs, for a total of $m = 4$ constraints. For example, one can generate the translational circuit equation for joint h_6:

$$\mathbf{r}_1 + \mathbf{r}_4 - \mathbf{r}_6 - \mathbf{r}_8 = 0 \tag{26}$$

where $\mathbf{r}_6 = 0$ and $\mathbf{r}_8 = x\hat{\boldsymbol{\imath}}$ from the constitutive equations for revolute and prismatic joints, respectively. For the rigid body, $\mathbf{r}_1 = x_1\hat{\boldsymbol{\imath}} + y_1\hat{\boldsymbol{\jmath}}$, and $\mathbf{r}_4 = -l_1 \sin\theta_1 \hat{\boldsymbol{\imath}} - l_1 \cos\theta_1 \hat{\boldsymbol{\jmath}}$ for the body-fixed vector.

Note that the circuit equations represent the zero summation of displacement vectors around a closed kinematic chain. The reaction space for h_6 is spanned by unit vectors $\hat{\boldsymbol{\imath}}$ and $\hat{\boldsymbol{\jmath}}$, since the revolute joint prevents translations along these two axes (but allows rotation along $\hat{\boldsymbol{k}}$, which defines the joint motion space). Projecting the vector circuit equation (26) onto these two unit vectors, and substituting all constitutive equations, results in the kinematic constraint equations:

$$x_1 - l_1 \sin\theta_1 - x = 0 \tag{27}$$

$$y_1 - l_1 \cos\theta_1 = 0 \tag{28}$$

which are easily verified by hand. These equations, which are very systematically generated in terms of $\mathbf{q}$, correspond to two of the four rows in the constraint equations (3) for this example. The other two equations are obtained in exactly the same manner from the cotree joint h_7.

To obtain the dynamic equations of the system, the cutset equations for each branch are projected onto the motion space for that branch. This is why, in the previous example, the cutset equation for the capacitor mass m_5 was projected onto the unit vector $\hat{\boldsymbol{\imath}}$ defining its motion space.

Note that one can also generate the dynamic equations by combining linear graph theory with analytical mechanics, e.g. the principle of virtual work. This approach is very useful for incorporating flexible bodies into the multibody system model

[30]. It has been implemented using the Maple symbolic programming language into a multibody dynamics program called DynaFlex[3], which can reduce the DAEs to ODEs by means of symbolic coordinate partitioning. DynaFlex can be also be used to model electromechanical multibody systems, as shown in the next section.

3.2.3 Linear Graph Model of Robot

For the PD-controlled robot manipulator in Figure 4, the linear graph representation is shown in Figure 16.

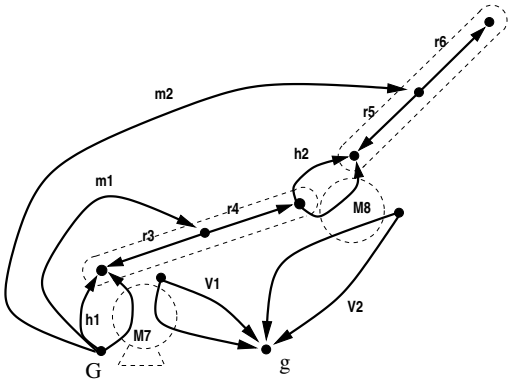


Fig. 16. Linear graph model of robot manipulator

The mechanical subgraph is obtained by applying the systematic procedure described in the previous section:

- one node is added for each body-fixed reference frame, e.g. a center of mass frame, joint connection point, or force application point;
- these nodes are connected by edges corresponding to different physical components (rigid body m, revolute joint h, rigid arm r).

In a similar manner, the electrical subgraph is directly obtained by drawing an edge for each physical component in the electrical circuit, on a one-to-one basis (e.g. voltage source V). As described previously, the two physical domains are coupled by the DC-motor transducer elements (M), which have an edge associated with each domain. The constitutive equations for the electrical and mechanical edges are defined by equations (6) and (7), respectively.

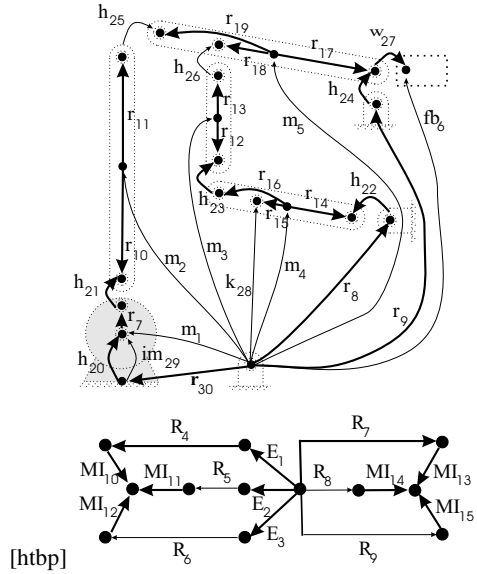
Note that many of the contributing terms in these transducer constitutive equations could be modelled by separate elements in the electrical and mechanical subgraphs, as was done in the bond graph model; here, they are combined into this "subsystem" representation of the DC-motor, for modelling convenience [29].

[3] http://real.uwaterloo.ca/∼dynaflex/

To generate a minimal set of system equations, all rigid arms and revolute joints were selected into the tree of the mechanical subgraph. For the electrical subsystem, the voltage sources were selected into the tree and a current formulation was applied. The system equations were automatically generated in symbolic form by DynaFlex; these ODEs are explicit functions of the joint angles θ_1 and θ_{1-2} and the two motor currents i_7 and i_8. These 4 ODEs were exported to Matlab and solved using a standard numerical integrator. As shown in Figures 10 and 11, the DynaFlex results are in exact agreement with those from the independent software package Electran.

3.2.4 Linear Graph Model of Gate System

Figure 17 depicts the linear graph representation of the parking gate system shown in Figures 5 and 6. The mechanical subgraph consists of components discussed in previous sections, plus three new components: a weld joint (w), a flexible body (fb), and an induction motor (im). The weld joint, as its name suggests, simply locks two reference frames together.



[htbp]

Fig. 17. Linear graph model of parking gate

The flexible body element represents the flexible parking barrier. Details regarding the constitutive equations for the flexible body can be found in [30]. To summarize, a Rayleigh beam model is used in conjunction with polynomial shape functions that represent the axial and torsional deformations, as well as bending about two lateral axes.

For this planar mechanism, only the in-plane bending was considered; 5 elastic variables were used to model this bending deflection. The value of 5 was obtained by

progressively adding more deformation variables until the simulation results converged. It was also found that including axial deformations in the flexible beam model had no effect on the simulation results.

The mechanical domain is coupled to the electrical domain by the induction motor. The constitutive equation for the mechanical edge (im_{29}) of this transducer is defined by equation (8).

The graph for the electrical domain is almost identical to the circuit shown in Figure 6, the only difference being that the inductors have been replaced by mutual inductance components. Note that the induction motor transducer has multiple edges in the electrical subgraph. The constitutive equation for each one of these *MI* edges is defined by equation (10).

By selecting an appropriate tree (shown in bold in Figure 17) and using a current formulation, DynaFlex was used to generate 14 ordinary differential equations in symbolic form: 5 for the rigid multibody system in terms of joint coordinates $\beta_{20} - \beta_{24}$, 5 for the deformation variables of the beam, and 4 for the electrical system in terms of the cotree resistor currents i_5, i_6, i_8 and i_9. These ODEs were exported to Matlab and solved by a standard numerical integrator.

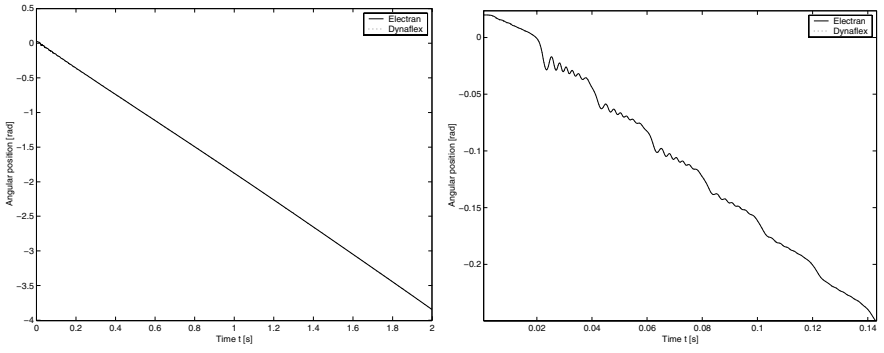


Fig. 18. Angular displacement of input link

Shown in Figures 18 and 19 are the numerical results from DynaFlex and Electran for the angular displacement of the input link and the current through a rotor inductor, respectively. The plots on the right of these figures are over a shorter time scale, in order to show the slight differences between the DynaFlex and Electran results that are not visible in the plots on the left. These results have been obtained using a 1:53 gear ratio between the motor and the 6-bar mechanism.

The results from the two software packages are nearly identical, with only slight differences in the currents. There are two possible sources of this difference:

- DynaFlex uses a Rayleigh beam model, while the Electran results were generated with a flexible beam model that consisted of several rigid bodies connected by rotational springs.

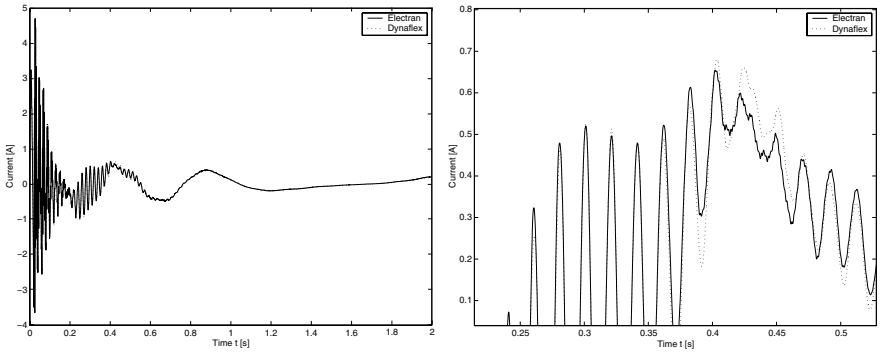


Fig. 19. Current through rotor inductor L_{r2} (MI_{15})

- the Electran results were generated with a different set of joint coordinates than those used by DynaFlex.

Note that the input link rotor oscillations shown in Figure 18 are damped. The only dissipative elements in this model are the electrical resistance; changing the resistance value has an effect on the damping of these mechanical oscillations. This illustrates the tight interaction between the electrical and mechanical subsystems in this multidisciplinary application.

4 Conclusions

The two most prominent unified theories for modelling multidisciplinary systems — linear graph theory and bond graph theory — have been examined in detail. Although similar in their decomposition of a system into a collection of discrete component models that are combined using topological equations, they are quite different in their graphical representation, the variables that they use, and the range of problems for which they are suited.

A bond graph model provides a single, unified representation of a multidisciplinary application, in which the coupling between physical domains is represented explicitly by a transducer element in the graph. Associated with each element are effort and flow variables, the product of which is the power flowing through the graph. Bond graphs are very powerful for modelling systems governed by scalar variables, e.g. electrical networks and 1-dimensional mechanical systems, and a number of software packages are available for simulating bond graph models. However, they were not designed to use tensor variables or mechanical displacements, and are therefore not very well-suited to the modelling of planar or spatial mechanical systems. Furthermore, the popular use of the force-effort analogy leads to a discrepancy between the multibody system topology and the junctions used to model this topology, the use of absolute coordinates in bond graph models leads to relatively large systems of DAEs, and systems with several bodies require many tedious manual calculations

of kinematic transformations. Further research is needed to address these issues in bond graph modelling of multibody systems.

A linear graph model also provides a single, unified representation of a multidisciplinary application. However, the coupling between domains is not as evident in the graph, which consists of separate parts for each domain; this coupling is embedded in the constitutive equations for transducer elements, which have an edge in each of the two connected domains. Graph-theoretic modelling is supported by very few software products, but it is very well-suited to the modelling of multi-dimensional multidisciplinary applications. This can be attributed to the use of mechanical displacements as across variables, the ability to use tensors as through and across variables, and the existence of models for a variety of mechanical joints and rigid or flexible bodies. A unique and powerful feature of a graph-theoretic model is that, by selecting a tree, one can control the variables that appear in the final system equations. For multibody systems, these variables can include absolute or joint coordinates, or some combination of the two.

5 Acknowledgements

Special thanks goes to Laurent Sass and Chad Schmitke for their work on modelling and simulating the examples, and for many fruitful discussions. The author's research is funded by a Premier's Research Excellence Award (Province of Ontario), Waterloo Maple Inc., and the Natural Sciences and Engineering Research Council of Canada.

References

1. Biggs N, Lloyd E, Wilson R (1976) Graph Theory: 1736-1936. Oxford University Press, Oxford
2. Breedveld P, Dauphin-Tanguy G (1992) Bond Graphs for Engineers. Elsevier Science, Amsterdam
3. Crandall S, Karnopp D, Kurtz E, Pridmore-Brown D (1968) Dynamics of Mechanical and Electromechanical Systems. McGraw-Hill, New York
4. Gayford M (1971) Electroacoustics – Microphones, Earphones and Loudspeakers. Elsevier Publishing Company, New York
5. Karnopp D, Margolis D, Rosenberg R (2000) System Dynamics: Modeling and Simulation of Mechatronic Systems. John Wiley and Sons, New York
6. Koenig H, Tokad Y, Kesavan H (1967) Analysis of Discrete Physical Systems. McGraw-Hill, New York
7. Muegge B (1996) Graph-Theoretic Modelling and Simulation of Planar Mechatronic Systems. University of Waterloo, Canada
8. Norton R (2001) Design of Machinery, 2nd ed. McGraw-Hill, New York
9. Paynter H (1961) Analysis and Design of Engineering Systems. MIT Press, Cambridge, Massachusetts
10. Roe P (1966) Networks and Systems. Addison-Wesley, Reading, Massachusetts
11. Rowell D, Wormley D (1997) System Dynamics: An Introduction. Prentice Hall, New Jersey

12. Thoma J (1975) Introduction to Bond Graphs and their Applications. Pergamon Press, Oxford
13. Sass L, Fisette P, Grenier D (2002) Modelling of mechatronic systems — study of the actuation of a swinging barrier. In: Proceedings of The 8th Mechatronics Forum International Conference, University of Twente, Enschede, Netherlands
14. Birkett S, Roe P (1990) The Mathematical Foundations of Bond Graphs - III. Matroid Theory. J Frank Inst 327(1):87–108
15. Favre W, Scavarda S (1998) Bond Graph Representation of Multibody Systems with Kinematic Loops. J Frank Inst 335B(4):643–660
16. Fisette P, Postiau T, Sass L, Samin J-C (2002) Fully Symbolic Generation of Complex Multibody Models. Mech Struct Mach 30(1):31–82
17. Hadwich V, Pfeiffer F (1995) Principle of Virtual Work in Mechanical and Electromechanical Systems. Arch Appl Mech 65:390–400
18. Jerkovsky W (1978) The structure of multibody dynamics equations. J Guid Control 1(3):173–182
19. Karnopp D (1997) Understanding Multibody Dynamics Using Bond Graph Representations. J Frank Inst 334B(4):631–642
20. Kübler R, Schiehlen W (2000) Modular Simulation in Multibody System Dynamics. Multibody Sys Dyn 4:107–127
21. Lovekin D, Heppler G, McPhee J (2000) Design and Analysis of a Facility for Free-floating Flexible Manipulators. Trans CSME 24(2):375–390
22. Maisser P, Enge O, Freudenberg H, Kielau G (1997) Electromechanical Interactions in Multibody Systems Containing Electromechanical Drives. Multibody Sys Dyn 1:281–302
23. McPhee J (1998) Automatic Generation of Motion Equations for Planar Mechanical Systems Using the New Set of "Branch Co-ordinates". Mech Mach Theory 33(6):805–823
24. McPhee J (1996) On the use of linear graph theory in multibody system dynamics. Nonlin Dyn 9:73–90
25. Papadopoulos E, Gonthier Y (2002) On the Development of a Real-Time Simulator Engine for a Hydraulic Forestry Machine. Int Journal Fluid Power 3(1):55–65
26. Perelson A, Oster G (1976) Bond Graphs and Linear Graphs. J Frank Inst 302(2):159–185
27. Sass L, McPhee J, Schimitke C, Fisette P, Grenier D (2004) A Comparison of Different Methods for Modelling Electromechanical Multibody Systems. Multibody Sys Dyn 12(3):209–250
28. Scherrer M, McPhee J (2003) Dynamic Modelling of Electromechanical Multibody Systems. Multibody Sys Dyn 9:87-115
29. Schmitke C, McPhee J (2003) A Procedure for Modeling Multibody Systems Using Subsystem Models. Int J Multiscale Comp Eng 1(2):139-159
30. Shi P, McPhee J (2000) Dynamics of Flexible Multibody Systems Using Virtual Work and Linear Graph Theory. Multibody Sys Dyn 4:355–381
31. Tiernego M, Bos A (1985) Modelling the Dynamics and Kinematics of Mechanical Systems With Multibond Graphs. J Frank Inst 319(1/2):37–50
32. Trent H (1955) Isomorphisms between linear graphs and lumped physical systems. J Acoust Soc Am 27:500–527
33. Zeid A, Chung C-H (1992) Bond Graph Modeling of Multibody Systems: A Library of Three-Dimensional Joints. J Frank Inst 329(4):605–636