MATRIXx™

SystemBuild™ Aerospace Model Libraries

NATIONAL INSTRUMENTS™

April 2004 Edition
Part Number 370762B-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada (Calgary) 403 274 9391, Canada (Ottawa) 613 233 5949, Canada (Québec) 450 510 3055, Canada (Toronto) 905 785 0085, Canada (Vancouver) 514 685 7530, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427, India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Malaysia 603 9131 0918, Mexico 001 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 095 783 68 51, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227, Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Trademarks

AutoCode™, MATRIXx™, National Instruments™, NI™, ni.com™, SystemBuild™, and Xmath™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Conventions

The following conventions are used in this manual:

»	The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.
	This icon denotes a note, which alerts you to important information.
bold	Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.
italic	Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.
`monospace`	Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.
`monospace italic`	Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Contents

Chapter 1
Overview

Chapter 2
Environmental Library

Chapter 3
Six-Degree-of-Freedom Library

Chapter 4
Attitude Geometry Library

1 Overview

Several libraries of SystemBuild models are used extensively in the aerospace industry. They perform common tasks in aerospace simulations. These libraries contain examples, which you are encouraged to modify as required by your particular application. These models are copyrighted by National Instruments. They are provided *as is*.

These models are listed below under the chapters in which they appear:

- Chapter 2, *Environmental Library*:
  - *The Atmospheric Model*
  - *Gravity Models*
    - *Spherical Gravity Model*
    - *Fifth-Order Gravity Model*
- Chapter 3, *Six-Degree-of-Freedom Library*:
  - *Single Rigid Body Dynamics Model*
- Chapter 4, *Attitude Geometry Library*:
  - *3D Vector and Matrix Manipulations*
  - *Quaternion Manipulations*
  - *Cross-Product Attitude Determination*
  - *Three-Axis Rotations*

These models are presented as components to be incorporated in a complete simulation. They are not intended to be used as standalone models.

Unless otherwise noted, all outputs from these models use units based on meters, kilograms, and seconds (MKS). You must ensure that the inputs also are based on MKS and that all angular units are radians. Wherever applicable, signal names include the units used.

The sample outputs presented in the documentation are based on generic simulations and are not intended to present the best solution for every application.

The models in these libraries are all continuous-time models. In most applications, continuous-time modules yield greater accuracy. To generate real-time code using AutoCode, refer to the AutoCode documentation.

Loading the Aerospace Libraries

You can find the aerospace libraries in the directory, `SYSBLD/examples/aerolib_env`, where *SYSBLD* is the SystemBuild directory located in your MATRIXx distribution. The filenames and the associated model descriptions appear in Table 1-1.

Table 1-1. Aerospace Library Files

Filename	Description
`ATM.ascii`	Atmospheric model
`GRV.ascii`	Gravity model
`SRB.ascii`	Single rigid body model (SRB)
`AG.ascii`	Attitude geometry library

Since `$SYSBLD` is defined as an environmental variable inside MATRIXx, we suggest that you copy these files to your working directory or load them into SystemBuild from the Xmath command area.

To copy an aerospace library file to your current working directory:

```
copyfile "$SYSBLD/examples/aerolib_env/aero_filename";
```

To load an aerospace library file directly into SystemBuild:

```
load "$SYSBLD/examples/aerolib_env/aero_filename";
```

Reference Frame Descriptions

The most commonly used reference frames in the aerospace libraries are earth centered inertial (ECI), earth centered fixed (ECF), and vehicle body frame.

The ECI frame is defined with the origin in the center of the earth, the X vector pointing at the vernal equinox, the Z vector pointing through the north pole, and the Y vector completing the right-handed coordinate frame.

The ECF frame is defined with the origin in the center of the earth, the X vector pointing at the zero meridian through the equator, the Z vector pointing through the north pole, and the Y vector completing the right-handed coordinate frame.

Unless otherwise noted, the vehicle body frame can be any reference frame with its origin at the center of mass of the vehicle and fixed to the vehicle. The L must be used consistently. Currently, the single rigid body six-degree-of-freedom kernel is the only model that can use a body frame because it implicitly assumes that the vehicle is both rigid and single-body. A body frame is not meaningful in applications that model flexible- or multiple-body dynamics.

SuperBlock Naming Conventions

To help you incorporate these models into existing simulations, each model follows a prescribed naming convention. All SystemBuild SuperBlocks in a model begin with the same two- or three-letter prefix.

As mentioned above, signal names include a description of units wherever applicable.

2

Environmental Library

This library contains environmental models—an atmospheric model and two gravity models.

The Atmospheric Model

The atmospheric model provides atmospheric parameters to an air vehicle simulation. A SuperBlock, shown in Figure 2-1, calculates local pressure, density, temperature, and speed of sound as a function of altitude. The atmospheric model is based on the condensed U.S. Standard Atmosphere (1962).[1]

The atmospheric model calculates the local atmospheric parameters using linear interpolation blocks and published data. The published data limits the pressure, density, and temperature parameters to altitudes ranging from 0 to 700 Km. The speed of sound also is limited within altitudes ranging from 0 to 90 Km. Figure 2-1 shows the layout of this model.

The only input to this model is altitude in meters. The model outputs the local pressure in Pascals, the local density in Kg/m^3, the local temperature in degrees Kelvin, and the local speed of sound in m/s.

[1] Valley, Shea, L., *Handbook of Geophysical and Space Environments*, © 1965, McGraw Hill, Inc.

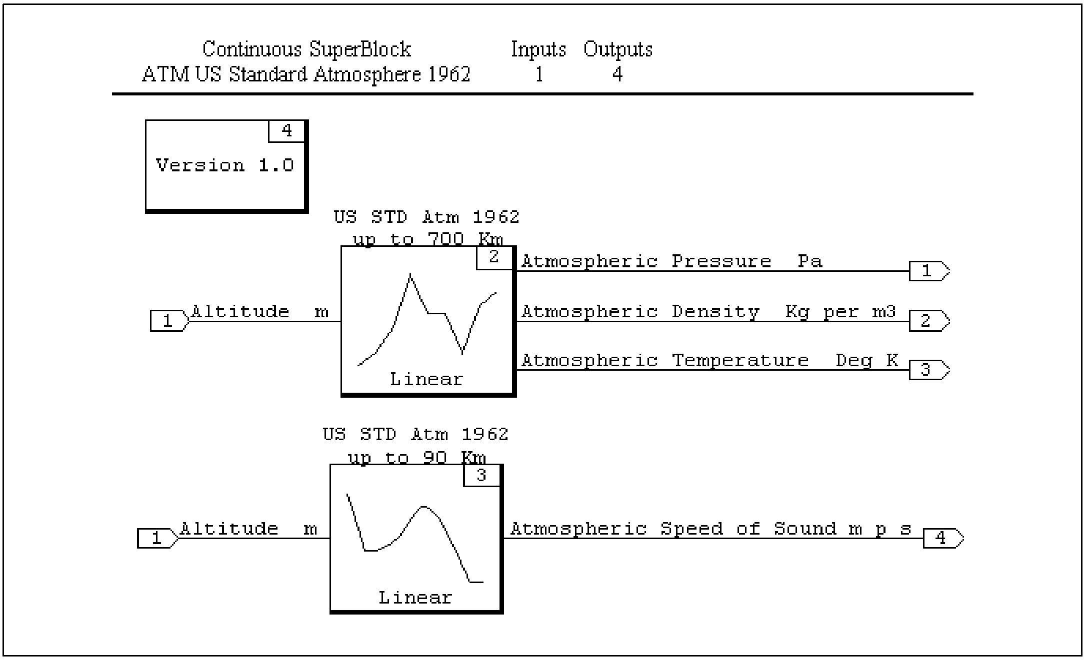

Figure 2-1. Atmospheric Model

Figures 2-2, 2-3, 2-4, and 2-5 show the atmospheric profiles generated by using input altitudes within valid ranges.

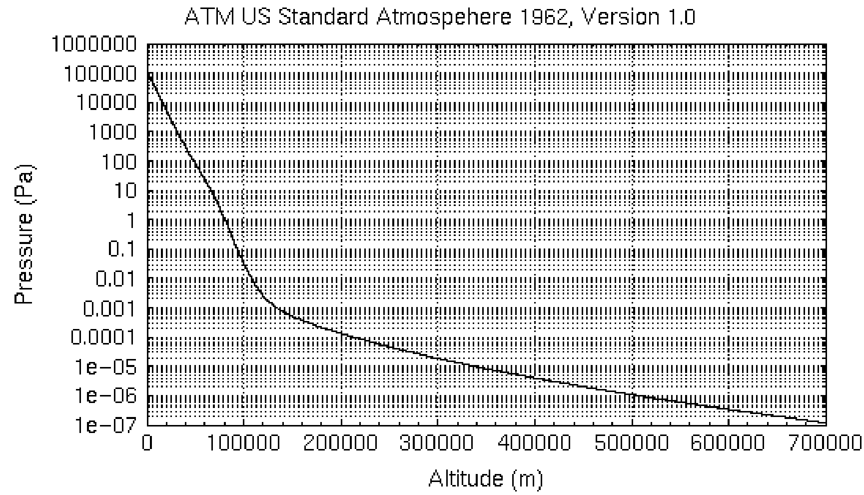

Figure 2-2. Pressure versus Altitude

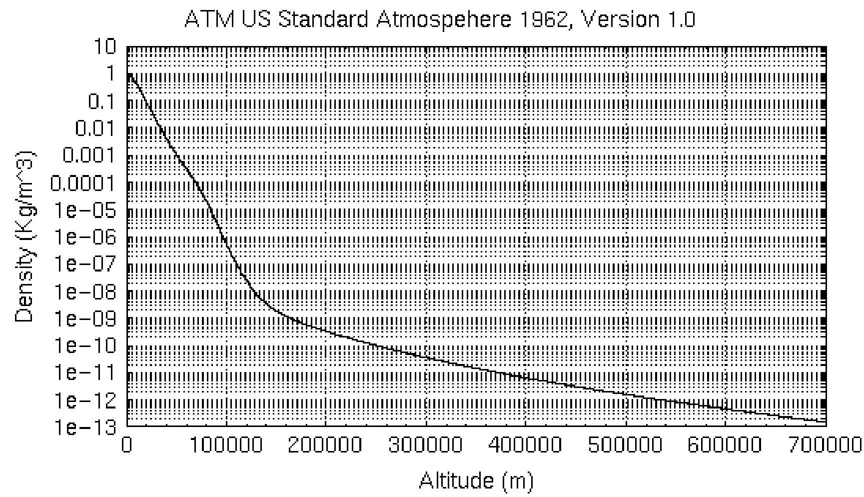

Figure 2-3. Density versus Altitude

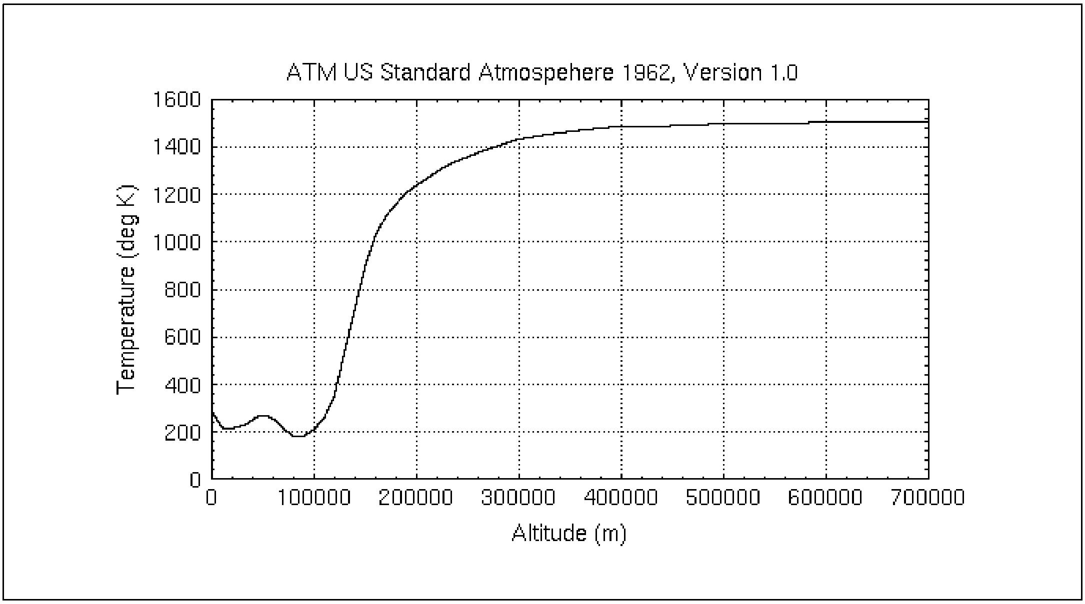

Figure 2-4. Temperature versus Altitude

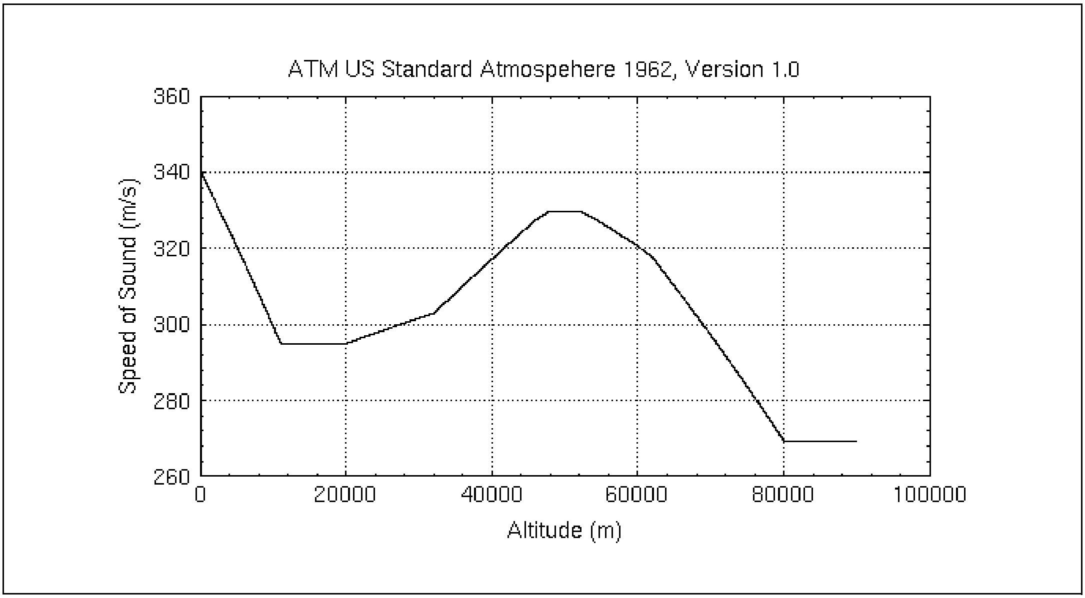

Figure 2-5. Speed of Sound versus Altitude

Gravity Models

The gravity module consists of two gravity models—a simple spherical gravity model and a fifth-order gravity model. These models provide acceleration due to gravity as a function of position and are limited to earth gravity only.

Spherical Gravity Model

The spherical gravity model implements a simple $1/r^2$ model, as shown in Figure 2-6. Parameters in this model include g, the acceleration at the surface of the earth, and r_e, the radius of the earth. The user may replace these parameters with values from other sources.

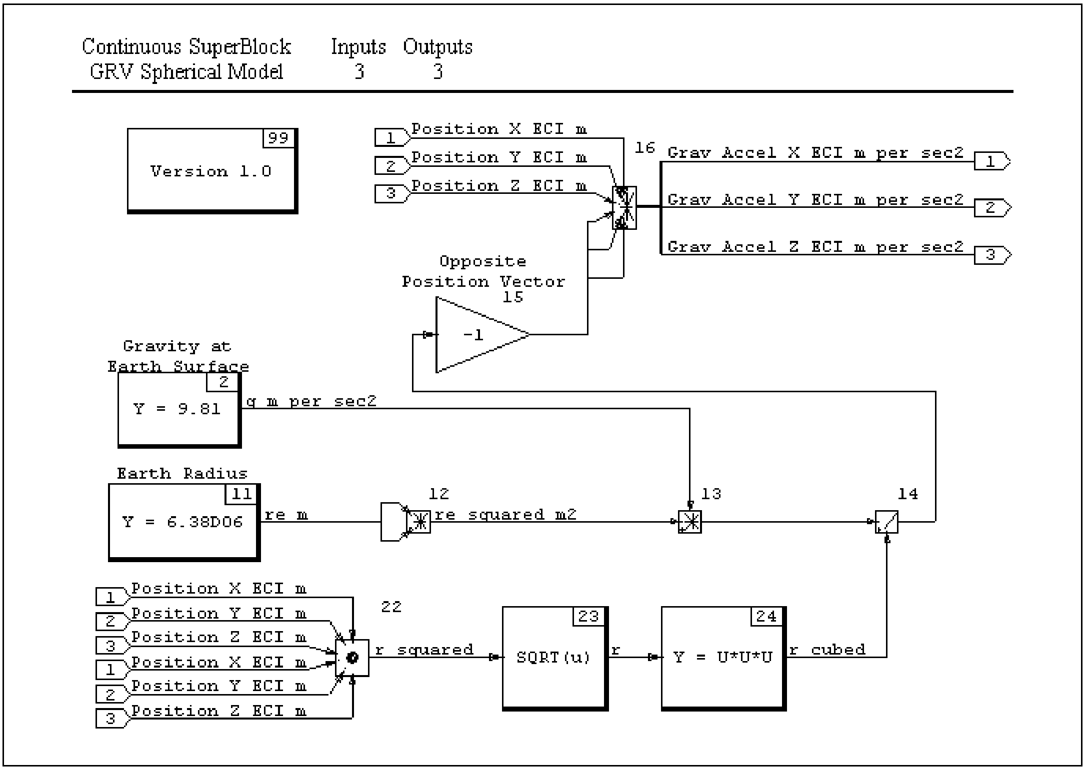

Figure 2-6. The GRV Spherical Model

Inputs

The inputs to the gravity model are, in order:

1. ECI *X* position of the vehicle, in meters
2. ECI *Y* position of the vehicle, in meters
3. ECI *Z* position of the vehicle, in meters

Outputs

The outputs from the gravity model are, in order:

1. ECI *X* component of gravity acceleration of the vehicle, in meters per second^2
2. ECI *Y* component of gravity acceleration of the vehicle,in meters per second^2
3. ECI *Z* component of gravity acceleration of the vehicle, in meters per second^2

Any earth-centered vehicle position can be provided as input to this spherically symmetric model. The output is then the vehicle gravity acceleration in the same coordinates.

Fifth-Order Gravity Model

The fifth-order gravity model shown in Figure 2-7 is based on equations from Kaplan's *Modern Spacecraft Dynamics and Control*.

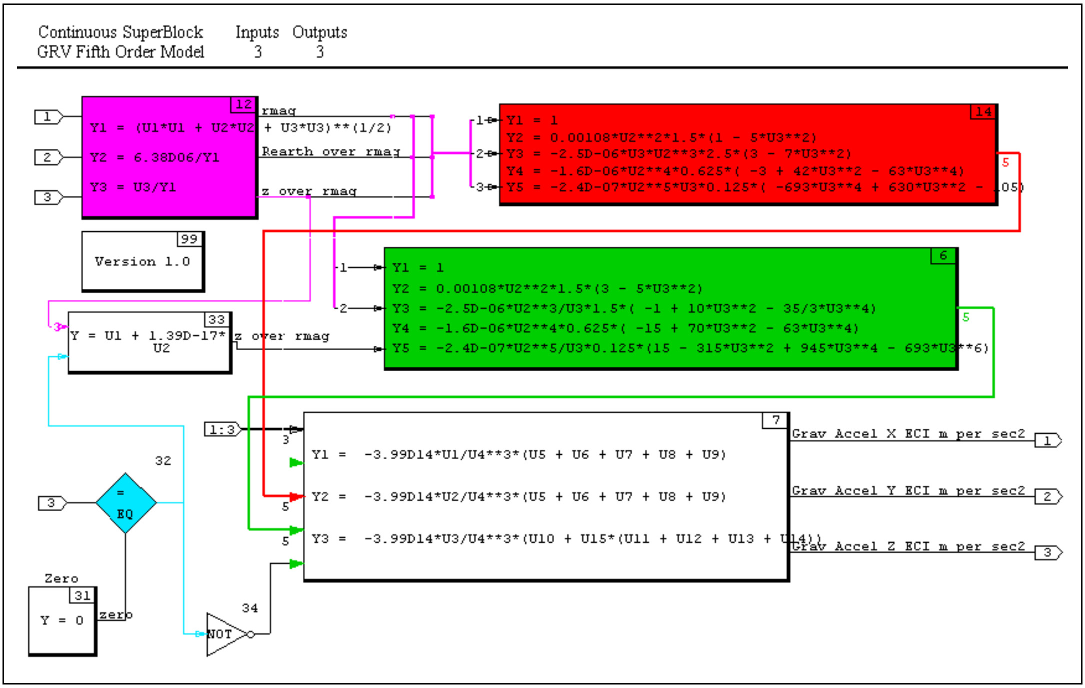

Figure 2-7. Fifth-Order Gravity Model

The equations that follow represent the potential of the earth, where μ_Θ is the earth's gravitational constant, J, contains the zonal harmonic coefficients, and P is the associated Legendre function.

$$U_\Theta r, \phi, \theta = \frac{\mu_\Theta}{r}\left[1 - \left(\sum_{k=2}^{\infty}\left(\frac{R_\Theta}{r}\right)^k J_k P_k \cos\phi + \sum_{k=2}^{\infty}\sum_{j=1}^{k}\left(\frac{R_\Theta}{r}\right)^k P_k^j \cos\phi\{C_k^j \cos j\theta + S_k^j \sin j\theta\}\right)\right]$$

Converting the above equation to the inertial frame and expanding to five terms yields the following:

$$g_{xIner}(2) = J_2\left(\frac{R_\Theta}{r}\right)^2 1.5\left(1 - 5\left(\frac{r_{zIner}}{r}\right)^2\right)$$

$$g_{xIner}(3) = J_3\left(\frac{R_\Theta}{r}\right)^3\left(\frac{r_{zIner}}{r}\right)2.5\left(3 - 7\left(\frac{r_{zIner}}{r}\right)^2\right)$$

$$g_{xIner}(4) = J_3\left(\frac{R_\Theta}{r}\right)^4 0.625\left(-3 + 42\left(\frac{r_{zIner}}{r}\right)^2 - 63\left(\frac{r_{zIner}}{r}\right)^4\right)$$

$$g_{xIner} = -G\left(\frac{r_{xIner}}{r^3}\right)(1 + g_{xIner}(2) + g_{xIner}(3) + g_{xIner}(4) + g_{xIner}(5))$$

$$g_{yIner} = -G\left(\frac{r_{yIner}}{r^3}\right)(1 + g_{yIner}(2) + g_{yIner}(3) + g_{yIner}(4) + g_{yIner}(5))$$

$$g_{xIner}(5) = J_5\left(\frac{R_\Theta}{r}\right)^5\left(\frac{r_{zIner}}{r}\right)0.125\left(-105 + 630\left(\frac{r_{zIner}}{r}\right)^2 - 693\left(\frac{r_{zIner}}{r}\right)^4\right)$$

where:

$$g_{yIner}(I) = g_{xIner}(I)$$

$$g_{zIner}(2) = J_2\left(\frac{R_\Theta}{r}\right)^2 1.5\left(3 - 5\left(\frac{r_{zIner}}{r}\right)^2\right)$$

$$g_{zIner}(3) = J_3\left(\frac{\left(\frac{R_\Theta}{r}\right)^3}{\frac{r_{zIner}}{r}}\right)1.5\left(-1 + 10\left(\frac{r_{zIner}}{r}\right)^2 - \frac{35}{3}\left(\frac{r_{zIner}}{r}\right)^4\right)$$

$$g_{zIner}(4) = J_4\left(\frac{R_\Theta}{r}\right)^4 0.625\left(-15 + 70\left(\frac{r_{zIner}}{r}\right)^2 - 63\left(\frac{r_{zIner}}{r}\right)^4\right)$$

$$g_{zIner}(5) = J_5\left(\frac{\left(\frac{R_\Theta}{r}\right)^5}{\frac{r_{zIner}}{r}}\right)0.125(15 +$$

$$\left(-315\left(\frac{r_{zIner}}{r}\right)^2\right) + 945\left(\frac{r_{zIner}}{r}\right)^4 - 693\left(\frac{r_{zIner}}{r}\right)^6\right)$$

$$g_{zIner} = -G\left(\frac{r_{zIner}}{r^3}\right)(1 + g_{zIner}(2) + g_{zIner}(3) + g_{zIner}(4) + g_{zIner}(5))$$

$J1 = 1$
$J2 = 0.0018263$
$J3 = -2.532153\text{e-}6$
$J4 = -1.610988\text{e-}6$
$J5 = -2.357857\text{e-}7$

Inputs

The inputs to the gravity model are, in order:

1. ECI *X* position of the vehicle, in meters
2. ECI *Y* position of the vehicle, in meters
3. ECI *Z* position of the vehicle, in meters

Outputs

The outputs from the gravity model are, in order:

1. ECI *X* component of gravity acceleration of the vehicle, in meters per second^2
2. ECI *Y* component of gravity acceleration of the vehicle, in meters per second^2
3. ECI *Z* component of gravity acceleration of the vehicle, in meters per second^2

The gravity model is an implementation of zonal harmonics, so the vehicle ECF position can be provided as input, and the output is then the vehicle gravity acceleration in ECF coordinates.

Figure 2-8 shows an example of the fifth-order gravity model output. The inputs are from a simple ballistic trajectory.

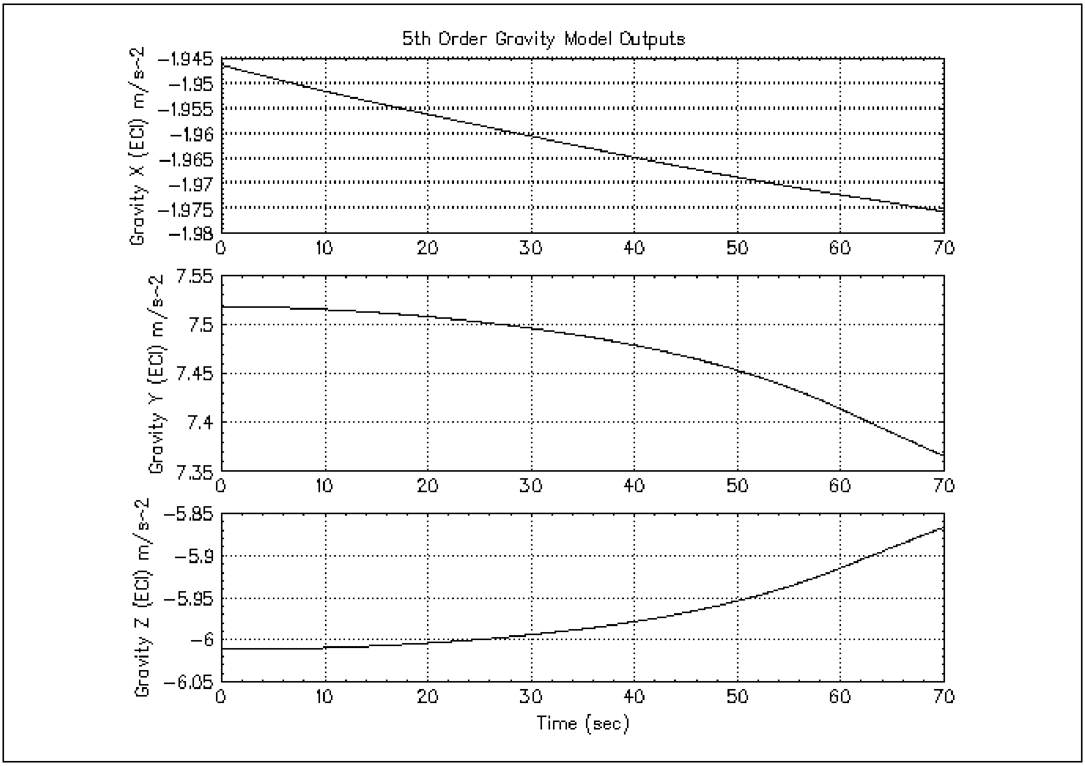

Figure 2-8. Fifth-Order Gravity Model Outputs

For the same ballistic trajectory inputs, the simple gravity model produces the outputs shown in Figure 2-9.

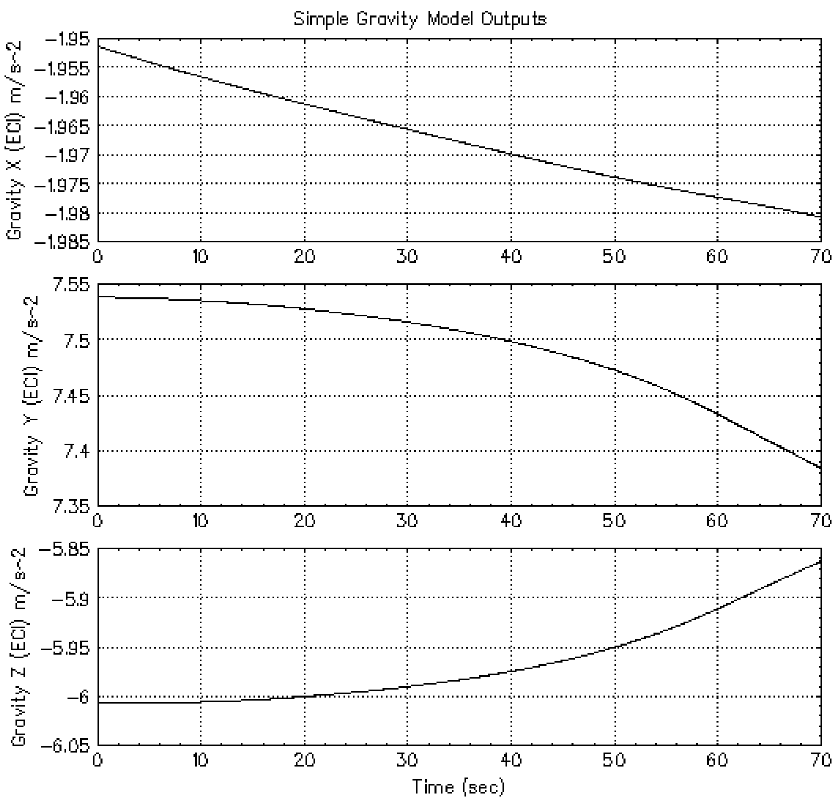

Figure 2-9. Simple Model Gravity Outputs

Usage Notes

- The fifth-order gravity model has been successfully used in several aerospace programs, although higher fidelity models exist.
- Both models contain an algebraic connection from the inputs to the outputs. You should be aware of this and avoid creating algebraic loops within your model.

3

Six-Degree-of-Freedom Library

The Six-Degree-of-Freedom (SDOF) library contains one model, the single rigid body (SRB) dynamics model.

The single rigid body model is a full six-degree-of-freedom propagator that takes forces and torques as inputs and computes the time history of position and attitude. It is limited to vehicles that can be modeled as a single rigid body and for which all forces and torques are known.

The SRB model was designed to be easily incorporated into larger models. Inputs may be connected directly to this block diagram in their most common forms (without coordinate transformations). In addition, the SRB model outputs include all of the data commonly required by simulations.

Single Rigid Body Dynamics Model

The SRB model, shown in Figure 3-1, includes all equations of motion for a single rigid body, including attitude dynamics models, attitude propagation equations, translational dynamics models, and translational kinematics models.

To incorporate the variable mass properties of aerospace vehicles due to expenditure of fuel, the vehicle mass and moments of inertia are included as inputs. Using this model you can implement variable mass properties or simply connect constant mass properties.

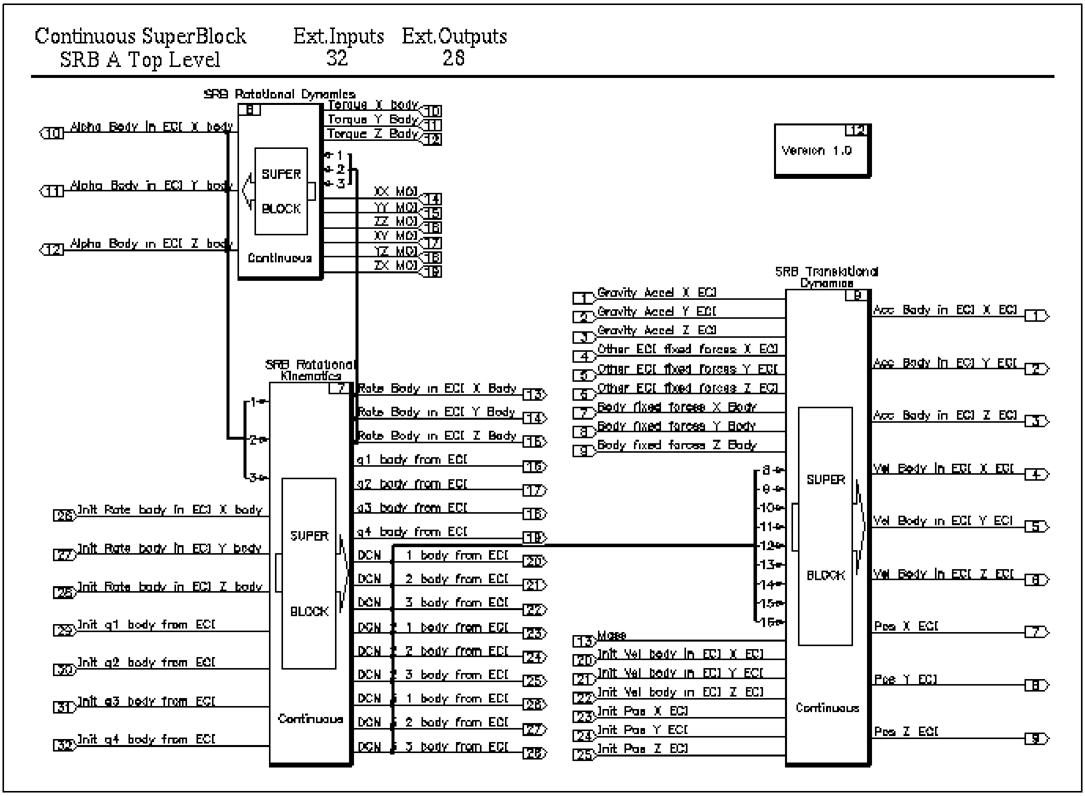

Figure 3-1. Top Level of Single Rigid Body Model

The forces in a typical aerospace vehicle model fall in one of three categories:

- Gravity
- Non-gravity forces expressed in inertial coordinates
- Non-gravity forces expressed in body coordinates

Examples of non-gravity forces expressed in inertial coordinates typically include geomagnetic forces, solar pressure, satellite drag, and others. Examples of non-gravity forces expressed in body coordinates typically include thrusters, outgassing, aircraft drag, and others. For convenience, the SRB model has a different set of inputs for each of the three categories. This eliminates the need to transform body-fixed forces to inertial, allowing you to easily implement alternative gravity models.

Most aerospace vehicle models include only torques expressed in body coordinates (excluding inertial coordinates), so only one set of torque inputs is provided, and these must be expressed in body coordinates. The SRB model solves this problem, computing the direction cosine matrix (DCM) to transform inertial coordinates to body coordinates.

The top-level SuperBlock in this module is named SRB A Top Level. All other SuperBlocks within this module are below this SuperBlock in the hierarchy. You can provide the inputs to the model by other simulation models. For example, you can hook the outputs of the gravity model described in Chapter 2, *Environmental Library*, to the first three inputs of this model.

Inputs

The inputs are, in order:

1. Gravity acceleration in the inertial *X* direction
2. Gravity acceleration in the inertial *Y* direction
3. Gravity acceleration in the inertial *Z* direction
4. The inertial *X* component of non-gravity forces, expressed in inertial coordinates
5. The inertial *Y* component of non-gravity forces, expressed in inertial coordinates
6. The inertial *Z* component of non-gravity forces, expressed in inertial coordinates
7. The body *X* component of non-gravity forces, expressed in body coordinates
8. The body *Y* component of non-gravity forces, expressed in body coordinates
9. The body *Z* component of non-gravity forces, expressed in body coordinates
10. The body *X* component of torque
11. The body *Y* component of torque
12. The body *Z* component of torque
13. Vehicle mass
14. *XX* moment of inertia of the vehicle
15. *YY* moment of inertia of the vehicle
16. *ZZ* moment of inertia of the vehicle
17. *XY* product of inertia of the vehicle
18. *YZ* product of inertia of the vehicle

19. *ZX* product of inertia of the vehicle
20. Inertial *X* component of initial velocity of the vehicle relative to inertial space
21. Inertial *Y* component of initial velocity of the vehicle relative to inertial space
22. Inertial *Z* component of initial velocity of the vehicle relative to inertial space
23. Inertial *X* component of the initial vehicle position
24. Inertial *Y* component of the initial vehicle position
25. Inertial *Z* component of the initial vehicle position
26. Body *X* component of the initial angular velocity of the vehicle in inertial space
27. Body *Y* component of the initial angular velocity of the vehicle in inertial space
28. Body *Z* component of the initial angular velocity of the vehicle in inertial space
29. First component of the initial quaternion to transform inertial coordinates to body coordinates
30. Second component of the initial quaternion to transform inertial coordinates to body coordinates
31. Third component of the initial quaternion to transform inertial coordinates to body coordinates
32. Fourth component of the initial quaternion to transform inertial coordinates to body coordinates

Notice that the gravity inputs (inputs 1, 2, and 3) are accelerations, not forces, while inputs 4 through 9 are forces, not accelerations. The gravity accelerations must be in the same coordinates as the inertially fixed forces.

The force inputs (inputs 4 through 9) must not include any gravity. Inputs 4, 5, and 6 must not include forces represented in inputs 7, 8, and 9. Inputs 7, 8, and 9 must not include forces represented in inputs 4, 5, and 6.

The fourth element of the quaternion in inputs 29 through 32 should be the cosine of half of the Euler slew angle.

Outputs

The outputs from this block diagram are:

1. Inertial *X* component of total acceleration of the vehicle relative to inertial space
2. Inertial *Y* component of total acceleration of the vehicle relative to inertial space
3. Inertial *Z* component of total acceleration of the vehicle relative to inertial space
4. Inertial *X* component of velocity of the vehicle relative to inertial space
5. Inertial *Y* component of velocity of the vehicle relative to inertial space
6. Inertial *Z* component of velocity of the vehicle relative to inertial space
7. Inertial *X* component of the vehicle position
8. Inertial *Y* component of the vehicle position
9. Inertial *Z* component of the vehicle position
10. Body *X* component of the angular acceleration of the vehicle in inertial space
11. Body *Y* component of the angular acceleration of the vehicle in inertial space
12. Body *Z* component of the angular acceleration of the vehicle in inertial space
13. Body *X* component of the angular velocity of the vehicle in inertial space
14. Body *Y* component of the angular velocity of the vehicle in inertial space
15. Body *Z* component of the angular velocity of the vehicle in inertial space
16. First component of the quaternion to transform inertial coordinates to body coordinates.
17. Second component of the quaternion to transform inertial coordinates to body coordinates
18. Third component of the quaternion to transform inertial coordinates to body coordinates
19. Fourth component of the quaternion to transform inertial coordinates to body coordinates
20. Element (1,1) of the DCM to convert inertial coordinates to body coordinates

21. Element (1,2) of the DCM to convert inertial coordinates to body coordinates
22. Element (1,3) of the DCM to convert inertial coordinates to body coordinates
23. Element (2,1) of the DCM to convert inertial coordinates to body coordinates
24. Element (2,2) of the DCM to convert inertial coordinates to body coordinates
25. Element (2,3) of the DCM to convert inertial coordinates to body coordinates
26. Element (3,1) of the DCM to convert inertial coordinates to body coordinates
27. Element (3,2) of the DCM to convert inertial coordinates to body coordinates
28. Element (3,3) of the DCM to convert inertial coordinates to body coordinates.

The fourth element of the quaternion in outputs 16 through 19 is the cosine of half of the Euler slew angle.

Examples

Figure 3-2 shows an example of the position outputs of this model when run with inputs for a ballistic trajectory. Figure 3-3 shows the corresponding velocity outputs.

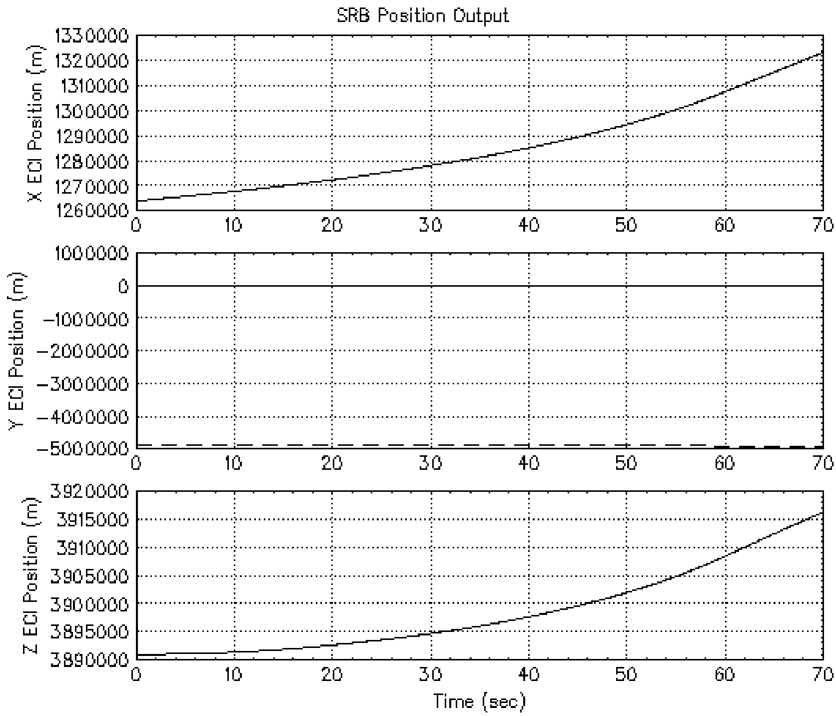

Figure 3-2. Inputs for Ballistic Trajectories

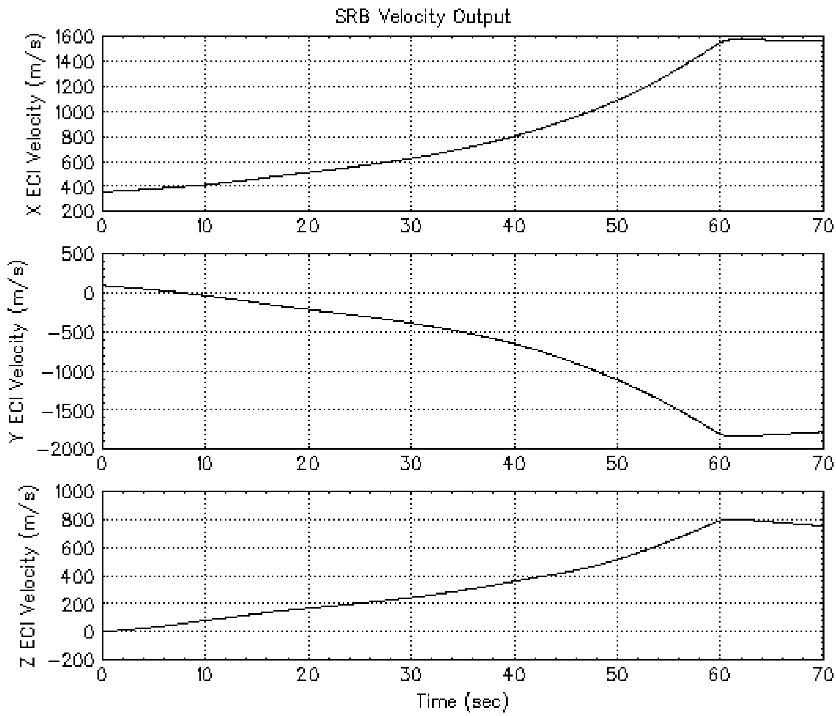

Figure 3-3. Velocity Output

Usage Notes

- The SRB model is non-dimensional in the sense that if you provide the inputs in consistent base units, the outputs are in the same base units. The only exception to this is that all inputs involving angles must be expressed in units derived from radians, and all outputs are expressed in units derived from radians.
- No assumptions are made about the definition of the body coordinates, so you may employ any convenient orthogonal right-handed system fixed in the vehicle. Notice, however, that the same body coordinate system must be used consistently in all inputs and outputs.
- No assumptions are made regarding the definition of the inertial frame or the inertial coordinates, so you are free to use any frame that can be approximated as inertial with sufficient accuracy for the project at hand.

- All SRB model SuperBlock names begin with `SRB`. To avoid naming conflicts, do not prefix your own SuperBlock names with `SRB`. This naming convention has the added benefit of ordering all SRB SuperBlocks consecutively in the SystemBuild Catalog alphabetical listing.
- This model runs under the SystemBuild environment for matrix and complex manipulations, and the symbol "I" is commonly used to denote either the identity matrix or sqrt (–1). As a result, there is no convenient symbol to denote the inertial reference frame used in this module.

  Signal names in SRB block diagrams use the letter "N" to denote the inertial reference frame. This notation is taken from Kane and Levinson.[1]
- The equations implemented in the SRB model are mathematically exact. However, you must keep the following limitations in mind:
  - No vehicle is a truly rigid body. Most aerospace vehicles undergo structural and thermal flexing and have fuel that sloshes. Most aerospace vehicles are not truly single bodies because they contain gyros, valves, control surfaces, or other small machinery.
  - Any numerical integration method might introduce noticeable errors when an unstable or neutrally stable vehicle has no closed loop control and the simulation covers an extended period of time. An example of this would be orbit propagation covering several months with no orbit adjust maneuvers.
  - Any simulation can be no more accurate than its inputs. All forces and torques fed into the SRB model are only mathematical approximations and will have errors. Similarly, all mass properties and initial conditions will be imperfect.
- You must ensure that the initial velocity, initial position, initial angular rate, and initial quaternion are constant during the simulation.
- The integrators in the SuperBlocks SRB Rotational Kinematics, SRB Translational Kinematics, and SRB Quaternion Propagator each feed into a sum block to inject initial conditions. These initial conditions are supplied as inputs to the respective SuperBlock, which are passed through the hierarchy from the top level SuperBlock. As an alternative, you can supply these initial conditions with an algebraic expression block within the SuperBlock containing the integrator. This eliminates the need for the inputs to supply the initial conditions throughout the hierarchy.

---

[1] Kane, Thomas R. & Levinson, David A., *Dynamics, Theory and Applications*, © 1985 McGraw Hill, Inc.

The quaternion calculation provides an example of how to supply initial conditions using an algebraic expression block. In the SRB Quaternion Propagator SuperBlock, you can add an algebraic expression block with four outputs. This block would simply assign values to each of the four outputs. The values would feed into the first four inputs of the sum block, replacing the existing inputs.

All blocks with initial conditions as inputs have the initial conditions as their last inputs. This facilitates eliminating them if you decide to use an algebraic expression block. The blocks in the SRB model are SRB Quaternion Propagator, SRB Rotational Kinematics, and SRB A Top Level.

If you decide to model the vehicle with constant mass properties, you can replace the mass properties inputs in the appropriate SuperBlocks with an algebraic block in the low-level SuperBlock. The procedure for doing this is similar to the procedure for eliminating initial conditions.

- There is an algebraic connection from the input forces to the output accelerations. Therefore, be aware that if the SRB model is used in a larger model, there should be no algebraic connection from the output accelerations to the input forces. Otherwise, an algebraic loop results.
- Similarly, there is an algebraic connection from the input torques to the output angular acceleration. To prevent an algebraic loop, there should be no algebraic connection from the output angular acceleration to the input torques.

4

Attitude Geometry Library

The Attitude Geometry library is a series of SuperBlocks that are implementations of equations commonly used in attitude dynamics. Included are 3D vector and matrix manipulations, quaternion manipulations, basic attitude determination equations, and three-axis rotation equations.

These SuperBlocks were designed to be easily incorporated into larger models. The inputs to these blocks can be supplied by the larger simulation, and they are in commonly used forms. Similarly, the outputs can be easily used elsewhere in the larger model because they are in commonly used forms.

The Attitude Geometry Library contains several SuperBlocks that are useful to an attitude control engineer. These blocks are intended to be used individually. You can select any that are useful for the problem at hand and ignore the others.

All of the SuperBlocks within this library have names which begin with `AG`. You can avoid naming conflicts by avoiding SuperBlock names beginning with `AG`. All of the SuperBlocks that are specific to the cross-product attitude determination function begin with `AG CPAD`.

Most of these SuperBlocks are self-contained. They do not incorporate other SuperBlocks within them. The only exception to this is the AG CPAD Cross Product Attitude Determination SuperBlock, which includes the AG CPAD Ref to Temp SuperBlock and other blocks not specific to this function.

3D Vector and Matrix Manipulations

There are several SuperBlocks included for manipulation of 3D vectors and transformation matrices. Each of these SuperBlocks is standalone—that is, it does not contain any other SuperBlocks.

Each of these SuperBlocks is a fairly simple block diagram. However, these basic functions are used so frequently that it is useful to have them readily available.

Normalizing a Vector

SuperBlock AG Normalize 3 Vector creates a unit vector parallel to and in the same direction as its input vector by dividing the input vector by its magnitude. The inputs to the SuperBlock are, in order:

1. *X* component of vector
2. *Y* component of vector
3. *Z* component of vector

Its outputs are, in order:

1. *X* component of unit vector
2. *Y* component of unit vector
3. *Z* component of unit vector

The coordinates of the output vector are the same as the coordinates of the input vector.

If the input is the zero vector, then the associated unit vector is undefined. In this case, the output is also a zero vector.

Transposing a 3 × 3 Matrix

SuperBlock AG Transpose 3 × 3 Matrix transposes a 3 × 3 matrix. Its inputs are, in order:

1. Element 1,1 of the matrix to be transposed
2. Element 1,2 of the matrix to be transposed
3. Element 1,3 of the matrix to be transposed
4. Element 2,1 of the matrix to be transposed
5. Element 2,2 of the matrix to be transposed
6. Element 2,3 of the matrix to be transposed
7. Element 3,1 of the matrix to be transposed
8. Element 3,2 of the matrix to be transposed
9. Element 3,3 of the matrix to be transposed

Its outputs are, in order:

1. Element 1,1 of the transposed matrix
2. Element 1,2 of the transposed matrix
3. Element 1,3 of the transposed matrix
4. Element 2,1 of the transposed matrix

5. Element 2,2 of the transposed matrix
6. Element 2,3 of the transposed matrix
7. Element 3,1 of the transposed matrix
8. Element 3,2 of the transposed matrix
9. Element 3,3 of the transposed matrix

Inverting a 3 × 3 Matrix

SuperBlock AG Invert 3 × 3 matrix inverts a 3 × 3 matrix. Its inputs are, in order:

1. Element 1,1 of the matrix to be inverted
2. Element 1,2 of the matrix to be inverted
3. Element 1,3 of the matrix to be inverted
4. Element 2,1 of the matrix to be inverted
5. Element 2,2 of the matrix to be inverted
6. Element 2,3 of the matrix to be inverted
7. Element 3,1 of the matrix to be inverted
8. Element 3,2 of the matrix to be inverted
9. Element 3,3 of the matrix to be inverted

Its outputs are, in order:

1. Element 1,1 of the inverted matrix
2. Element 1,2 of the inverted matrix
3. Element 1,3 of the inverted matrix
4. Element 2,1of the inverted matrix
5. Element 2,2 of the inverted matrix
6. Element 2,3 of the inverted matrix
7. Element 3,1 of the inverted matrix
8. Element 3,2 of the inverted matrix
9. Element 3,3 of the inverted matrix

If the input matrix is singular, this SuperBlock fails.

Multiplying a 3 × 3 Matrix by a Vector

SuperBlock AG Matrix Vector Multiply 3 × 3 multiplies a 3 × 3 matrix by a vector. Its inputs are, in order:

1. Element 1,1 of the pre-multiplying matrix
2. Element 1,2 of the pre-multiplying matrix
3. Element 1,3 of the pre-multiplying matrix
4. Element 2,1 of the pre-multiplying matrix
5. Element 2,2 of the pre-multiplying matrix
6. Element 2,3 of the pre-multiplying matrix
7. Element 3,1 of the pre-multiplying matrix
8. Element 3,2 of the pre-multiplying matrix
9. Element 3,3 of the pre-multiplying matrix
10. Element 1 of the post-multiplying vector
11. Element 2 of the post-multiplying vector
12. Element 3 of the post-multiplying vector

The outputs are, in order:

1. Element 1 of the product
2. Element 2 of the product
3. Element 3 of the product

Multiplying Two 3 × 3 Matrices

SuperBlock AG Multiply 3 × 3 Matrices multiplies two 3 × 3 matrices. The inputs are, in order:

1. Element 1,1 of the pre-multiplying matrix
2. Element 1,2 of the pre-multiplying matrix
3. Element 1,3 of the pre-multiplying matrix
4. Element 2,1 of the pre-multiplying matrix
5. Element 2,2 of the pre-multiplying matrix
6. Element 2,3 of the pre-multiplying matrix
7. Element 3,1 of the pre-multiplying matrix
8. Element 3,2 of the pre-multiplying matrix
9. Element 3,3 of the pre-multiplying matrix
10. Element 1,1 of the post-multiplying matrix

11. Element 1,2 of the post-multiplying matrix
12. Element 1,3 of the post-multiplying matrix
13. Element 2,1 of the post-multiplying matrix
14. Element 2,2 of the post-multiplying matrix
15. Element 2,3 of the post-multiplying matrix
16. Element 3,1 of the post-multiplying matrix
17. Element 3,2 of the post-multiplying matrix
18. Element 3,3 of the post-multiplying matrix

The outputs are, in order:

1. Element 1,1 of the product
2. Element 1,2 of the product
3. Element 1,3 of the product
4. Element 2,1 of the product
5. Element 2,2 of the product
6. Element 2,3 of the product
7. Element 3,1 of the product
8. Element 3,2 of the product
9. Element 3,3 of the product

Quaternion Manipulations

This library includes manipulations of Euler symmetric parameters, more commonly called *quaternions*.

All quaternions used in these blocks are defined to have the fourth element equal to one-half of the cosine of the Euler rotation angle.

Quaternion Multiplication

Two successive arbitrary rotations can be expressed as one single rotation. Mathematically combining two quaternions, representing two successive rotations, to represent the final single rotation is commonly referred to as *quaternion multiplication*. The SuperBlock AG Quaternion Multiplier performs this operation.

The inputs are, in order:

1. Element 1 of the quaternion representing the first rotation
2. Element 2 of the quaternion representing the first rotation
3. Element 3 of the quaternion representing the first rotation
4. Element 4 of the quaternion representing the first rotation
5. Element 1 of the quaternion representing the second rotation
6. Element 2 of the quaternion representing the second rotation
7. Element 3 of the quaternion representing the second rotation
8. Element 4 of the quaternion representing the second rotation

The outputs are, in order:

1. Element 1 of the quaternion representing the resultant rotation
2. Element 2 of the quaternion representing the resultant rotation
3. Element 3 of the quaternion representing the resultant rotation
4. Element 4 of the quaternion representing the resultant rotation

Converting a Quaternion to DCM

The SuperBlock AG Quaternion to DCM converts a quaternion to its equivalent direction cosine matrix (DCM). The inputs are, in order:

1. Element 1 of the quaternion
2. Element 2 of the quaternion
3. Element 3 of the quaternion
4. Element 4 of the quaternion

The outputs are, in order:

1. Element 1,1 of the equivalent DCM
2. Element 1,2 of the equivalent DCM
3. Element 1,3 of the equivalent DCM
4. Element 2,1 of the equivalent DCM
5. Element 2,2 of the equivalent DCM
6. Element 2,3 of the equivalent DCM
7. Element 3,1 of the equivalent DCM
8. Element 3,2 of the equivalent DCM
9. Element 3,3 of the equivalent DCM

Converting DCM to a Quaternion

SuperBlock AG DCM to Quaternion converts a DCM to its equivalent quaternion. The inputs are, in order:

1. Element 1,1 of the DCM
2. Element 1,2 of the DCM
3. Element 1,3 of the DCM
4. Element 2,1 of the DCM
5. Element 2,2 of the DCM
6. Element 2,3 of the DCM
7. Element 3,1 of the DCM
8. Element 3,2 of the DCM
9. Element 3,3 of the DCM

The outputs are, in order:

1. Element 1 of the equivalent quaternion
2. Element 2 of the equivalent quaternion
3. Element 3 of the equivalent quaternion
4. Element 4 of the equivalent quaternion

Cross-Product Attitude Determination

Frequently, attitude determination instruments provide a vector to a known reference point. Examples of this include star sensors and sun sensors. Two such simultaneous measurements provide sufficient information to completely determine the attitude of an aerospace vehicle. The algebraic process by which this is done is commonly known as *cross-product attitude determination*.

Cross-product attitude determination is an algebraic process in which two separate vectors, each expressed in two different coordinate reference frames, are combined to find the relative orientation of the two coordinate reference frames. Call the two vectors V1 and V2, and call the two coordinate reference frames A and B.

The SuperBlock AG CPAD CrossProduct Att Det contains the algorithms necessary to compute the attitude, given two reference vectors. The inputs are, in order:

1. The *X* component of vector V1 expressed in frame A
2. The *Y* component of vector V1 expressed in frame A
3. The *Z* component of vector V1 expressed in frame A
4. The *X* component of vector V1 expressed in frame B
5. The *Y* component of vector V1 expressed in frame B
6. The *Z* component of vector V1 expressed in frame B
7. The *X* component of vector V2 expressed in frame A
8. The *Y* component of vector V2 expressed in frame A
9. The *Z* component of vector V2 expressed in frame A
10. The *X* component of vector V2 expressed in frame B
11. The *Y* component of vector V2 expressed in frame B
12. The *Z* component of vector V2 expressed in frame B

The outputs are, in order:

1. Element 1,1 of the coordinate transformation from frame B to frame A
2. Element 1,2 of the coordinate transformation from frame B to frame A
3. Element 1,3 of the coordinate transformation from frame B to frame A
4. Element 2,1 of the coordinate transformation from frame B to frame A
5. Element 2,2 of the coordinate transformation from frame B to frame A
6. Element 2,3 of the coordinate transformation from frame B to frame A
7. Element 3,1 of the coordinate transformation from frame B to frame A
8. Element 3,2 of the coordinate transformation from frame B to frame A
9. Element 3,3 of the coordinate transformation from frame B to frame A

Note This method of attitude determination is sub-optimal in the sense that it uses only an instantaneous measurement and ignores previous and future measurements.

This method of attitude determination assumes there is no noise on the measurements. If there is noise, this method selects a coordinate transformation that forces vector V1 to be perfectly aligned between the two reference frames; V2 might not be perfectly aligned. The selected coordinates transformation forces V2 expressed in the A frame, V2 expressed in the B frame, and V1 to be co-planar.

Three-Axis Rotations

The Attitude Geometry Library includes 12 SuperBlocks representing the 12 possible three-axis rotations. The inputs to each of these SuperBlocks are the three rotation angles, and the outputs are the elements of the DCM representing the equivalent rotation.

The AG Space 3 1 2 to DCM SuperBlock represents three consecutive rotations. Assume two reference frames called A and B. These two frames are initially aligned relative to each other. Then B performs three rotations relative to A:

1. B rotates about the A frame Z-axis. The Z-axis is chosen because this is the first rotation of the 3-1-2 series.
2. B rotates about the A frame X-axis. The X-axis is chosen because this is the second rotation of the 3-1-2 series.
3. B rotates about the A frame Y-axis. The Y-axis is chosen because this is the third rotation of the 3-1-2 series.

The output of the SuperBlock is the DCM representing the final orientation of B relative to A.

The other 11 SuperBlocks have names that reflect the order of rotations in an analogous fashion. These 12 SuperBlocks represent all possible combinations of rotations.

Note These SuperBlocks are unlike the Three-Axis Rotation block in the SystemBuild palette. This block represents consecutive rotations about the B frame axes and not about the A frame axes.

The rotations represented by these SuperBlocks are called *space* rotations, and the rotations represented by the Three-Axis Rotation block in the SystemBuild palette are called *body* rotations.[1]

To form a block analogous to the Three-Axis Inverse block in the SystemBuild pallette, use the appropriate SuperBlock from this library, and feed its outputs through the AG Transpose 3 × 3 Matrix SuperBlock, also in this library.

[1] This terminology is borrowed from Kane, Thomas R. & Levinson, David A., *Dynamics, Theory and Applications*, © 1985 McGraw Hill, Inc.

A

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at `ni.com` for technical support and professional services:

- **Support**—Online technical support resources include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at `ni.com/support`. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting `ni.com/support`. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training and Certification**—Visit `ni.com/training` for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit `ni.com/alliance`.

If you searched `ni.com` and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of `ni.com/niglobal` to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Index

P

R

S

T

W