

AUTOMATED MODELING AND SIMULATION USING THE BOND GRAPH METHOD FOR THE AEROSPACE INDUSTRY

Jose J. Granda, Professor
Department of Mechanical Engineering
California State University, Sacramento
Sacramento, California 95819 USA
grandajj@ecs.csus.edu

Raymond C. Montgomery, AST
Dynamics and Control Branch
NASA Langley Research Center
Hampton, Virginia 23681 USA
r.c.montgomery@larc.nasa.gov

ABSTRACT

Bond graph modeling was originally developed in the late 1950s by the late Prof. Henry M. Paynter of M.I.T. Prof. Paynter acted well before his time as the main advantage of his creation, other than the modeling insight that it provides and the ability of effectively dealing with Mechatronics, came into fruition only with the recent advent of modern computer technology and the tools derived as a result of it, including symbolic manipulation, MATLAB, and SIMULINK and the Computer Aided Modeling Program (CAMP-G). Thus, only recently have these tools been available allowing one to fully utilize the advantages that the bond graph method has to offer. The purpose of this paper is to help fill the knowledge void concerning its use of bond graphs in the aerospace industry. The paper first presents simple examples to serve as a tutorial on bond graphs for those not familiar with the technique. The reader is given the basic understanding needed to appreciate the applications that follow. After that, several aerospace applications are developed such as modeling of an arresting system for aircraft carrier landings, suspension models used for landing gears and multibody dynamics. The paper presents also an update on NASA's progress in modeling the International Space Station (ISS) using bond graph techniques, and an advanced actuation system utilizing shape memory alloys. The later covers the Mechatronics advantages of the bond graph method, applications that simultaneously involves mechanical, hydraulic, thermal, and electrical subsystem modeling.

INTRODUCTION

In response to this technological gap, bond graph theory has recently made its way into the Engineering curricula of Mechanical, Electrical, Computer Science departments of many major universities, and complex industrial applications have been recently developed. These applications have been primarily confined to the nuclear, electrical power, and automotive industries. And although elementary examples relative to the aerospace industry exist, the bond graph method remains mostly unknown in aerospace industry circles. Recently however, NASA has initiated a study to use

bond graphs in modeling the International Space Station (ISS), reference 1.

In designing complex dynamic systems, the aircraft industry has adopted classical modeling techniques based on representation of dynamic system components relations in the form of equations or block diagrams. With the proliferation of computers in the late 1970's and 1980's, it was possible to model and simulate complex nonlinear systems. Transforming equations or block diagrams into high level computer code allowed computer simulations to obtain time dependent system response and simulations in the frequency domain. During the 70's when technological advances used in the US Space program were transferred to industry in the public sector, the industry adopted programs like CSMP (Continuous System Modeling Program), ACSL (Advanced Continuous Simulation Language) (reference 2), CSSL Continuous Systems Simulation Language, DSL (Digital Simulation Language). All these programs allowed engineers to start with basic differential and algebraic equations or with their representation in block diagrams. These were transformed in lines of code representing individual constitutive relations. Most of them allowed these relations to be described in random order saving tremendous computational and programming time. These programs translated the user description of the system and internally produced executable code that could sequentially be computed. The later step was transparent to the user.

In the late 1980's and the beginning of the 1990's, the use of microcomputers gave raise to new possibilities and new software. The industry has widely adopted the use of MATLAB and SIMULINK. MATLAB is a programming language in contrast to the simulation languages listed above. This means that in order to produce simulation results, the input needs to be a logically organized computer code. Therefore in order to produce a simulation in MATLAB, a logical sequential program needs to be developed. Of course, MATLAB offers very nice tools that simplify this task. SIMULINK has saved programming time also since the ability to enter block diagrams in graphical form allows a more direct interface in tune with the engineer's thinking and the theory behind classical

methods. This approach has been very appealing to design and controls system engineers. If all is well then what is the purpose of this new so-called *bond graph* technique and why should it be better than existing techniques?

The answer to this question is very simple. In order to produce a model in any simulation program like ACSL or in a programming language such as MATLAB and SIMULINK, the user has to have the differential equations, which in turn need, to be converted into computer code or interconnected block diagrams that represent such mathematical expressions. The Bond Graph technique offers a method, which by transforming the real system into a topological bond graph, that describes the type of elements and their connections, can be entered directly in graphical form into a software package like CAMP-G (Computer Aided Modeling Program with Graphical Input). Then, the computer directly generates not only the differential equations but also the actual MATLAB/SIMULINK code in the form of .m, mdl and .s files. This approach produces equations in first order form, thus using the state space format. This representation bridges the Bond Graph method and classical methods since the description in a MATLAB m file follows the same syntax regardless of the method used. Therefore, once a bond graph is entered into the computer, the production of the state space form with all its system matrices is computer generated. In this context the bond graph approach has sidestepped the equation derivation process, a major and key step in the whole process. The automation of the modeling process is a worthwhile effort to make the simulation with MATLAB and SIMULINK even simpler. Using this approach, one can also obtain transfer functions in symbolic and numerical form thus allowing many of the MATLAB toolboxes to be used. Moreover, the method presents the unique feature of being able to model systems in different energy domains using the same approach with a single model, thus it becomes ideal for modeling and simulation of Mechatronics and control systems.

BOND GRAPH NOTATION

Shown in figure 1 is the symbol of a bond graph.

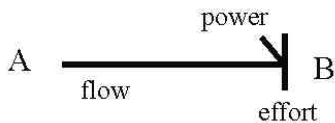


Figure 1. Bond graph symbol

This symbol represents the dynamics between two components of a dynamic system. It has three components. There is the line that joins element A to element B. There is a half arrow that indicates that power flows either from A to B or from B to A and

finally the third component is the causal stroke, a perpendicular small line either at end A or at end B. The latter indicates which power variable is the input and which is the output to A or to B. A summary of the fundamentals follows but more details can be found in reference 3.

1. Equivalence between bond graphs and block diagrams

The bond graph symbol described above contains intrinsically a description on which way power flows and what are the equations between A and B. The half arrow indicates the power flow and the causal stroke indicates which variable is the input and which is the output. Depending on the type of element will generate differential equations or static algebraic equations. Figure 2 expands and explains these concepts and their relations to block diagrams.

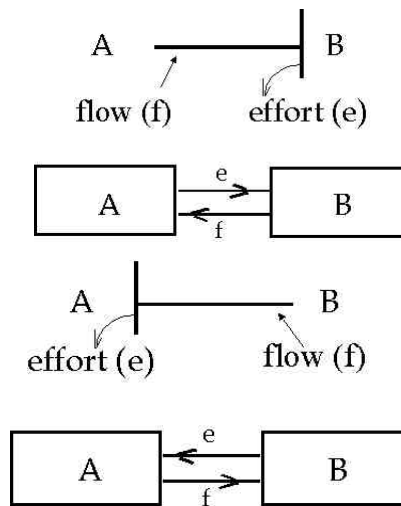


Figure 2. Bond Graph and equivalent Block Diagram

The relationship between the input and the output in an individual element is called the "constitutive relation". This relation classifies the type of elements into energy storing elements such as inertias and capacitances, energy dissipating elements such as dampers, resistors, or elements that transform energy such as transformers and gyrators. The basic elements are I (Inertia), C (Capacitance), R (Resistance), TF (Transformer), GY (Gyrator), SE (Source of effort, SF (Source of flow), 1 (Common flow junction), 0 (Common effort junction). With these nine elements systems in the electrical, mechanical, rotational, hydraulic and thermal systems or a combination of these can be built.

2. Power and State Variables

Bond graph models operate with generalized variables, two power variables (effort) and (flow), also known in other methods as through and across variables and two

state variables p (generalized momentum) and q (generalized displacement). Figure 3 shows a chart with the equivalent of variable descriptions in different energy domains.

Power Variables

	Effort (e)	Flow (f)
Mechanical Translation	Force (F) Newtons N	Velocity (v) m/s
Mechanical Rotation	Torque (T) N-m	Angular velocity (w) rad/s
Electrical	Voltage Volts V	Current (i) Amperes A
Hydraulic	Pressure (P) N/m ²	Volume flow (q) m ³ /s

State Variables

	Generalized Displacement	Generalized Momentum
Mechanical Translation	Displacement (x) m	Momentum N-s
Mechanical Rotation	Angular displacement Rad	Angular Momentum N-m-s
Electrical	Charge (q) A-s	Flux linkage V-s
Hydraulic	Volume m ³	Pressure Momentum N-s/ m ²

Figure 3. Power Variables and State Variables

Using the variable definitions, the elements mentioned above and the junctions, dynamic systems can be assembled. Let's take on first a suspension system such in a landing gear.

3. Differential Equations, Bond Graphs and Classical Block Diagrams

In order to illustrate the equivalency before more complex systems are analyzed, it is constructive to apply basic principles to establish the differential equations and then apply the bond graph method to provide a better understanding for those who are not familiar with this method.

Using the system shown in figure 4 is easy to recognize that it is a fourth order system. Considering it has two degrees of freedom, two-second order differential equations can be obtained by simply applying Newton's law. Once the system is represented in state variable form (first order form), then four first order differential equations, one for each state variable are generated.

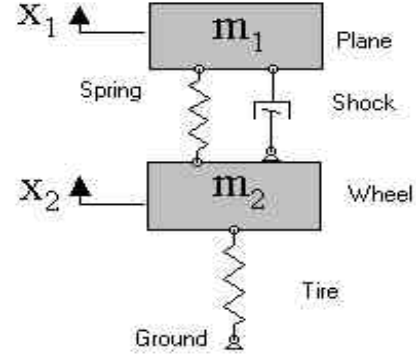


Figure 4. Simple suspension fourth order system

Using a free body diagram and applying Newton's law summation of forces in the vertical direction yields.

$$\sum F = \frac{d(mv)}{dt} \quad (1)$$

$$b_1(\dot{x}_2 - \dot{x}_1) + k_1(x_2 - x_1) - m_1g = m_1\ddot{x}_1 \quad (2)$$

$$-b_1(\dot{x}_2 - \dot{x}_1) - k_1(x_2 - x_1) + k_2x_2 - m_2g = m_2\ddot{x}_2 \quad (3)$$

These second order differential equations can be solved using a simulation program. The software would perform integration of the second order equations. The equations however can be transformed into the state space form from the first order differential equations.

Changing variables and using the following transformations yield the following equations.

$$y_1 = x_1 \quad \text{and} \quad \frac{dy_1}{dt} = y_2 \quad (4)$$

$$y_3 = x_2 \quad \text{and} \quad \frac{dy_3}{dt} = y_4 \quad (5)$$

$$\frac{dy_2}{dt} = \frac{d^2y_1}{dt^2} = \frac{d^2x_1}{dt^2} = \ddot{x}_1 \quad (6)$$

Using equations (2) then yields

$$\frac{dy_2}{dt} = \frac{b_1}{m_1}(\dot{x}_2 - \dot{x}_1) + \frac{k_1}{m_1}(x_2 - x_1) - \frac{m_1}{m_1}g \quad (7)$$

Therefore:

$$\frac{dy_2}{dt} = \frac{b_1}{m_1}(y_4 - y_2) + \frac{k_1}{m_1}(y_3 - y_1) - \frac{m_1}{m_1}g \quad (8)$$

Now the fourth equation is obtained

$$\frac{dy_4}{dt} = \frac{d^2y_3}{dt^2} = \frac{d^2x_2}{dt^2} = \ddot{x}_2 \quad (9)$$

and using equation (3)

$$\frac{dy_4}{dt} = -\frac{b_1}{m_2}(\dot{x}_2 - \dot{x}_1) - \frac{k_1}{m_2}(x_2 - x_1) + \frac{k_2}{m_2}x_2 - \frac{m_2}{m_2}g \quad (10)$$

Therefore:

$$\frac{dy_4}{dt} = -\frac{b_1}{m_2}(y_4 - y_2) - \frac{k_1}{m_2}(y_3 - y_1) + \frac{k_2}{m_2}y_3 - \frac{m_2}{m_2}g \quad (11)$$

It is clear that the equations 4, 5, 8 and 11 form a set of state space equations in first order form. These set of equations derived by a conventional method such as applying Newton's equations can be solved also using conventional solutions using MATLAB and its tools tailored to first order differential equations. These equations can also be arranged in matrix form.

4. Differential equations from the Bond Graph Model.

Now, based on the principles outlined above with the bond graph method, the system is described in a topological way outlining the element types and the way they are connected. Figure 5 shows the landing gear suspension with its corresponding bond graph in the schematic form and the bond graph form. The mass spring, damper system and its equivalent bond graph are outlined in the dotted boxes. The I junctions indicate the velocities, the SE elements represent the force of gravity, the I (mass), R (damper), C (spring).

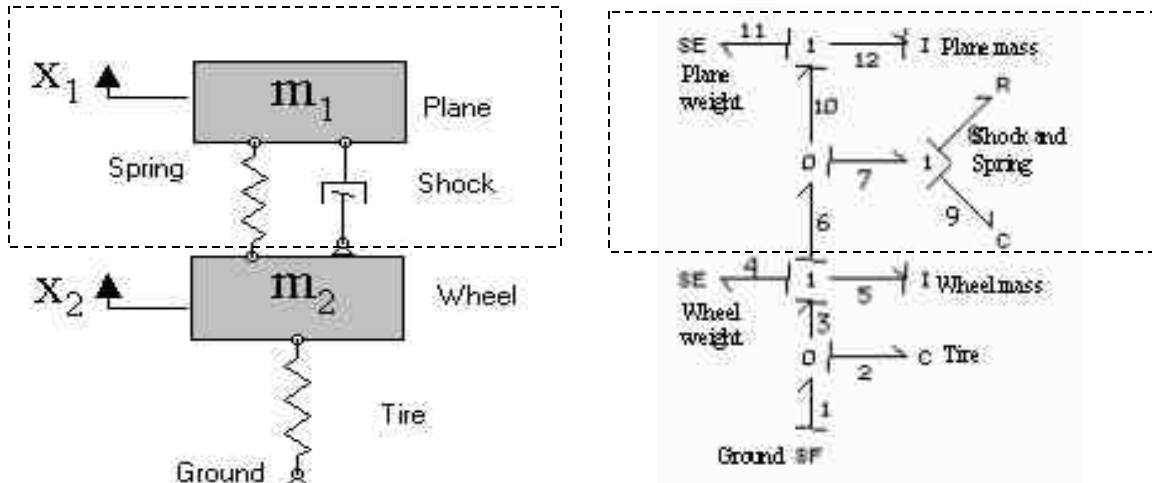


Figure 5. Physical system and corresponding Bond Graph

Differential and algebraic equations are derived from the bond graph, references 4 and 5.

Now the bond graph is entered into CAMP-G and the differential equations are computer generated. These computer-generated equations using the power variables and state variables notation are.

$$dQ2=SF1-P5/I5 \quad (12)$$

$$dQ9=P5/I5-P12/I12 \quad (13)$$

$$dP12=P5/I5*R8-P12/I12*R8+Q9/C9-SE11 \quad (14)$$

$$dP5=Q2/C2-SE4-P5/I5*R8+P12/I12*R8-Q9/C9 \quad (15)$$

It is as simple as entering the bond graph in graphical form as an input to obtain the equations above. So, the next step is to prove these equations are in fact equivalent to those state space equations obtained from the set of equations displayed in set of equations (1) through (3). If these are in fact equivalent, the principle behind the bond graph method advantages is soundly established.

In order to prove this point let's compare equations (4), (5), (8), and (11) with the set of equations (12) through (15). These equations are in the Cauchy form of the differential first order equations, but can be arranged into matrix form for the typical state variable representation of the dynamics of the system. In comparing the two sets, let's arrange them into matrix form.

$$\begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_3}{dt} \\ \frac{dy_2}{dt} \\ \frac{dy_4}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_1}{m_1} & \frac{k_1}{m_1} & -\frac{b_1}{m_1} & \frac{b_1}{m_1} \\ \frac{k_1}{m_2} & (-\frac{k_1}{m_2} - \frac{k_2}{m_2}) & \frac{b_1}{m_2} & -\frac{b_1}{m_2} \end{bmatrix} \begin{bmatrix} y_1 \\ y_3 \\ y_2 \\ y_4 \end{bmatrix} \quad (16)$$

State Space System Matrix A

Now the computer generated matrix from CAMP-G using equations 12-15

$$\begin{bmatrix} \frac{dq_2}{dt} \\ \frac{dq_9}{dt} \\ \frac{dp_{12}}{dt} \\ \frac{dp_5}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{I_5} \\ 0 & 0 & -\frac{1}{I_{12}} & \frac{1}{I_5} \\ 0 & \frac{1}{C_9} & -\frac{R_8}{I_{12}} & \frac{R_8}{I_5} \\ \frac{1}{C_2} & -\frac{1}{C_9} & \frac{R_8}{I_{12}} & -\frac{R_8}{I_5} \end{bmatrix} \begin{bmatrix} q_2 \\ q_9 \\ p_{12} \\ p_5 \end{bmatrix} \quad (17)$$

These two systems are equivalent, but use different sets of state variables. Substituting the physical variable definitions of the bond graph variables, position (q) and momentum (p), in the bond graph system should produce the other state variable system. Consider then the first row of (17), equation (12), which should be equivalent to equation 5:

$$\frac{dq_2}{dt} = -\frac{1}{I_5} p_5 \rightarrow \frac{dx_2}{dt} = -\frac{1}{m_2} m_2 \dot{x}_2 \quad (18)$$

$$\therefore \frac{dy_3}{dt} = y_4 \quad \text{which is equation (5)} \quad (19)$$

Now consider the second row of (17), equation (13), equivalent to equation (4):

$$\frac{dq_9}{dt} = \frac{p_5}{I_5} - \frac{p_{12}}{I_{12}} \rightarrow \quad (20)$$

$$\frac{dx_2}{dt} - \frac{dx_1}{dt} = \frac{1}{m_2} m_2 \dot{x}_2 - \frac{1}{m_1} m_1 \dot{x}_1 \quad (21)$$

Rearranging yields

$$\frac{dx_1}{dt} = \frac{dx_2}{dt} - \frac{1}{m_2} m_2 \dot{x}_2 + \frac{1}{m_1} m_1 \dot{x}_1 \quad (22)$$

$$\therefore \frac{dy_1}{dt} = y_2 \quad \text{which is equation (4)} \quad (23)$$

Now consider the third row of (17), equation (14), equivalent to equation (8):

$$\frac{dp_{12}}{dt} = \frac{1}{C_9} q_9 - \frac{R_8}{I_{12}} p_{12} + \frac{R_8}{I_5} p_5 - m_1 g \quad (24)$$

$$m_1 \ddot{x}_1 = k_1(x_2 - x_1) - \frac{b_1}{m_1} m_1 \dot{x}_1 + \frac{b_1}{m_2} m_2 \dot{x}_2 - m_1 g \quad (25)$$

$$\frac{dy_2}{dt} = \frac{k_1}{m_1} (y_3 - y_1) - \frac{b_1}{m_1} y_2 + \frac{b_1}{m_1} y_4 - \frac{m_1 g}{m_1} \quad (26)$$

Which is equation (8). And finally the fourth row of (17), equation (15) should be equivalent to (11):

$$\frac{dp_5}{dt} = \frac{1}{C_2} q_2 - \frac{1}{C_9} q_9 + \frac{R_8}{I_{12}} p_{12} - \frac{R_8}{I_5} p_5 - m_2 g \quad (27)$$

$$m_2 \ddot{x}_2 = k_2 x_2 - k_1(x_2 - x_1) + \frac{b_1}{m_1} m_1 \dot{x}_1 - \frac{b_1}{m_2} m_2 \dot{x}_2 - m_2 g \quad (28)$$

$$\frac{dy_4}{dt} = \frac{k_2}{m_2} y_3 - \frac{k_1}{m_2} (y_3 - y_1) + \frac{b_1}{m_2} y_2 - \frac{b_1}{m_2} y_4 - \frac{m_2 g}{m_2} \quad (29)$$

Which is equation (11). This demonstrates that there is a one to one correspondence in the systems of equations from the bond graph technique and the state variable equations developed from Newton's laws. Thus, in this case, the computer aided modeling approach is a precise method for generating models automatically and reliably. This approach is extensively explained in reference 4.

There is also another consequence of this demonstration. Since MATLAB and SIMULINK utilize the state space form to describe dynamic systems, it follows that once the files are generated in MATLAB from the bond graph in (.m) files form, all the tools of MATLAB and SIMULINK are immediately available for use on the computer generated model. Moreover reference 6 shows that from this symbolic computer generated state space form, computer generation of block diagrams in SIMULINK can be done. In block diagram form, most engineers and technical staff can understand the model whether or not they have an understanding of bond graph modeling or not.

AIRCRAFT ARRESTING MECHANISM

Shown in figure 6 is a schematic diagram of the aircraft arresting system for a plane landing on an aircraft carrier. The system consists of the plane, its arresting cables, pulleys, dampers and a water squeezer that provides a high nonlinear damping capacity to be able to arrest the plane in the short distance required while at the same time keeping the arresting system sound and safe.

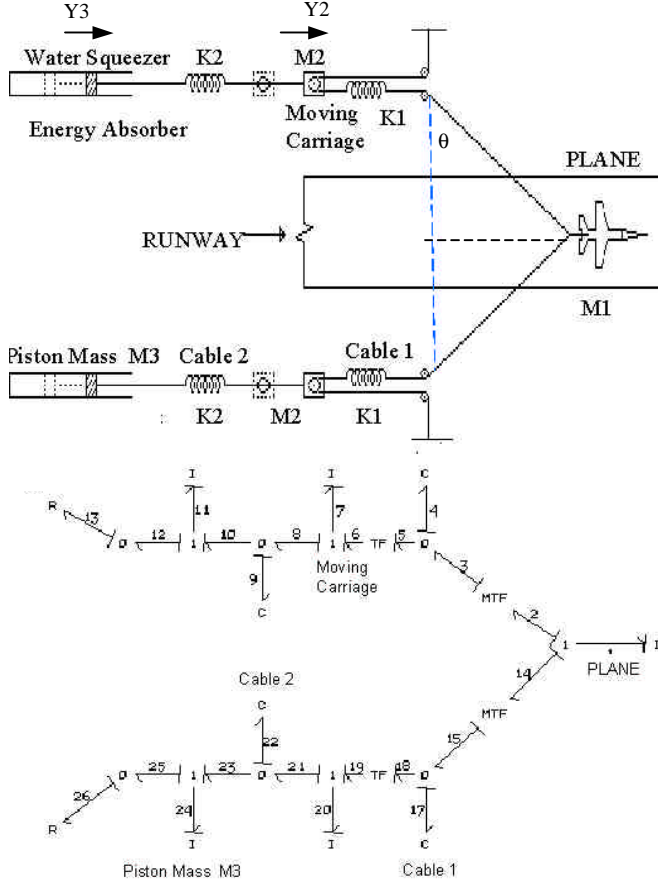


Figure 6. Aircraft hydro-mechanical arresting system.

The design objective is to determine the range of aircraft weights and speeds that can be accommodated without exceeding the working limits of the cables or the allowable piston travel. The airplane comes in contact with the cable that is connected to a set of pulleys. A water squeezer performs as an energy dissipater. The cables connecting the pulley system and the water squeezer can stretch. The appropriate compliance and damping effects are considered.

In order to appreciate the whole process, let's proceed with the complete analysis of this system considering the non-linear geometry as well as the non-linear effects of the water squeezer. Using the model the physical parameter relations and geometry of the system are considered first to determine the governing relations between the physical elements.

1. Cable tension

$$f_{k1} = \begin{cases} K1 (Y1 - 2Y2) & Y1 > 2Y2 \\ 0 & Y1 < 2Y2 \end{cases} \quad (30)$$

$$f_{k2} = \begin{cases} K2 (Y2 - Y3) & Y2 > Y3 \\ 0 & Y2 < Y3 \end{cases} \quad (31)$$

2. Water squeezer force

The drag force from the water squeezer is given by the following expression:

$$fD = f(Y3) \left(\frac{dy}{dt} \right)^2 \quad (32)$$

Where fD is the force and $f(Y3)$ is determined by a specified function. This is the governing non-linear constitutive relation of the water squeezer.

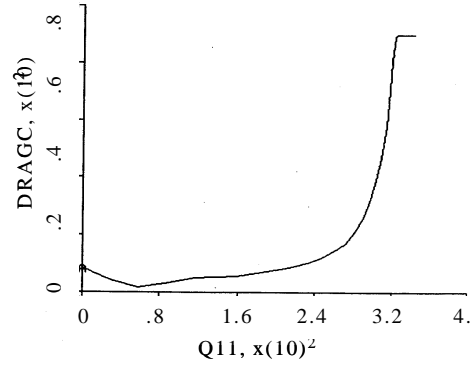


Figure 7. Water squeezer properties.

3. System Geometry

Since bond graph modeling techniques require knowledge of the kinematics interactions among components of the system (velocity relations), this is a fundamental step in the correct representation of the model. Figure 8 shows the details of the system geometry used to define the following expressions:

$$\text{Cable elongation, } Y1 = \sqrt{(x^2 + h^2)} - h \quad (33)$$

$$\text{Cable angle, } \sin(q) = \frac{x}{h + Y1} = \frac{x}{\sqrt{x^2 + h^2}} \quad (34)$$

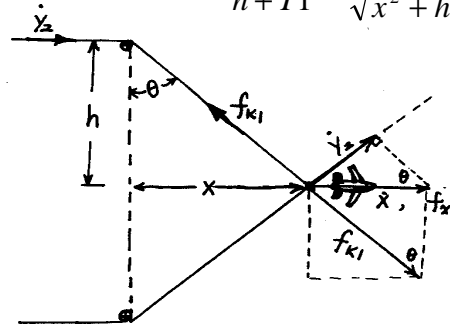


Figure 8. Geometry of Cables operation

The initial conditions are:

$$\begin{aligned} \dot{Y}_3 &= 0 & Y_3 &= 0 \\ \dot{Y}_2 &= 0 & Y_2 &= 0 \\ \dot{X}_0 &= 200 \text{ ft/sec (Run 1)} & X(0) &= 0 \\ &= 250 \text{ ft/sec (Run 2)} \\ &= 300 \text{ ft/sec (Run 3)} \end{aligned} \quad (35)$$

The runway length is restrictive. In this case the available runway length is 1000 ft thus necessitating the arresting system. The layout of the bond graph shown in figure 6 is similar to the schematic of the physical arresting system making model components easily identified with their physical counterparts. In the bond graph model the airplane is treated as an inertia element. Geometric relations drive the kinematics from attitude to velocity. The kinematics transformations of the velocity components dX/dt and $dY2/dt$ are modeled as modulated transformers (MTF) since the kinematics depends on the change of $\sin(\theta)$. The compliance of the cable is modeled as a non-linear C element. A non-linearity of the system is that cables transmit only tension and not compression. The pulley system requires another fixed kinematics transformation and is modeled as a TF element. The moving carriage is modeled as an inertia element (m2). The compliant cable between the water squeezer and the carriage is again modeled as a non-linear C element that does not allow compression. The water squeezer itself has its own piston inertia effect (m3), which is modeled as a non-linear energy dissipater.

The drag coefficient is dependent on a specified function of the piston displacement in the water squeezer. The function is specified by a set of values, which the computer has to interpolate and use at integration intervals. The values of the function are represented in the model in the form of table definitions, which a simulation language can process. The simulation program performs an automatic linear interpolation as needed.

RESULTS ANALYSIS

Once the model has been translated and compiled, it is possible to perform several simulations with different physical parameters or external inputs. In this case a model is studied to test the cable forces, displacements and other variables when the airplane lands at different speeds. The variables used are displayed in bond graph notation, which corresponds to the physical variables shown in Table 1. Partial results from three runs at different aircraft speed are tabulated in Table 2.

Table 1. Physical Variables

FIIN = Aircraft landing Velocity	Q1 = Airplane displacement
F1 = Force acting on the Airplane	THETA = Cable angle
Q4 = Front cable elongation	E4 = Cable force
Q9 = Rear cable elongation	E9 = Rear cable force
Q11 = Water squeezer displacement	E13 = Water squeezer drag force

Table 2: Simulation Results

Variable	Run 1	Run 2	Run 3	Units
FIIN	200	250	300	ft/sec
Q1	564.28	587.91	604.73	ft
F1	12.32	11.09	10.18	ft/sec
THETA	77.51	77.99	78.32	degrees
Q4	2.34	4.362	6.2	ft
E4 (max)	12773.28	19846.36	28204.55	lb
Q9 (max)	0.815	1.275	2.05	ft
E9	20621.64	34607.88	51878.54	lb
Q11	226.45	238	246.24	ft
E13	17990.46	31135.38	46190.17	lb

1. Model validation

Other modeling methods and simulation languages have been used to study this model, and thus it serves as a benchmark. The results of reference 2 are identical to those obtained herein; thus they verify the validity of the bond graph model and the accuracy of the simulation results that have been developed for this case, as compared to the techniques of reference 2.

2. Graphical Results

Using the case for which the initial velocity is 300 ft/sec, the graphical output of the simulation is shown below. Figure 9 shows the displacement, velocity and acceleration of the airplane as it lands.

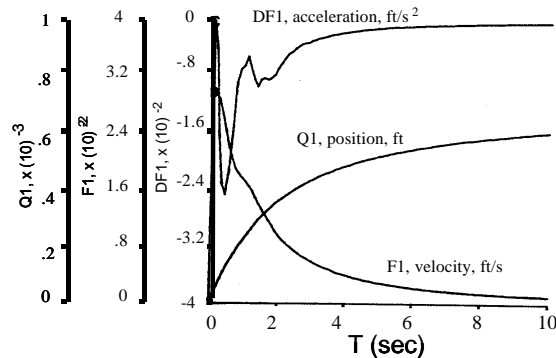


Figure 9. Airplane displacement (Q1), velocity (F1), and acceleration (DF1).

The chart clearly shows a correspondence with reality as the transient engagement takes place and then the velocity goes to zero. The results shown in Table 2 show variables other than design variables and determine the safety of the arresting system. Other important variables can be displayed. The angle that the cables form with the reference line, the cables

stretch and the water squeezer piston position are shown in figure 10.

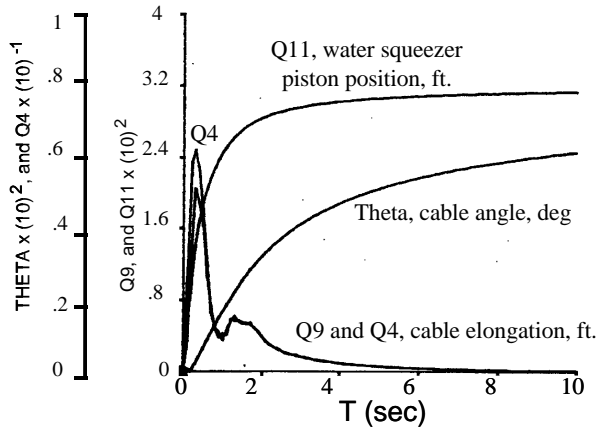


Figure 10. Cable angle, THETA, cable elongation, Q4 and Q9, and water squeezer piston position, Q11.

Design of the water squeezer requires knowledge of how the dissipating force and piston position change with time. Figure 11 shows these variations.

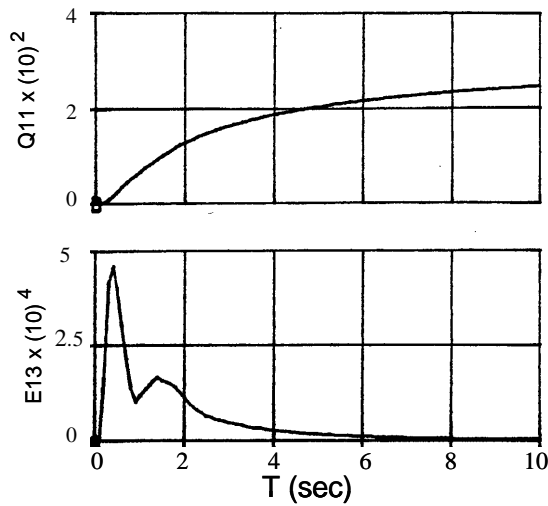


Figure 11. Water squeezer dissipating force (E13) And piston displacement (Q11).

Finally it is important that cable tensions not exceed their specifications. CAMP-G generates the code of all the forces (efforts), all the flows (velocities) and state variables of the different components that can be modified by the user to plot variables of interest.

AIRCRAFT DYNAMICS

Bond graphs can represent rigid body motion so that rigid body airplane dynamics can be modeled with the

technique. Figure 12 shows a small airplane and the system of coordinates. The motion of this airplane is described and controlled by a set of Euler's equations, for translation and for rotation. These equations are shown in figure 13.

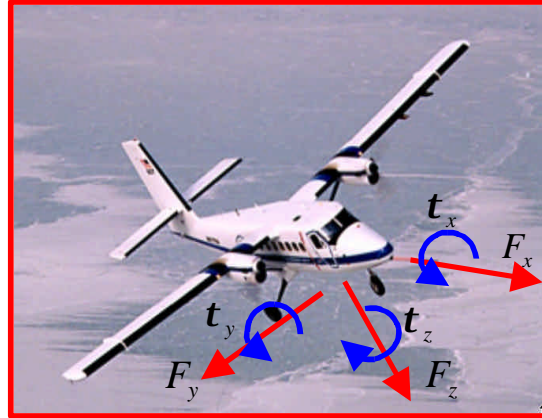


Figure 12. - The physical system.

$$\begin{aligned}
 F_x &= m\dot{v}_x + mw_y v_z - mw_z v_y \\
 F_y &= m\dot{v}_y + mw_z v_x - mw_x v_z \quad (\text{Forces}) \\
 F_z &= m\dot{v}_z + mw_x v_y - mw_y v_x \\
 t_x &= J_x \dot{w}_x + w_y J_z w_z - w_z J_y w_y \\
 t_y &= J_y \dot{w}_y + w_z J_x w_x - w_x J_z w_z \quad (\text{Moments}) \\
 t_z &= J_z \dot{w}_z + w_x J_y w_y - w_y J_x w_x
 \end{aligned}$$

Figure 13. The Euler equations.

These Euler equations are represented by the bond graph shown in figure 14. Using the approach outlined here, this bond graph model can serve as a modular representation of a rigid body in three-dimensional motion for the translational or the rotational components.

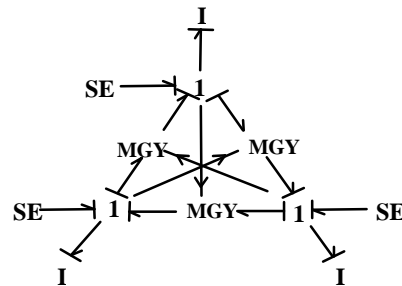


Figure 14. Bond graph equivalent to the Euler equations.

In the case of an airplane, we have a single rigid body. However this bond graph model can also be attached to one that represents flexibility of the wings for example. Reference 1 and 3 offer a detailed discussion how a

rigid body can be modeled together with a set of flexible bodies. The intention in showing this here is not to perform a complete simulations of an airplane but to illustrate the capabilities of the bond graph method to do that because using this basic model the rest of the airplane can be developed but that belongs to a future paper.

SPACE STATION APPLICATION UPDATE

As a first step in applying bond graph techniques to the space station, reference 1 dealt with modeling a rigid central body and two flexible appendages in planar motion. The next step to be undertaken is to model the full six-degree of freedom dynamics of two rigid bodies attached at a point on both bodies and allowing only a single rotational degree of freedom in relative motion. For this simplified problem one of the rigid bodies might represent the space station core body and the other, say, a Photo-Voltaic Array, PVA. The model development is significantly more complex than that attempted before and follows the outline steps provided in reference 3. These steps and progress in the task are reported herein.

First, from the bond graph point of view, the complete dynamics of a rigid body, translational and rotational can be represented as in figure 15, which was generated using Camp-G (reference 4 and 6). To achieve such a simplified graph the assumption of a body-fixed, centric, principle axes must be used. In figure 15, u,v,w are the components of linear velocity of the CG and p,q,r are the components of angular velocity of the body. In figure 12, the background dots are reference points in Camp-G at which one may place bond graph elements.

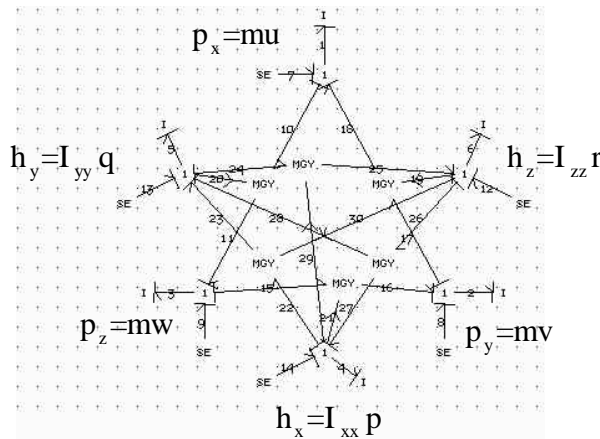


Figure 15. The complete dynamics of a rigid body.

The dynamics of an arbitrary point, A, on the body can thus be represented by the graph of figure 16. Note that

the two constant transformers that provide inputs to the u_A zero junction reflect the contribution of the angular velocity of the body to the x-velocity component of the point A due to the moment arm from the center of gravity to the point, and are thus constant in body-fixed coordinates for the rigid body.

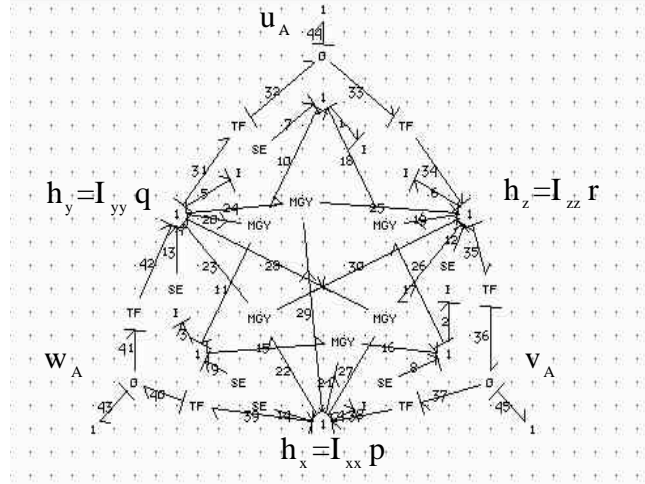


Figure 16. The complete dynamics of fixed point A on a rigid body.

Following reference 3, one must write the relative deflection of the bodies, both linear and angular, in terms of the coordinates of the two bodies, i.e. the differences between the positional coordinates of point A on body one, both the linear and angular position coordinates, and those of the desired attachment point, say B, on the other body. Six equations will result that define the relative linear and angular motion between the bodies, δ in the notation of reference 1, and they will involve the six coordinates of the inertial kinematics of each body, i.e. twelve variables.

These equations can be derived symbolically via the CAMP-G/MATLAB system. This relates the time derivatives of relative linear and angular displacements between the two bodies to the *inertial* linear and angular velocities of each individual body. The equations will have a generalized symbolic linear form in the state variables of the individual bodies, but the coefficients of these terms will be highly nonlinear, involving the complete coordinate list, say x .

As in reference 3, when the equations are put into matrix form of $d\delta/\delta\tau = T(x) dx/dt$, the transpose of the matrix $T(x)$ theoretically and automatically provides the relationship between the relative gap forces, λ , and the forces and moments applied to the two individual bodies relative to their respective centers of gravity.

On the space station, electrical motors are used to control the angular motion of appendages. And it is this approach to control of the relative motion between the

bodies that is assumed here. The mass and inertia characteristics of the motor stator mechanism is assumed to be included in that of body one, the core body, while similar characteristics for the rotor mechanism are assumed included in body two, the PVA.

Figure 17 is a schematic of the motor model. We assume that the relative gap forces, F and T in the figure, (λ in reference 2) are the stator and rotor bearing forces and that the control torque, T_e , is applied to the stator and rotor in the sense indicated in the figure.

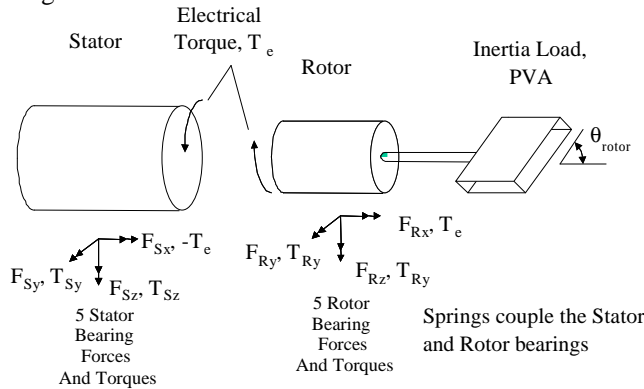


Figure 17. Schematic of the motor model.

Figure 18 illustrates the application of the control torque, T_e , to the Stator/Core body showing the effects of relative bearing deflections of the rotor. Again, the mass and inertia characteristics of the motor stator are included in that of the core body. A similar figure can be generated for the PVA bearing forces and torques and the control torque, but this is left for the reader in the interest of saving page space.

Finally, using the Star-of-David notation for a rigid body, used in reference 3, a schematic of the complete bond graph model for the problem is presented in figure 19. Stiff springs with dampers are used to model the gap/bearing forces and torques. Further, it is assumed that the electrical control is input as a pure torque source of effort, S_E , so that the stator and rotor gap forces of figure 17 are the same but opposite in sign. Thus the S and R subscript notation can be dropped in figure 19 and replaced with gap, or g, subscripts.

The effects of bearing alignment on the application of torque (figure 18) are represented component-wise. The electrical torque, S_E , is resolved into components along the nominal rotation axis of the rotor and normal to it, i.e. the bearing forces and torques. The bearing components are included in the high stiffness bearing springs representing the gap.

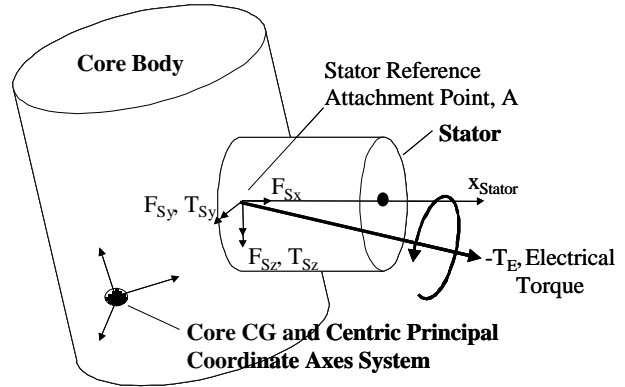


Figure 18. Control and bearing forces for the Stator.

Detailed modeling of the electrical control torque is not included herein since it is a well-solved modeling problem whose structure depends significantly on the types of electrical amplifiers used.

CONCLUSION

In this paper we have presented an overview of the bond-graph modeling technique, which offers a compact graphical representation for modeling of Mechatronics physical systems. Software tools are now available to aid the engineer in developing bond graphs. CAMP-G is one such tool that interfaces directly with MATLAB/SIMULINK, which additionally provides a simulation capability and the symbolic manipulator needed to develop the nonlinear transformers that are required for practical implementation of complex systems.

The aircraft-landing model presented illustrates the whole process from conception to simulation to graphical results analysis. The bond graph model offers a unique multi energy approach that has been verified using results of the same model by other methods. In making the comparison one realizes clearly that the set of equations produced by the bond graph method resulted directly in state space form without the need to transform the second order equations using auxiliary variables. Such is a long process if one derives the equations manually with the intention to program them into MATLAB. The computer-automated approach presented here contributes greatly in getting the model from reality to simulation quickly and precisely.

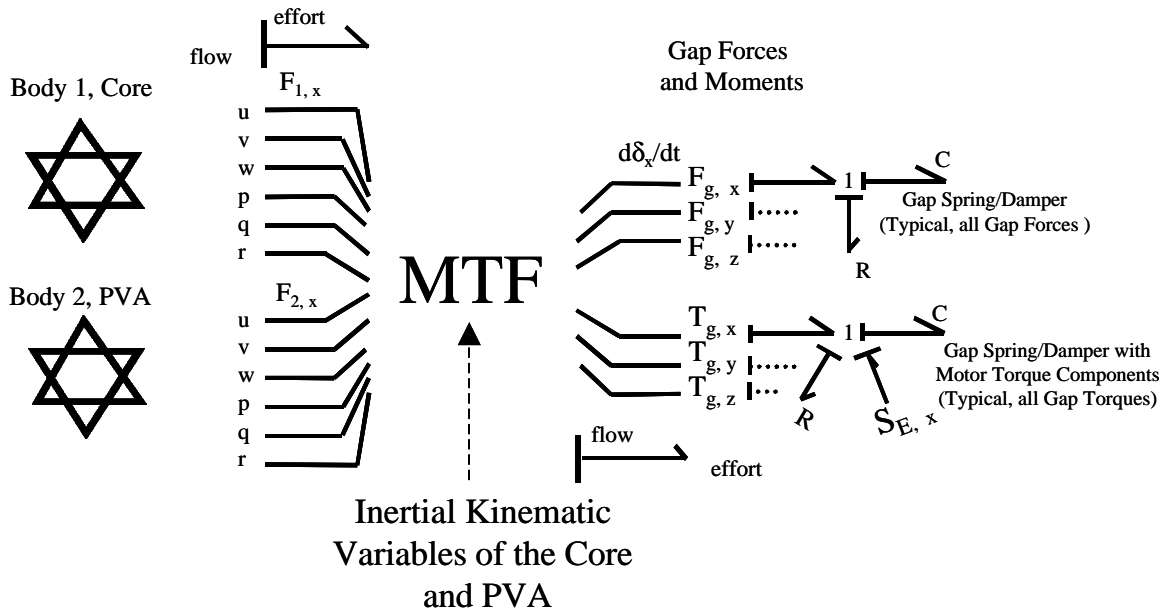


Figure 19. A schematic diagram of the complete two-body bond graph for three-dimensional space

Using this approach, the engineer does not even have to generate the model on paper because the software will allow him to edit the bond graph, or save it for future use. It allows the addition or the deletion of components to the model. It allows also the interconnection of elements in different energy domains. This means that using the CAMP-G/MATLAB, CAMP-G/SIMULINK or other like CAMP-G-ACSL one can generate complex models with applications to the aircraft industry. It is obvious that this approach demonstrates clearly the creation of Mechatronics models in an integrated fashion thus allowing the analysis of models of different energy domains and of different levels of complexity with single integral model.

REFERENCES

1. Montgomery, R. C. and Granda J. J.: Using Bond Graphs for Articulated, Flexible Multi-bodies, Sensors, Actuators, and Controllers with Application to the International Space Station. Proceedings of the 2003 International Conference on Bond Graph Modeling and Simulation ICBGM'2003. ISBN: 1-56555-257-1, Edited by Francois Cellier and Jose J. Granda, Volume 35, No. 2, pp. 239 – 246.
2. Mitchell and Gauthier, Associates, Inc. ACSL User's Manual. 2000.
3. Karnopp, D. C., Margolis, D. L., and R. C. Rosenberg. System Dynamics. Modeling and Simulation of Mechatronics Systems. Wiley and Sons 2000, pp. 307-324.
4. Granda J. J.: Computer Generation of Physical System Differential Equations Using Bond Graphs. J. Franklin Institute. January, 1985
5. Karnopp D. C.: Direct Programming of Continuous System Simulation Languages Using Bond Graph Causality. Transactions. SCS, Vol. 1, July 1984.
6. Granda J. J. "Computer Generated Block Diagrams from Bond Graph Models CAMP-G as a Tool Box for SIMULINK" Proceedings of the International Conference on Bond Graph Modeling and Simulation ICBGM'2001 Phoenix, Arizona. January 2001.