

# Creating Flight Simulator Landing Gear Models Using Multidomain Modeling Tools

Terry Denery<sup>1</sup>, Jason R. Ghidella<sup>2</sup>, Pieter J. Mosterman<sup>3</sup>, and Rohit Shenoy<sup>4</sup>  
*The MathWorks, Inc. Natick, MA, 01760*

**This paper uses commercial off-the-shelf (COTS) domain specific modeling software to create a high fidelity plant model of an aircraft's landing gear for inclusion into a full aircraft flight simulator. The use of domain specific modeling software enables detailed modeling of the physics and facilitates accurate computational simulation of the aerodynamic and mechanical loads that occur when the landing gear are deployed and retracted during landing and take-off operations. The parameter design space is easily searched by considering a number of different landing scenarios including touching down on one wheel first, to optimize the design.**

## Nomenclature

$A_b$	=	acceleration in the body-fixed frame
$V_b$	=	velocity in the body-fixed frame
$V_e$	=	velocity in the Earth-fixed reference frame
$X_e$	=	position in the
$\omega$	=	angular velocity in the body-fixed frame
$d\omega/dt$	=	angular acceleration in the body-fixed frame

## I. Introduction

COMPUTATIONAL models have become essential to the successful design of modern engineered systems such as aircraft, automobiles, and consumer electronics such as mobile telephones. The use of a computational model as a design tool allows searching a much larger parameter space more quickly than what would be possible by hardware realizations and this has led to products with highly optimized performance and efficiency.

To efficiently employ models for design purposes, it is important to capture the physics of the envisioned realization in as much detail as possible so the design is an accurate reflection of the eventual product. This prevents unanticipated behavior and expensive re-design iterations once the product is realized and the unmodeled phenomena manifest themselves.

The challenge of the modeler then is to identify and capture a sufficient level of detailed dynamic behavior while maintaining computational performance. This detailed behavior may relate to high order phenomena such as, for example, inertial effects of a leaking fluid valve, as well as parasitic phenomena such as, for example, electromagnetic waveform propagation. In many cases, these behaviors may be 'non-functional', i.e., not part of the designed behavior but critical to successfully produce and deploy a system.

Using general purpose modeling languages to capture the complexity of such detailed model proportion is difficult because of the generic concepts that have no immediate bearing on the domain in which the system under design operates. For example, to use general purpose differential equations for modeling physical systems requires great insight and discipline not to inadvertently neglect including constraints that are inherent in the physical domain such as conservation of energy.

---

<sup>1</sup> Principal Application Engineer, 3 Apple Hill Drive, Natick, MA, 01760, AIAA member

<sup>2</sup> Senior Team Lead, Technical Marketing, 3 Apple Hill Drive, Natick, MA, 01760, AIAA member

<sup>3</sup> Senior Research Scientist, 3 Apple Hill Drive, Natick, MA, 01760, AIAA member

<sup>4</sup> Technical Marketing Manager, 3 Apple Hill Drive, Natick, MA, 01760

Domain specific modeling tools aim to address these inherent shortcomings of general purpose tools. For example, SimMechanics<sup>1</sup> is a tool that is dedicated to modeling multi-body systems, for which it includes specific language elements such as bodies that can be connected by different types of joints, a variety of constraint drivers, and the like. This allows the modeling of a multi-body system to proceed using semantic notions that are closely related to the specific domain, and the model will closely reflect the topology of the physically connected elements.

Often, domain-specific tools such as SimMechanics are based on modeling the exchange of energy between elements, and with power being the time-derivative of energy, this results in capturing the power connections. Since power is the product of two conjugate variables, for example force and velocity, the connections represent two variables. In a computational sense, one of these tends to be required to compute the other and so there is an opposing flow of computation: Where the force along a connection may be computed from one connected element, the velocity is computed based on the behavior of the other.

In traditional modeling tools that are based on signal flow, such as Simulink<sup>®2</sup>, the two conjugate variables have to be explicitly identified as separate signals and the flow of computation has to be established by the model designer so that the model can be fit into the general purpose and explicit representation. This is a tedious and error-prone process and may prevent the construction of models with the level of detail that is desired.

To further support the modeling task, domain specific tools may provide model components that embody complex and detailed behavior with standardized characteristics. For example, the Aerospace Blockset<sup>3</sup> contains components that model wind gusts, shear, and turbulence all based on reference standards. Such support for compositionality is essential in addressing the design challenges of modern engineered systems.

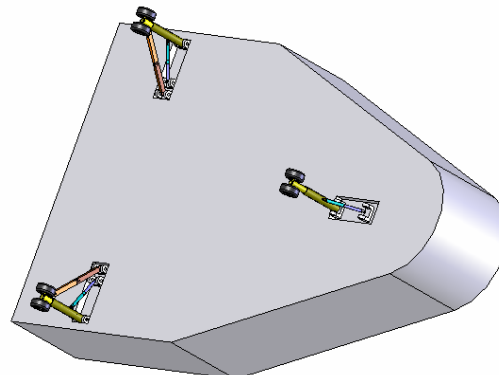
Once a detailed model has been obtained that is structurally correct, additional accuracy can be attained by tuning the model parameters with real data acquired from the actual system that is being modeled. This is especially important when combining models of phenomena such as friction with detailed physical models. Friction models, which are not easily designed, are often obtained by using accepted mathematical relationships and then tuning parameter values till the behavior matches the physical system being modeled. One tool that can help simplify this process is Simulink Parameter Estimation<sup>4</sup> which uses optimization techniques to tune parameters of detailed models with Input/Output data. In the case of friction on a joint, the joint can be actuated with a predefined force profile and the resulting displacement can be measured at regular intervals. The force inputs and resulting displacement measurements can then be used to tune the friction model.

With the availability of detailed models, Model-Based Design facilitates integrating the different aspects of the overall design task such as requirements capture, control synthesis, multi-objective response optimization, system identification and parameter estimation, and code generation for software or hardware implementations.

This paper presents the use of multi-domain modeling for the study of aircraft landing gear. In particular, mechanical and aerospace-specific model parts are designed and integrated into a comprehensive model of the landing gear behavior. A scenario where the aircraft lands on one wheel set first is analyzed in terms of applied forces required to perform this maneuver safely and effectively.

## II. Modeling Landing Gear in the HL-20 Flight Simulation Model

This work extends a model of the NASA HL-20 lifting body airframe that is an example case study in the Aerospace Blockset. The model described in the case study was based on previous work presented by



**Figure 1. The HL20 has a three point landing gear system that employs a four-bar linkage for each of the aft systems, and a single-bar linkage for the nose gear.**

Jackson and Cruz<sup>5</sup>. The original model did not include landing gear forces and moments. A modified version of the HL-20 lifting body airframe model that is also included in the Aerospace Blockset does include a simplified version of landing gear forces and moments as well as their aerodynamic effects based on work by Banks<sup>6</sup> and Baarspul<sup>7</sup>.

The work presented in this paper elaborates the mechanical model of the HL-20 lifting body airframe model to include direct modeling of the mechanical linkages employed in the landing gear systems.

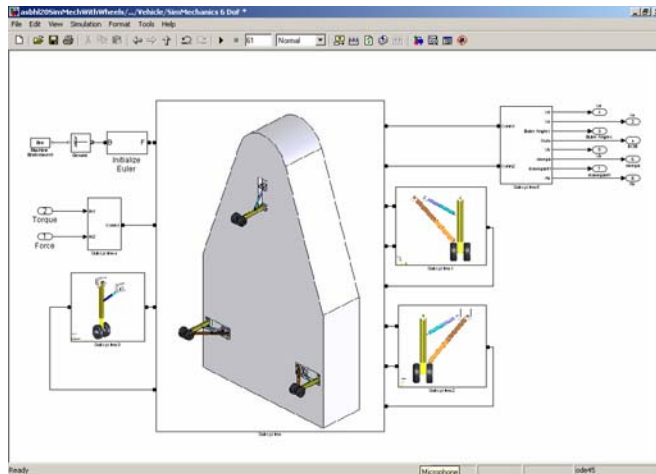
The HL-20 lifting body airframe model includes a detailed flight environment model<sup>3, 8</sup>, which contains rich models for air density, wind, terrain, and gravity that are based upon reference standards used in the Aerospace community and are included as blocks in the Aerospace Blockset. The impact of this highly detailed environment model is ultimately reflected by the accuracy of the calculated gravitational and aerodynamic loads felt by the aircraft vehicle, and will be used to determine the loading of the landing gear as the vehicle touches-down.

The underside of the flight vehicle is shown in Figure 1. This figure shows the mechanical linkages that make up the landing system. The mechanical model for this work was elaborated upon through the use of SimMechanics, which enables efficient modeling and simulation of full 3-dimensional mechanical systems in Simulink. It is based upon the principles of rigid-body dynamics and designed for general motion analysis of mechanical assemblies containing internal linkages. SimMechanics and the solution method that it employs are described in a paper by Wood and Kennedy<sup>9</sup>.

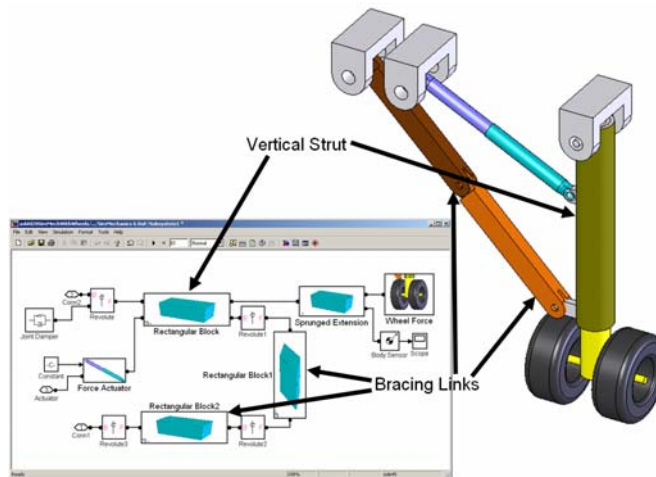
Modeling the mechanical linkages in the same environment as the aerodynamics requires an interface or connection between the two domains to be established. Practically speaking, this means forming a connection between the SimMechanics modeling of the mechanics and the Aerospace Blockset modeling of the Aerodynamics. The key to establishing this connection is to build a mechanical model that satisfies the interfaces set up in the Aerospace Blockset for attaching to the aerodynamics and gravity models.

### A. Mechanical Modeling in Simulink<sup>®</sup> with SimMechanics

Figure 2 and Figure 3 show the mechanical model of the flight vehicle. These figures display the block diagram modeling environment of SimMechanics. This modeling environment offers blocks for defining bodies, joints, and constraints to describe the structure of a mechanical assembly, and it follows the process for organizing models in Simulink that is enabled through the use of subsystems. The joint blocks are defined by assigning the orientation of the relevant revolute and prismatic axes of the joint. The body



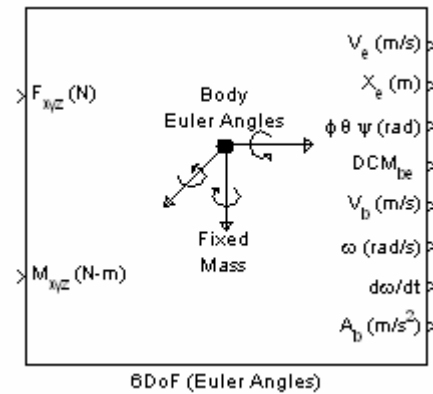
**Figure 2. The flight vehicle mechanical model is built in Simulink<sup>®</sup> with SimMechanics and is composed of the main airframe and 3 landing gear linkage subsystems.**



**Figure 3. The SimMechanics model directly corresponds to the main landing gear mechanical linkage components where each moving part is represented by a body block and each connection by a joint block.**

blocks are defined by assigning mass and inertia properties for the body, and by assigning local coordinate systems, which establish the center-of-gravity (CG) location and mark locations for joint attachments and provide a reference for joint orientations. The lines that join blocks simply represent physical connections, unlike traditional Simulink models, where the lines form a computationally directed mathematical relationship between the blocks.

The modeling environment also provides blocks for actuation and sensing. The sensing blocks extract mechanical response and performance data from the simulation and make it available to traditional Simulink blocks as Simulink signals. The actuation blocks take Simulink signals from traditional Simulink blocks and apply them to actuate motion through assignments for forces, torques or prescribed motions. It is through the sensing and actuation blocks that a subsystem constructed using SimMechanics blocks can be interfaced to a control algorithm, and into a larger system-level model that can include other components. Further information describing the components, capabilities and use of SimMechanics is available in the SimMechanics User's Guide<sup>1</sup>.

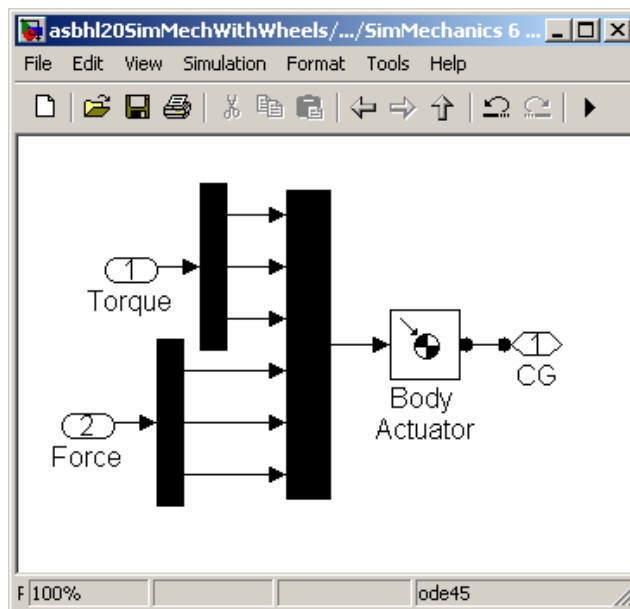


**Figure 4. The 6DoF (Euler Angles) block of Aerospace Blockset is a single-body mechanical model that defines the relationship of the forces and moments applied to the vehicle with its resulting motion.**

### B. Building an Aerodynamic and Mechanical model

The Aerospace Blockset Euler angle implementation of the six degree of freedom equations of motion, (6DoF (Euler angles)), block is shown in Figure 4. This block lets the user represent a flight vehicle with a single-body model. It defines the relationship of the forces and moments applied to the flight vehicle with the vehicle motion. Being a single-body model, it does not directly model the landing gear linkages, and more importantly, account for the internal motion and forces generated by a landing gear system. It does however define the interface needed of any mechanical vehicle model that is constructed to access the air density, wind, terrain and gravity models available in Aerospace Blockset.

So the first step in creating a detailed mechanical model of the landing gear is to create a SimMechanics-based version of the 6DoF (Euler Angles) block. Being SimMechanics-based, this new block will enable the inclusion of the mechanical linkages of the landing gear system. To accomplish this task, the input, output, and initial state and mass property assignment structure of the 6DoF (Euler Angles) block needs to be replicated in the SimMechanics-based 6 DoF block. By matching this format, the process of elaborating the HL-20 lifting body model is a simple matter of exchanging the Aerospace Blockset 6DoF (Euler Angles) block with the SimMechanics 6DoF block, and adding the landing gear linkage model through the use of SimMechanics.



**Figure 5. The SimMechanics Body Actuator Block allows vector signals for torque (moment) and force to be applied to the flight vehicle at its center of gravity.**

### 1. 6DoF Inputs

The inputs for the 6DoF block, as shown in Figure 4, are force and moment vector signals to be applied at the center of gravity (CG) of the vehicle. The Aerospace Blockset 6DoF (Euler angles) block considers the rotation of a body-fixed coordinate frame about an Earth-fixed reference frame. The right-handed Cartesian coordinate system is oriented with positive x aligned with vehicle forward direction through the nose of the vehicle, positive y is aligned with the lateral direction and is pointing to the right of the x-axis and positive z points down through the bottom of the vehicle towards the ground. This choice of coordinate system orientation also has implications on the assignment of the inertia tensor, which is discussed later in this paper.

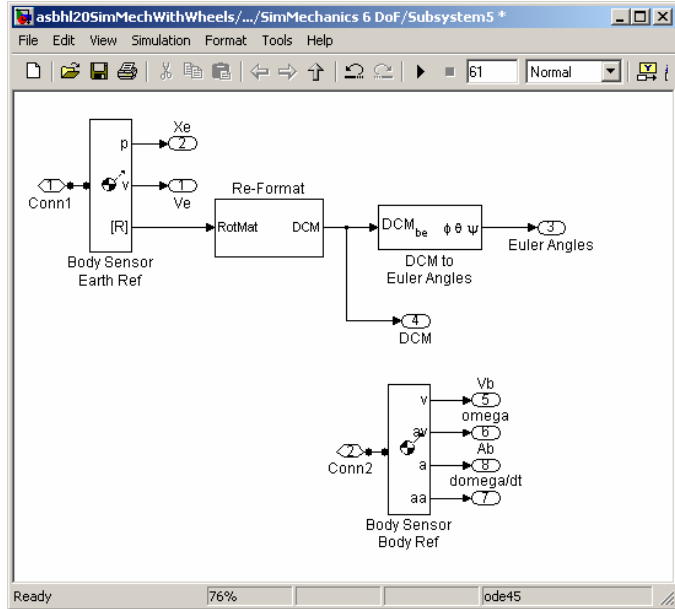
The SimMechanics body block that represents the airframe of the vehicle receives the forces and moments through an actuator block. As is shown in Figure 5, the actuator block connects to the CG port of the body to set the CG as the location where the forces and moments are applied. Through the properties of the block, the assignment is made for a local body-fixed reference frame for interpreting accurately the force and moment information.

### 2. 6DoF Outputs

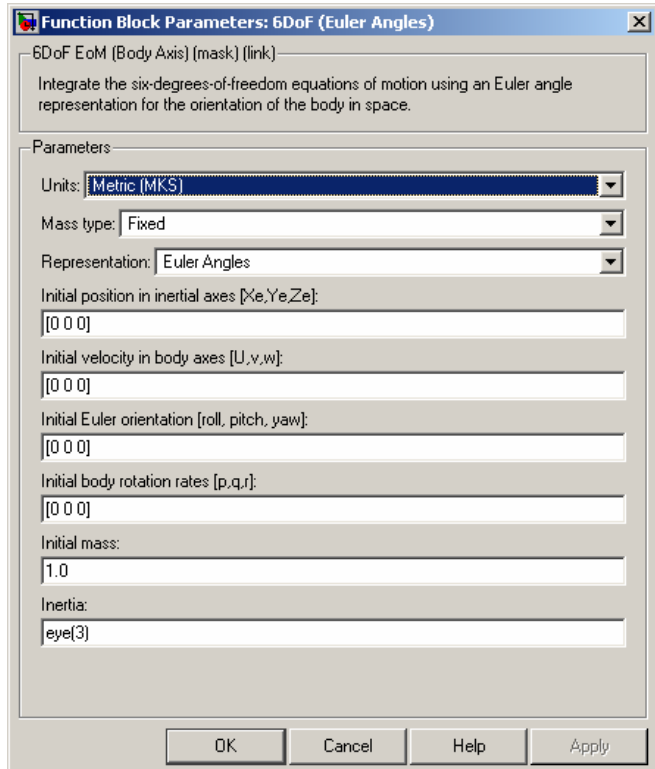
The outputs of the 6DoF block describe position, velocity and acceleration for both the translation and orientation of the flight vehicle, as shown in Figure 4. The translational data is reported through the quantities  $X_e$ ,  $V_e$ ,  $V_b$ , and  $A_b$ . The orientation data is reported through the Euler angles, direct cosine matrix (DCM),  $\omega$ , and  $d\omega/dt$ . The Euler angles and DCM are based on the Earth-fixed reference frame.

Figure 6 shows the signals that are sensed and the formulation that is necessary to reproduce the output of the 6DoF (Euler Angles) block. This figure shows that two sensor blocks are used to capture measurements for both the Earth-fixed and body-fixed reference frames. The formulation is necessary to reformat the orientation measurements to satisfy the required form of the 6Dof (Euler Angles) block.

The rotation matrix reported by SimMechanics uses a convention opposite to that of Aerospace Blockset. The rotation matrix of SimMechanics transforms an expression of a vector in Earth-fixed coordinates to an expression in body-fixed coordinates; the DCM of Aerospace Blockset follows the opposite



**Figure 6. Using SimMechanics sensor blocks the SimMechanics 6DoF system is equivalent to Aerospace Blockset 6Dof (Euler Angles) block output.**



**Figure 7. 6DoF (Euler Angles) block allows assignment of mass properties and initial conditions of the vehicle.**

convention, which makes it the inverse of the SimMechanics rotation matrix. Because these matrices are orthonormal, the DCM is determined by simply transposing the matrix reported by SimMechanics.

These matrices, of course, provide additional value in that they state the current orientation of the flight vehicle. There are many ways to state the orientation of a body, which include the use of quaternions and many standard conventions for Euler Angles. Aerospace Blockset provides a complete library of tools for automating the translation between these different approaches. The Euler angles are reported from the SimMechanics-based 6DoF block through the use of the Aerospace Blockset translation block DCM to Euler Angles.

### 3. Initial State and Mass Property Assignments

Through the parameter dialog box for the 6DoF (Euler Angles) block, Figure 7 shows the assignments required for the initial state and mass properties of the flight vehicle. The properties of mass and inertia are directly assigned to fields available in the Body block of SimMechanics. The initial state assignments are more complicated, as SimMechanics and Aerospace Blockset take different approaches for choosing states.

For SimMechanics, states are chosen from the degrees of freedom that are associated with each joint. For example, the relative rotation and rotational velocity of two bodies joined by a revolute joint may be selected as states. In fact, if there are no closed loops in the mechanical linkage, two states are associated with each degree of freedom for every joint in the system. For the Aerospace Blockset 6DoF block, however, there is only one body and no joints. The states available directly define the position and velocity of the translation and orientation of the flight vehicle.

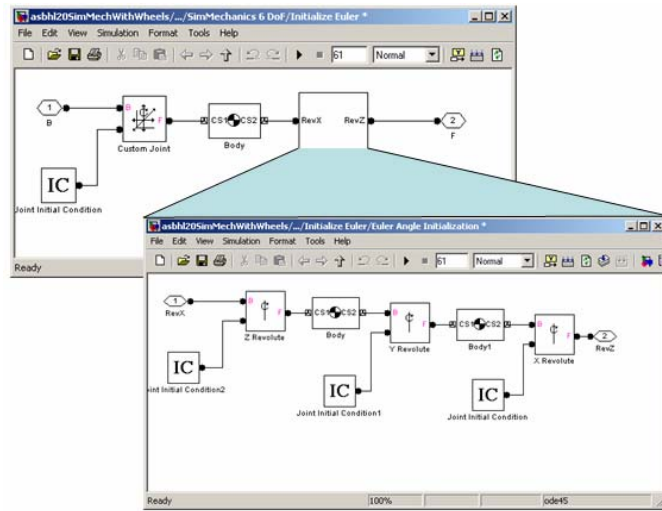


Figure 8. A configuration of massless bodies and revolute joints is used to assign initial states of the flight vehicle.

The difference in state definitions presents a challenge in creating a SimMechanics-based 6DoF that conforms to the characteristics of the Aerospace Blockset 6DoF blocks. The format of the Aerospace Blockset 6DoF (Euler Angles) blocks to define states in terms of body position and orientation requires that the SimMechanics model be configured such that joints exist for initial condition assignments that can express the same overall state as that prescribed by the Aerospace Blockset 6DoF blocks, and at the same time, they cannot introduce any constraints that influence the motion of the system. Once this configuration exists, a transformation of the Aerospace Blockset 6DoF block initial condition data is still needed so that it maps to the initial conditions of the joints in the SimMechanics-based 6DoF block.

To accommodate the initial state assignment, a configuration of joints and bodies is created so that translational position and velocity, and Euler angle and rate assignments can be made that will impart the appropriate initial conditions

```

Mask editor: Initialize Euler
Parameters Initialization Documentation
Dialog variables: Vb, Orient, Omega
Initialization commands:
phi = Orient(1);
theta = Orient(2);
psi = Orient(3);
R1 = [1 0 0; 0 cos(phi) sin(phi); 0 -sin(phi) cos(phi)];
R2 = [cos(theta) 0 -sin(theta); 0 1 0; sin(theta) 0 cos(theta)];
R3 = [cos(psi) sin(psi) 0; -sin(psi) cos(psi) 0; 0 0 1];
R = R1*R2*R3;
RBtoE = transpose(R);
Ve = RBtoE*Vb;
J = [1 (sin(phi)*tan(theta)) (cos(phi)*tan(theta)); 0 cos(phi) -sin(phi); 0
Erates = J*Omega;
    
```

Figure 9. MATLAB® m-code is used to define the initial conditions and relationship between Euler angle rates and the angular velocity of the flight vehicle.

on the flight vehicle body. This configuration is shown in Figure 8. The intermediate bodies enable the appropriate combination of joints, but do not possess mass or inertia, and, therefore, they do not affect the motion of the flight vehicle. A custom joint connects the CG of the flight vehicle to the first intermediate body, and through this joint the translational position and velocity initializations are assigned.

The series of three revolute joint connections, which is made possible by two massless bodies, connects the first intermediate body to a ground connection. This sequence of connections defines rotational angles that map directly to an Euler XYZ convention for orientation assignments. It is through the assignment of the appropriate Euler angles and Euler angle rates for the initial conditions of these joints that the initial state assignment is completed.

The Aerospace Blockset 6DoF (Euler Angles) block assignment for orientation is made through Euler angles employing the Euler XYZ convention. The block assignment for rate of change of orientation, however, is made through the initial angular velocity. The transformation between angular velocity and Euler angular rates is achieved through a matrix multiply. This transformation matrix is defined and the multiplication is executed using MATLAB<sup>®10</sup> m-code that is programmed directly into the SimMechanics-based 6DoF block through an initialization command sequence that is shown in Figure 9.

### C. Modeling Landing Gear Linkages

The elaborated mechanical model for the flight vehicle is shown in Figure 1. This figure shows that the landing gear system is made up of three sets of wheels that contact the ground. Two of the wheel sets are deployed by gear systems located on the aft end of the plane—these are called the main gear. The third wheel is deployed by a nose gear system at the fore of the plane. The aft located main gears employ a four-bar mechanical linkage with shock absorber. The nose gear employs a simple one-bar linkage with a shock absorber. For each gear system, a linear-action actuator deploys the gear and maintains the force to hold the deployed position. The wheel is modeled to account for the contact, rolling and friction between the wheel and the ground.

#### 1. The Main Gear

Figure 3 shows the four-bar mechanical linkage of the main landing gear and the SimMechanics-based Simulink model that represents it. The four-bar linkage consists of three bars in a sequence that is joined via revolute joints. Each end of this sequence is joined to the airframe via revolute joints to form a closed loop four-bar linkage. The shock absorber is implemented through a sprung extension that is sleeved by the vertical strut. The sleeve connection allows a prismatic motion of the sprung extension along the principal axis of the vertical strut. This motion is controlled passively via a spring/damper.

#### 2. The Bar Block

Figure 10 shows the structure of each bar element block. The bar is assumed to have the shape of a rectangular box, through which the appropriate selections for its length, height and width can be made to roughly approximate the mechanical motion properties of links that are typically used for applications of this nature. Through the parameter dialog box of this block, assignments are made for length, height and width, as well as mass density and angle.

Figure 11 shows the parameter dialog box for defining the properties of the Body block of the bar element. This figure shows that appropriate formulae are assigned to the mass and inertia fields to automate their calculation based upon the parameter assignments discussed above. The formulae are written using MATLAB m-code, which can be written to any field of any Simulink block. MATLAB m-code is ideal for

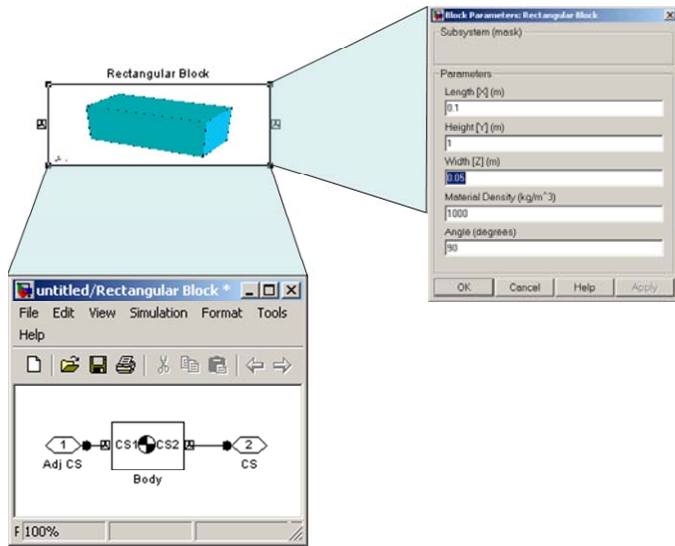


Figure 10. The parameter dialog box and structure of each bar element of the landing gear linkages.

this because it is a natural programming language for working with vector or tensor quantities such as the inertia tensor.

The local coordinate systems for the bar element are listed in the parameter dialog box as shown in Figure 11. One locates the CG and the other two locate the attachment points to the adjacent bodies. In defining the location of the coordinate system CS1, which marks one of the attachment points, an adjoining reference has been selected through the parameter dialog box of the Body block. A revolute joint links CS1 to a position fixed on the joined body, which is marked by a coordinate system local to that body. The adjoining reference simply states that the location of the origin of CS1 is expressed in x, y, and z values defined by the coordinate system of the joined body. The x, y, and z values are chosen as (0,0,0) to collocate CS1 with that coordinate system.

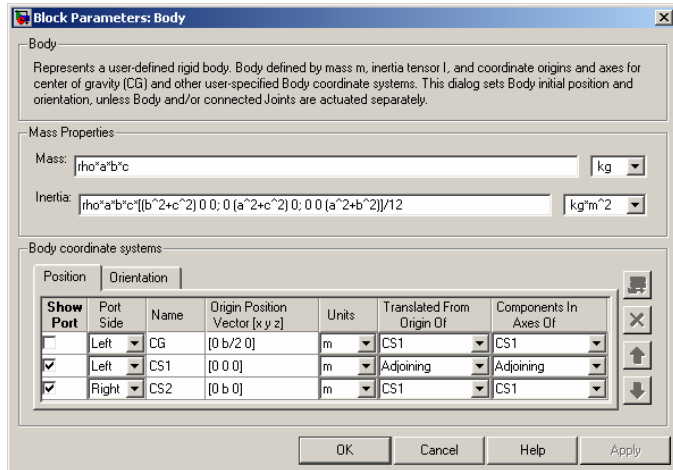


Figure 11. Parameter dialog box for the Body block of each bar element

By way of the adjoining reference, the position of CS1 is defined in the world, or Earth-fixed coordinate system. The rest of the local coordinate systems for the body are positioned relative to it. This is seen in Figure 11, which shows the positional assignments for CG and CS2. These assignments show that the MATLAB variable *b*, which is initialized from the “Height [Y] (m)” parameter of the bar element that is shown in Figure 10, is used in the y-field of the position assignment vector for each coordinate system.

Through the constraints imposed by the revolute joints, the motion of the main landing gear is 2-dimensional and planar relative to the airframe. The orientation of each bar can vary in only one dimension, which is a rotation about an axis perpendicular to this planar motion. The angle assignment

Table 1. Geometric Dimension and Mass Values for Main Gear

	Vertical Strut	Brace 1	Brace 2
<b>Cross Section (m<sup>2</sup>)</b>	0.02	0.02	0.02
<b>Length (m)</b>	1.0	0.707	0.707
<b>Volume</b>	0.02	0.0141	0.0141
<b>Mass Density (kg/m<sup>3</sup>)</b>	7500	7500	7500
<b>Mass</b>	150	106	106

for the initial position of each bar, which is fed through the user interface for defining the bar, sets this orientation value as the angle of CS1 relative to the body to which CS1 attaches.

The bar element block is important because it is the basis for every mechanical linkage element in this landing gear model. It is the portability and reusability of this block that provides its value. This is afforded through the adjoining reference, the formulation of mass, inertia and coordinate system assignments into the body definition, and through the ability to parameterize the block so that geometrical and material assignments can be made for every instance of the block.

Table 2. Properties of the Shock Absorber System

Mass of Strut + Wheels (kg)	300
Spring Constant (N/m)	300,000
Damping Constant (N-s/m)	100,000

### 3. Configuration of Main Gear System

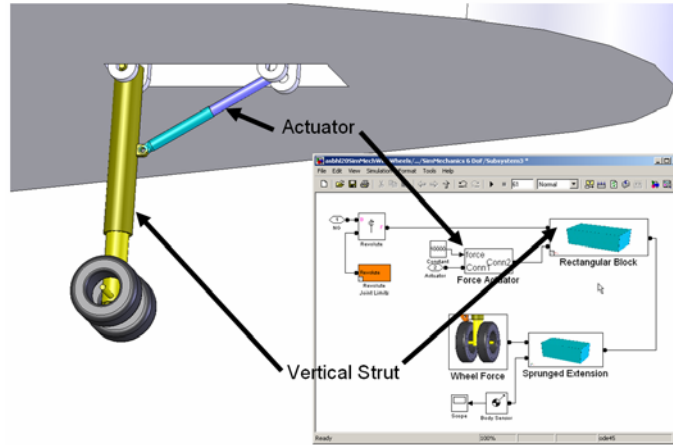
Table 1 lists values for the geometrical dimensions and mass for the main landing gear components. The composition of both of the main gear units is identical, although their configurations are mirror images of each other. Table 3 lists the properties for the shock absorber system, which includes mass, spring, and damper constants.

Not a great amount of research was spent verifying the appropriate dimensions that influence the mass of this system. In comparison to the vehicle mass of 11739 Kg, it appears that the mass of the landing gear contributes in a minor way to the mechanics of landing this system. The geometry, on the other hand, is

critical. It defines the location of landing loads relative to the vehicle CG, and the way the mechanism folds when the gear is retracted. The retractile folding is important because it is a potential failure mode for a one-wheel landing, where side loads on the wheel can initiate folding.

#### 4. The Nose Gear

Figure 12 shows the one-bar mechanical linkage of the nose landing gear and the SimMechanics-based Simulink model that represents it. The vertical strut and shock absorber have identical dimensions to those listed for the main gear. Similarly, the actuator and wheel models are identical for all three gear systems.



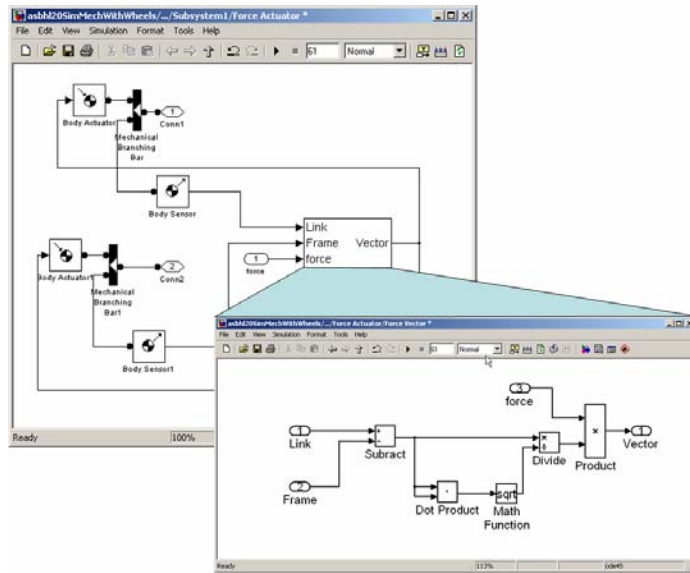
**Figure 12. The SimMechanics model of the nose gear mechanical linkage.**

#### 5. The Landing Gear Actuator

Figure 13 shows the Simulink model for the landing gear hydraulic actuator. It is identical for all 3 landing gears. It is a forced-based model, meaning that opposing forces are applied to the actuator's connection points to the frame and to the linkage. The magnitude of these forces is equal to a force input signal that is delivered from the model. For the application described in this paper, this signal is simply equal to a constant. An additional elaboration of this model might be to add a hydraulic model of the actuator using SimHydraulics<sup>TM</sup><sup>11</sup>. If the actuator force is generated electrically, then SimPowerSystems<sup>12</sup> could be used to fully capture the dynamics associated with the electric drive.

#### 6. Wheel-Ground Interaction

Figure 14 shows the Simulink model that captures the wheel/ground effects of contact, friction and rolling. The normal force between the wheel and ground is calculated based upon a penetration model. Similar to a spring/damper formula, the force applied to the wheel is proportional to the penetration depth and the penetration rate. It is implemented through an enabled subsystem that is triggered when the wheel surface falls below the ground surface. The lateral friction force is applied to account for the side loading on the wheel. It is applied in a direction aligned with the rotation axis of the wheel. The magnitude of the friction force is proportional to the normal force, and scaled by a friction coefficient. This force acts in the opposite direction to the lateral motion of the wheel. There is no steering of the wheel, and that the rolling direction is always aligned with the vehicle forward direction. The model assumes no resistance to rolling, so the force on the wheel in the rolling direction equals zero, though this could be easily elaborated to include rolling resistance if desired.



**Figure 13. The Simulink® model of the landing gear hydraulic force actuator.**

#### D. Modeling Aerodynamic Forces and Moments

The HL-20 model already included detailed environmental and aerodynamic coefficients models, from which the flight vehicle forces and moments are calculated. These existing subsystems of the HL-20 model worked directly with the new SimMechanics-based 6DoF and landing gear model.

An additional wind gust was modeled to inject a roll moment into the HL-20 close to landing. This was achieved by simply applying a moment about the x-axis of the vehicle (roll) at a specific time and duration just prior to landing.

### III. Analysis of the Landing Gear System during Landing

In this section, the modified HL-20 model is used to analyze the design of the landing gear system. We simulate two test cases, the first being the perfect, two-wheel landing where the wheel sets deployed by the two main gear touch the ground simultaneously. For the second test case, a one-wheel landing will be analyzed, where a wind gust rolls the vehicle close to touch down, such that the wheel set of the left main gear touches down first.

The second test case is a much more demanding test of the main landing gear design. During the initial part of touch down, only one wheel is carrying the weight of the plane. This subjects this landing gear linkage to a larger load with a side component that can activate retractile folding of the linkage. Computational simulation shows that the landing gear fails as it folds into its retractile position. To correct the landing gear design, it is necessary to increase the force of the landing gear actuator to sustain its deployed position against the increased side loading of a one-wheel landing. Conversely, the design engineer could build a mechanical lock in the landing gear mechanism, so that the landing gear could not retract, however the forces the lock would need to sustain would still be needed to be analyzed.

To analyze the success of the landings, a number of measurements for each landing gear set are observed. The first is a measurement of wheel height, which tracks the altitude for each of the three wheel sets. For the two-wheel landing, the two wheels of the main landing gear systems are expected to share a common Z-position. Referencing Figure 3, the next measurement expresses the angle of the vertical strut with the airframe. For a normal landing, this angle should not vary far from its reference position of 0 degrees. The next measurement expresses the compression of the shock absorber. This compression should not exceed a distance greater than a maximum value (set to 0.3 m for this analysis), but should be large enough, and gradual enough, to minimize the loads

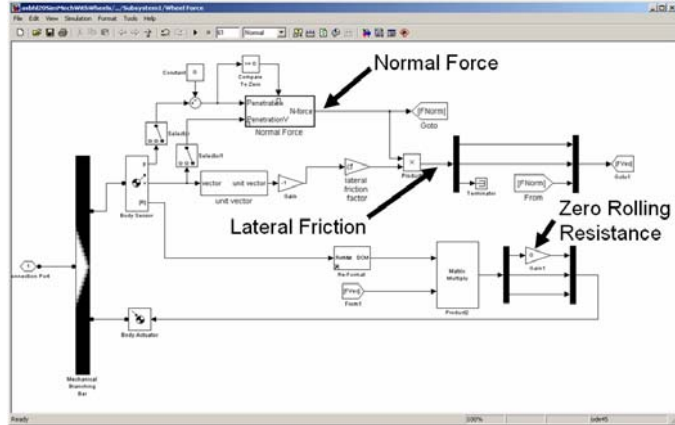


Figure 14. The Simulink® model of the wheel and ground forces on touch-down.

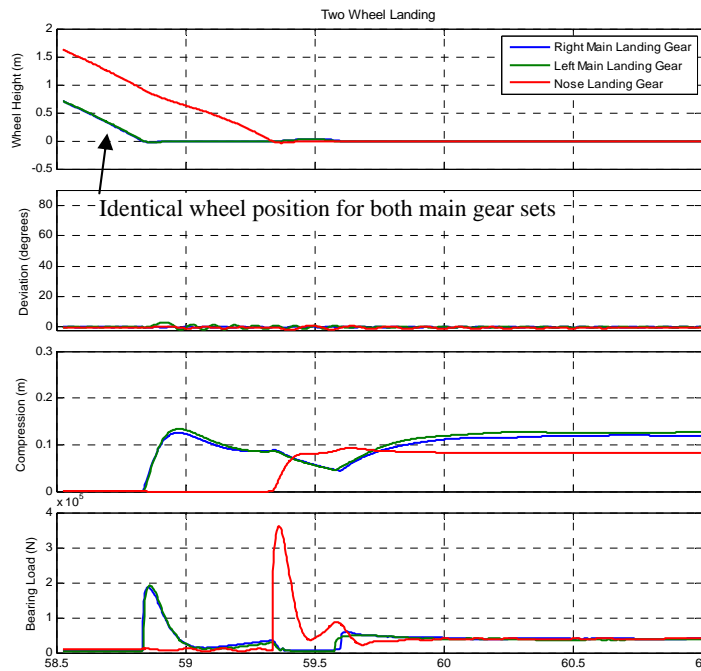


Figure 15. Results for a two-wheel normal landing.

carried in the linkage. The final measurement expresses the loads carried by the linkage. It is specifically a measurement of the bearing load of the revolute connection that attaches the vertical strut to the airframe.

### A. Two-Wheel Normal Landing

The results for the two-wheel landing are shown in Figure 15. This figure shows all four measurements discussed above.

The first measurement shows that the wheel position for the left (green curve) and right (blue curve) main gear are virtually identical, indicating simultaneous touch down between them. The kink in the curve indicates the point in time of the touch down as the wheels reach zero altitude. The red curve is for the nose gear, and shows that the nose gear wheel touches down about 1/2 second after the main gear wheels. The slight dip that each wheel makes below the surface is a result of the contact model used. The magnitude of this is about 3 centimeters, which can be thought of as representing the very physical effect of tire compression. In any event, the stiffness of the wheel and ground interaction seems sufficient for the goals of this work.

The second measurement shows the angle that the vertical strut makes with the airframe. The zero reference expresses the deployed position. The sustained value of zero that is shown in this measurement indicates that each gear maintained its deployed position, and, therefore, did not fail.

The third measurement shows successful performance of the shock absorber as its compression does not reach the hard limit of 0.3 meters. The final measurement shows a peak load on the nose-gear bearing of 350,000 N. The acceptability of this value would be left to the mechanical designers to evaluate. Because our shock absorber compression is easily satisfying the 0.3 meter requirement, a trade-off might be made to weaken the stiffness of the shock absorber to bring down the maximum bearing load. Such a trade-off analysis could be made with this model.

### B. One-Wheel Failed Landing

Figure 16 shows the resulting measurements for the failed landing simulation.

The first measurement shows that the wheel position curves are

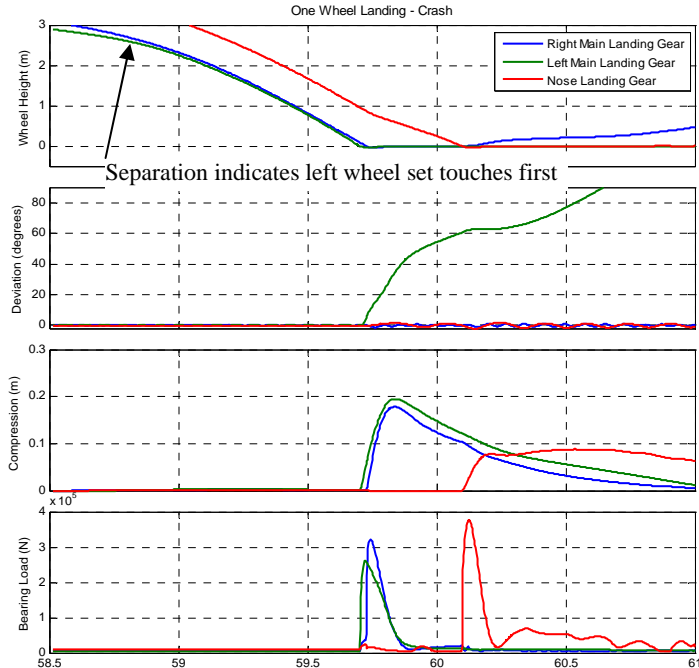


Figure 16. Results for one-wheel failed landing.

this measurement indicates that each gear maintained its deployed

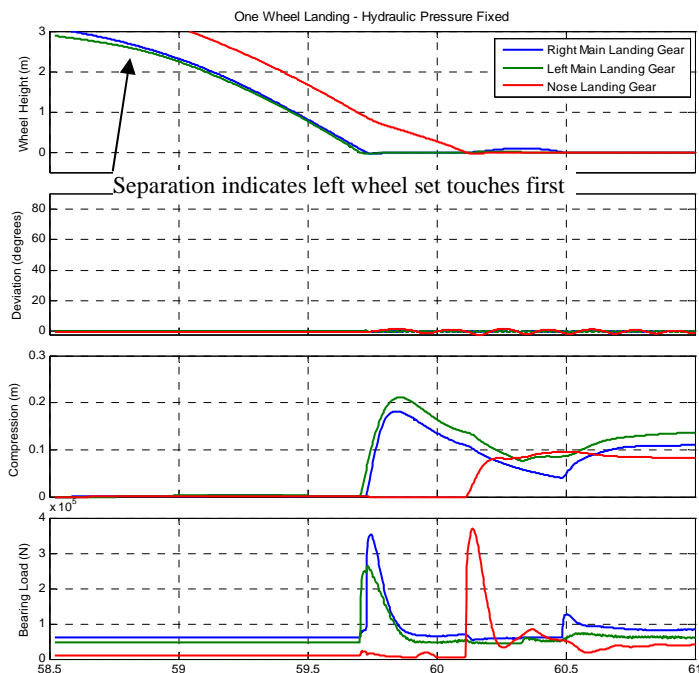


Figure 17. Results for one-wheel successful landing.

separated for the right and left main gear, indicating that the left wheel touches first. It is only a few hundredths of a second, but it will be seen that it is enough to trigger a failure of the landing gear.

The second measurement shows the failure, where the angle of the vertical strut for the left landing gear approaches a value of greater than 90 degrees. This angle expresses the collapse of the mechanism to a full retractile folding, which has been triggered by the frictional side loads felt by the wheel.

The third measurement shows greater compression for the shock absorber of the main gears. The left gear compression is about 60% greater than it was for the normal two wheel landing. This greater compression reveals itself in the bearing load measurements, too, where each main gear sees an increase in the peak bearing load that is about 40 to 60% greater than the case for the normal two wheel landing.

### C. One-Wheel Successful Landing

In order for the HL-20 to land on one landing gear, the force supplied by the landing gear hydraulic actuator needs to be increased. To test out this scenario, the actuator force was increased by a factor of 10, from 10,000 N to 100,000 N. Figure 17 shows the results for this test case.

Similar to the previous test case of the One-Wheel Failed Landing, the first measurement indicates that the left wheel touches first.

The second measurement shows the success increasing landing gear actuator force. This is shown by the vertical strut for each gear system maintaining its deployed position of a near-zero angle measurement.

As was shown for the One-Wheel Failed Landing, the successful landing sees greater compression of the shock absorber, and higher bearing loads, too. This indicates that the one-wheel landing should be the design case for choosing the spring and damping parameters of the shock absorber.

## IV. Conclusion

This paper discussed the modification of a six degree of freedom equations of motion block to include detailed domain specific models of the landing gear linkages for the HL-20 lifting body. The domain specific models enable detailed modeling of physical phenomena. The resulting model used the modified block to integrate different domain specific models and the resultant model was used to quickly search the parameter design space. To this end, a number of different landing scenarios were simulated, in particular to study the effects of landing on one wheel set because of a wind gust near landing. The analysis found that the landing gear actuator needed to exert greater force to stop the landing gear from collapsing because of the increased side loads experienced during the one wheel landing.

## References

- 
- <sup>1</sup> SimMechanics, "SimMechanics Users Guide", The MathWorks, Inc., Natick, MA, March 2006.
  - <sup>2</sup> Simulink, "Using Simulink", The MathWorks, Natick, MA, March 2006.
  - <sup>3</sup> Aerospace Blockset, "Aerospace Blockset User's Guide", The MathWorks, Inc., Natick, MA, March 2006.
  - <sup>4</sup> Simulink Parameter Estimation, "Simulink Parameter Estimation User's Guide", The MathWorks, Inc., Natick, MA, March 2006.
  - <sup>5</sup> Jackson, E. B., and C. L. Cruz, "Preliminary Subsonic Aerodynamic Model for Simulation Studies of the HL-20 Lifting Body," NASA TM-4302, August 1992.
  - <sup>6</sup> Banks, C., "A Discussion Of Methods Of Real-Time Airplane Flight Simulation", <http://www.aeroflight.com/papers/meng/index.html>
  - <sup>7</sup> Baarspul, M., "A Review of Flight Simulation Techniques," *Progress in the Aerospace Sciences*, Vol. 27, No. 1, 1990, pp. 1-120.
  - <sup>8</sup> Barnard, P., "Graphical Techniques for Aircraft Dynamic Model Development," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Providence, Rhode Island, Aug. 2004, CD-ROM.
  - <sup>9</sup> Wood, G. D., and Kennedy, D. C., "Simulating Mechanical Systems in Simulink and SimMechanics", Technical Report 91124v00, The MathWorks, Inc., Natick, MA, 2003.
  - <sup>10</sup> MATLAB, "Getting started with MATLAB" The MathWorks, Natick, MA, March 2006.
  - <sup>11</sup> SimHydraulics, "SimHydraulics User's Guide", The MathWorks, Natick, MA, March 2006.
  - <sup>12</sup> SimPowerSystems, "SimPowerSystems User's Guide", The MathWorks, Natick, MA, March 2006.