

*VisSim Tutorial Series*

**Introduction to  
6-DOF Simulation of  
Air Vehicles**

*Robert Josselson, PhD,  
ITT Aerospace Systems Group*

## VisSim Tutorial Series

### Introduction to 6-DOF Simulation of Air Vehicles

---

**Copyright** ©1997 Visual Solutions, Inc.

All rights reserved.

**Trademarks** VisSim is a trademark of Visual Solutions.

Excerpted with permission from *Modeling and Visual Simulation in Industry*, A. Mulpur and P. Darnell, International Thomson Computer Press, Boston, MA, 1997.

The information in this document is subject to change without notice and does not represent a commitment by Visual Solutions. Visual Solutions does not assume responsibility for errors that may appear in this document.

Other books in the VisSim Tutorial Series include:

- *Biomedical Systems: Modeling and Simulation of Lung Mechanics and Ventilator Controls Design*. Mike Borrello, Metran America, Inc.
- *Fundamentals of Mathematical Modeling and Simulation*. Peter Darnell and Arun Mulpur, Visual Solutions, Inc.
- *Heating, Ventilation and Air Conditioning (HVAC) Controls: Variable Air Volume (VAV) Systems*. Nebil Ben-Aissa, Johnson Controls, Inc.
- *Simulation of Communication Systems*. Eugene Estinto, Eritek, Inc.
- *Simulation of Motion Control Systems*. William Erickson, Indramat-Rexroth.

## **Table of Contents**

<b>Introduction .....</b>	<b>1</b>
<b>Coordinate System and Transformations .....</b>	<b>2</b>
<b>Derivation of the Equations of Motion .....</b>	<b>5</b>
<b>Keeping Track of the True Body Attitude .....</b>	<b>6</b>
Euler Angle Rates .....	6
<b>Aerodynamic Forces and Moments .....</b>	<b>8</b>
Aerodynamic Coefficients .....	9
Gravity .....	10
Angles of Attack and Side Slip .....	10
Sensed Accelerations .....	10
Local-Level Velocity and Position .....	11
Environmental Model .....	11
Graphical Simulation .....	12
Simulation Organization .....	12
6-DOF Simulation Implementation in VisSim .....	12
Simulation Example .....	13
Simulation Enhancements .....	18
Monte Carlo Simulations .....	19
<b>Summary .....</b>	<b>20</b>
<b>References .....</b>	<b>20</b>



## Introduction

A 6-DOF simulation has 6 degrees of freedom: three rotations and three translations. Since a vehicle both translates and rotates through the air, two coordinate systems keep track of the vehicle. The first is a known or fixed reference system, referred to here as a local-level coordinate system. The local-level system is a standard, right-hand, three-axis coordinate system with the x-y plane representing the horizontal ground plane; the z-axis is positive down. This is a fixed system that does not translate or rotate. It is commonly thought of as fixed to a flat, non-rotating earth for short duration flights.

The second coordinate system is fixed to the body of the simulated vehicle, referred to here as the body-fixed coordinate system. The x-axis is positive out the nose of the vehicle; the y-axis is positive out the right hand side looking forward; and the z-axis is positive looking down. The origin of the body-fixed coordinate system is located at the center of gravity of the vehicle. Its position is known with respect to the origin of the local-level coordinate system. The attitude of the vehicle is obtained relative to the local-level coordinate system orientation through three rotations known as Euler angles. Direction cosines (or quaternions) keep track of the Euler angles (or attitude) of the vehicle (the attitude of the body-fixed coordinate system) relative to the local-level coordinate system.

Figure 1 shows a top level block diagram of a 6-DOF simulation as implemented in VisSim. The inputs to the simulation and the outputs of the simulation are at the top left and right of the diagram. Typical inputs are the initial position and attitude of the vehicle, its initial speed, its initial angular rates, and its mass properties. Typical outputs are time history plots of parameters important to the designer, such as body attitude and rates, trajectory profiles, angles of attack and sideslip, and speed.

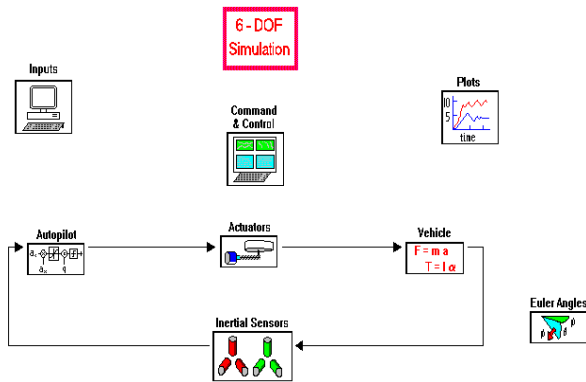


Figure 1. Top level representation of a 6-DOF simulation

The Command and Control compound block contains nominal flight commands for the autopilot, such as flight altitude and attitude, and maneuver commands, which may be heading, turn rates, or accelerations.

The functionality of the block in the lower right hand corner processes the true attitude of the vehicle through propagation of quaternions from body angular rates. It also generates direction cosines and Euler angles from the quaternions.

The loop at the center contains the equations of motion and models specific to the vehicle under simulation. The Vehicle compound block holds the equations of motion and the aerodynamic coefficients for the vehicle. The Autopilot compound block contains the control laws that stabilize the vehicle and allow it to maneuver. The Autopilot uses data from the on-board inertial sensors, usually rate gyros and accelerometers for feedback. The dynamics of the inertial instruments are modeled in the Inertial Sensors compound block. The outputs of the Autopilot are typically fin attitude commands to four fin actuators. The two horizontal tail surfaces, or fins, deflect together (either up or down) to effect a pitch maneuver. The two vertical fins also deflect together (either left or right) to effect a yaw, or heading, maneuver. All four fins deflect differentially, or splay, to maintain roll control. The actuator hardware is modeled in the Actuators compound block.

This tutorial discusses the necessary elements that constitute a basic 6-DOF simulation, derives both the equations of motion and the propagation of true attitude and position, and depicts its implementation in VisSim.

## Coordinate System and Transformations

The equations of motion for the 6-DOF trajectory are formulated in the body-fixed frame and transformed to the local-level (that is, earth-fixed), right-hand, orthogonal frame. Figure 2 shows the two coordinate systems. A local-level coordinate system is established at the start of simulation; it does not translate; it does not rotate. The location of the simulated body is referenced to this system. The attitude of the simulated body is also referenced to this local-level system.

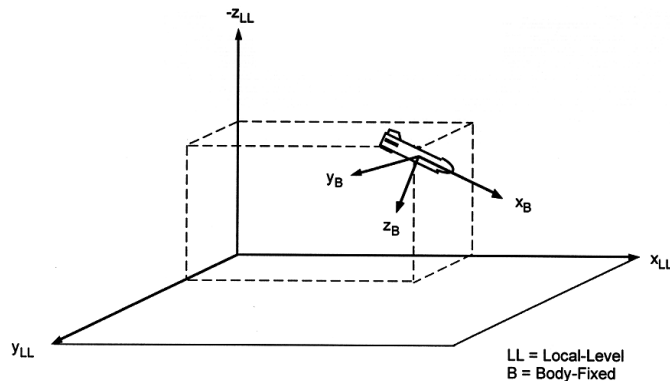


Figure 2. Two coordinate systems

Knowledge of the orientation of the body with respect to a local-level coordinate system is an essential piece of information in the simulation; it also allows vectors to be transformed from the local-level coordinate system to the body-fixed coordinate system and vice versa. The relationships between the local-level and the body-fixed coordinate systems are examined next.

Figure 3 shows the local-level and the body-fixed coordinate systems. The body-fixed coordinate system orientation is related to the local-level coordinate system through three successive rotations depicted in the figure. The process starts with a rotation through the angle  $\psi$  about the local-level z-axis, continues with a rotation through the angle  $\theta$  about the resulting intermediate y-axis, and ends with a rotation through the angle  $\phi$  about the body x-axis.

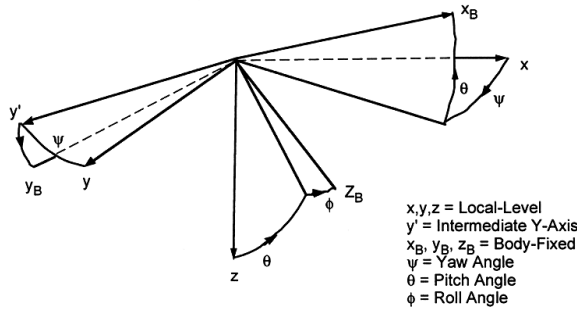


Figure 3. Relationship of local-level and body-fixed coordinate systems

The following are the three individual transformations that rotate a vector in a plane about a single axis from a starting or reference coordinate system to a second or destination coordinate system.

The first transformation represents a rotation about an x-axis through a roll angle  $\phi$ . The second represents a rotation about a y-axis through a pitch angle  $\theta$ . The third represents a rotation about a z-axis through a yaw angle  $\psi$ .

$$T_1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix}$$

$$T_2(\theta) = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix}$$

$$T_3(\psi) = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where s and c represent sine and cosine.

The following are three individual transformations that rotate a vector about a single axis from a second coordinate system back to a reference coordinate system.

$$T_3(-\psi) = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_2(-\theta) = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}$$

$$T_1(-\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix}$$

It is useful to note that for  $\alpha = \phi, \theta,$  or  $\psi$

$$T(-\alpha) = T(\alpha)^T$$

and

$$T(\alpha)^{-1} = T(\alpha)^T$$

where  $( )^T$  represents the matrix transpose operator.

A vector is transformed from a reference coordinate system (called the local-level and denoted by  $LL$ ) to a body-fixed coordinate system (denoted by  $B$ ) by subsequent application of these individual transformations. The resulting direction cosine transformation,  $C_{LL}^B$ , starts with  $\psi$ , proceeds with  $\theta$ , and ends with  $\phi$ . It is written as

$$C_{LL}^B = T_1(\phi) T_2(\theta) T_3(\psi)$$

where the subscript  $LL$  denotes the starting coordinate frame, the local-level, and the superscript  $B$  denotes the final coordinate frame, the body axes.

The matrix that makes up the standard form of this transformation is referred to as the direction cosine matrix. This matrix represents the relationship between a vector,  $A$ , and its three components, as shown in Figure 4.

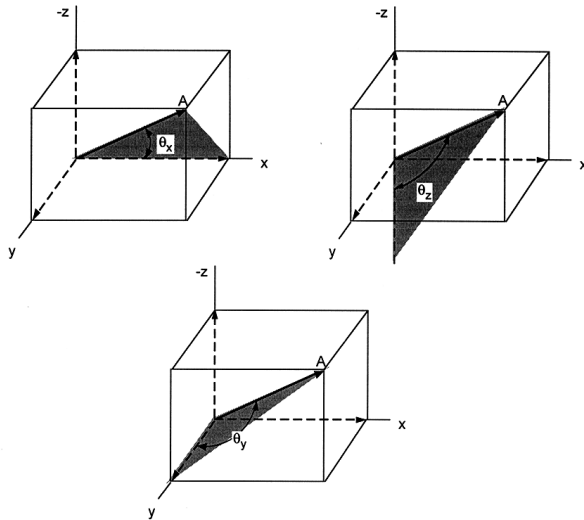


Figure 4. Direction cosine matrix relates a vector,  $A$ , to its  $x$ ,  $y$ , and  $z$ -axis components

The components are related to the angles that  $A$  forms with the  $x$ ,  $y$ , and  $z$  axes by

$$A_x = A \cos \theta_x$$

$$A_y = A \cos \theta_y$$

$$A_z = A \cos \theta_z$$



The direction cosine matrix is a 3 x 3 matrix of elements  $e_{ij}$

$$C_{LL}^B = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}$$

Its counterpart transforms a vector from the body-fixed coordinate system to the local-level and is given by

$$C_B^{LL} = T_3(-\psi) T_2(-\theta) T_1(-\phi)$$

$$= \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}$$

Noting that  $C_B^{LL} = (C_{LL}^B)^T$

$$C_B^{LL} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}^T = \begin{bmatrix} e_{11} & e_{21} & e_{31} \\ e_{12} & e_{22} & e_{32} \\ e_{13} & e_{23} & e_{33} \end{bmatrix}$$

The exact, true orientation of the body (assuming perfect propagation) relative to the local-level coordinate system at any time is extracted from the direction cosine matrix elements

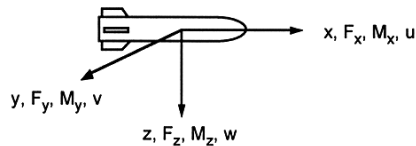
$$\psi = \tan^{-1} \frac{C_B^{LL}{}_{21}}{C_B^{LL}{}_{11}} = \tan^{-1} \frac{e_{12}}{e_{11}}$$

$$\theta = -\sin^{-1} C_B^{LL}{}_{31} = -\sin^{-1} e_{13}$$

$$\phi = \tan^{-1} \frac{C_B^{LL}{}_{32}}{C_B^{LL}{}_{33}} = \tan^{-1} \frac{e_{23}}{e_{33}}$$

## Derivation of the Equations of Motion

A 6-DOF simulation tracks the motion of a rigid body as it moves through the atmosphere. The moving vehicle can rotate as well as translate. The six fundamental differential equations that allow the motion to be tracked are presented below. A body-fixed reference system, as shown in Figure 5, is utilized in the derivation. The body-fixed coordinate system, and three components of force, moment, and velocity are depicted in the figure.



F = Force  
M = Moment (Rate of Change of Angular Momentum)  
u, v, w = Velocity in x, y, and z Axes

Figure 5. Body-fixed axes nomenclature

The desired three rotational accelerations in the body-fixed system are

$$\begin{aligned}\dot{p} &= \frac{M_x}{I_x} \\ \dot{q} &= \frac{M_y + (I_z - I_x) pr}{I_y} \\ \dot{r} &= \frac{M_z + (I_x - I_y) pq}{I_z}\end{aligned}$$

Equivalently, the three desired translational components of acceleration in the body-fixed system are

$$\begin{aligned}\dot{u} &= \frac{F_x}{m} + (rv - qu) \\ \dot{v} &= \frac{F_y}{m} + (pw - ur) \\ \dot{w} &= \frac{F_z}{m} + (qu - pv)\end{aligned}$$

These three translational accelerations and the three rotational accelerations are the desired six equations that form the basis of the equations of motion for a simulation. Because they are highly nonlinear, they are numerically integrated in the 6-DOF. The components of the forces  $F_x$ ,  $F_y$ , and  $F_z$  and the components of the moments  $M_x$ ,  $M_y$ , and  $M_z$  represent the effects of aerodynamics and gravity acting on the vehicle.

## Keeping Track of the True Body Attitude

As a simulated body translates and rotates from the external forces and moments, its attitude must be propagated to determine its true orientation. The simplest approach computes the Euler angle rates ( $\dot{\psi}$ ,  $\dot{\theta}$ , and  $\dot{\phi}$ ) from the three body-fixed rates  $p$ ,  $q$ , and  $r$ . These latter rates are merely the numerically integrated body rotational accelerations computed by the 6-DOF Equations of Motion. The relationship between the Euler angle rates and the body-fixed rates is now derived.

### ***Euler Angle Rates***

The angular velocity,  $\bar{\omega}$ , of a moving body is expressed in terms of its components in the body-fixed axes, shown in Figure 6, as

$$\bar{\omega} = p \bar{e}_{x_B} + q \bar{e}_{y_B} + r \bar{e}_{z_B}$$

where  $\bar{e}_{x_B}$ ,  $\bar{e}_{y_B}$ ,  $\bar{e}_{z_B}$  and are unit vectors along the body-fixed axes.

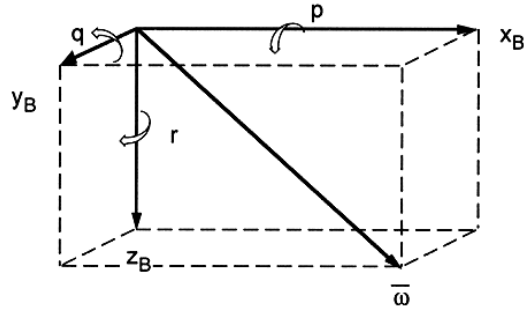


Figure 6. Angular rate nomenclature in body-fixed system

The angular velocity  $\bar{\omega}$  can also be expressed in terms of its components along the three axes of rotation relating the local-level and the body-fixed coordinate systems shown in Figure 3 as

$$\bar{\omega} = \dot{\psi} \bar{e}_z + \dot{\theta} \bar{e}_{y'} + \dot{\phi} \bar{e}_{x_B}$$

The local-level  $z$  and intermediate  $y'$  unit vectors are given in terms of the body-fixed system unit vectors as

$$\bar{e}_z = -\sin \theta \bar{e}_{x_B} + \cos \theta \sin \phi \bar{e}_{y_B} + \cos \theta \cos \phi \bar{e}_{z_B}$$

$$\bar{e}_{y'} = \cos \phi \bar{e}_{y_B} - \sin \phi \bar{e}_{z_B}$$

Setting the two equations for  $\bar{\omega}$  given above equal to each other and substituting for  $\bar{e}_z$  and  $\bar{e}_{y'}$  the Euler angular rates can be solved in terms of the body-fixed rates

$$\dot{\psi} = q \frac{s\phi}{c\theta} - r \frac{c\phi}{c\theta}$$

$$\dot{\theta} = q c\phi - r s\phi$$

$$\dot{\phi} = p + q t\theta s\phi + r t\theta c\phi = p + \dot{\psi} s\theta$$

The  $\dot{\psi}$  equation has a singularity at  $\theta = \pm 90^\circ$ . As the pitch attitude ( $\theta$ ) of the simulated vehicle approaches this condition, the numerical integration may blow up. Simulations that expect a pitch attitude near this singularity replace the propagation of Euler angles with the propagation of quaternions.

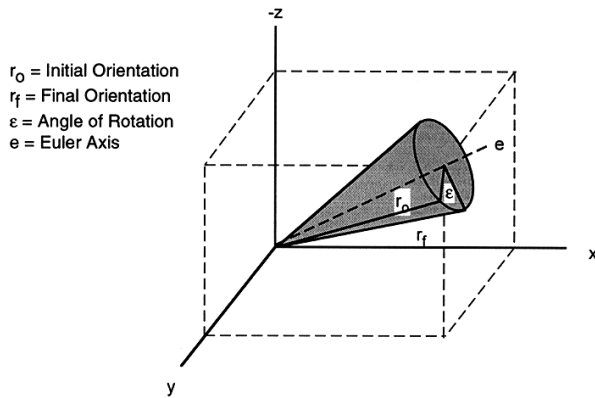


Figure 7. Quaternion angle of rotation of a rigid body about an Euler axis

The usage of quaternions is preferable to the integration of Euler angle rates because they preclude singularities (that is, at  $\theta = \pm 90^\circ$ ). Today's high speed processors suffer very little extra time penalty for the more numerically intensive calculation of quaternions.

## Aerodynamic Forces and Moments

There are three external forces and three external moments that act on the simulated body. These forces and moments are most easily handled in the body-fixed coordinate system. The forces and moments are functions of non-dimensional aerodynamic coefficients, which are based on numerical prediction and may also be verified by wind tunnel tests. The coefficients are converted to forces and moments using dynamic pressure and the vehicle's reference area and diameter. The values provided below are typical for a small projectile about 1 m in length.

$\bar{q} = \frac{1}{2} \rho V^2$	dynamic pressure, $\frac{kg}{m \cdot s^2}$
$\rho$ varies	air density, $kg/m^3$
$V$ varies	total vehicle velocity, $m/s$
$d = 0.125$	diameter, $m$
$S = 0.01227$	cross sectional reference area, $m^2$

Using these definitions, the following describe a basic set of aerodynamic forces and moments that act on an unpowered projectile:

$F_x = C_x \bar{q} S$	aerodynamic force in body x-axis
$F_y = C_y \bar{q} S$	aerodynamic force in body y-axis
$F_z = C_z \bar{q} S$	aerodynamic force in body z-axis
$M_x = C_l \bar{q} S d$	aerodynamic moment about body x-axis
$M_y = C_m \bar{q} S d$	aerodynamic moment about body y-axis
$M_z = C_n \bar{q} S d$	aerodynamic moment about body z-axis

## Aerodynamic Coefficients

The aerodynamic coefficients are given in term of the aerodynamic non-dimensionalized (that is, unitless) stability derivatives:

$$C_x = -C_D$$

$$C_y = C_{y\beta} \beta + C_{y\delta_R} \delta_R$$

$$C_z = -C_{L\alpha} \alpha + C_{L\delta_E} \delta_E$$

$$C_l = C_{\ell_p} \frac{d}{2V} p + C_{\ell\delta_A} \delta_A$$

$$C_m = C_{m\alpha} \alpha + C_{m_q} \frac{d}{2V} q + C_{m\delta_E} \delta_E$$

$$C_n = C_{n\beta} \beta + C_{n_r} \frac{d}{2V} r + C_{n\delta_R} \delta_R$$

The non-dimensionalized stability derivatives are provided by the aerodynamicist for the simulation. These coefficients are predicted based on theory and then verified by a wind tunnel test. For this small projectile example, they are

$C_D = 0.53$	axial drag coefficient
$C_{y\beta} = -6.0$	side force coefficient with side slip angle
$C_{y\delta_R} = 6.0$	side force coefficient with rudder deflection
$C_{L\alpha} = 6.3$	lift coefficient with angle of attack
$C_{L\delta_E} = 6.0$	lift coefficient with changes in elevator deflection
$C_{\ell_p} = -6.6$	roll moment coefficient with roll rate (damping)
$C_{\ell\delta_A} = 6.3$	roll moment coefficient with aileron deflection
$C_{m\alpha} = -5.6$	pitch moment coefficient with angle of attack
$C_{m_q} = -250$	pitch moment coefficient with pitch rate (damping)
$C_{m\delta_E} = 6.0$	pitch moment coefficient with elevator deflection
$C_{n\beta} = 5.6$	yaw moment coefficient with side slip angle
$C_{n_r} = -250$	yaw moment coefficient with yaw rate (damping)
$C_{n\delta_R} = 6.0$	yaw moment coefficient with rudder deflection
$\delta_A$	aileron deflection (left elevator down is positive)

$\delta_R$	rudder deflection ( $\equiv 0$ in this example)
$\delta_E$	elevator deflection ( $\equiv 0$ in this example)
$\alpha$	angle of attack
$\beta$	side slip angle

## Gravity

The acceleration due to gravity always acts in the positive z-axis (down) of the local-level coordinate system. The gravity vector is transformed from the local-level coordinate system to the body-fixed coordinate system and added to the other terms contributing to  $\dot{u}$ ,  $\dot{v}$ , and  $\dot{w}$ .

$$\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}_B = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} e_{13} g \\ e_{23} g \\ e_{33} g \end{bmatrix}$$

## Angles of Attack and Side Slip

The angle of attack is defined as the angle between the wing chord line and the relative wind vector. Since the vehicle in our example has no wings (only tail fins), the angle of attack is the angle between the longitudinal axis and the relative wind in the body x-z plane. The side slip is a similar angle in the body x-y plane. These aerodynamic angles are important aids in assessing a vehicle's ability to maneuver and also its stability. Some aerodynamic coefficients are heavily dependent on angle of attack. (Our simple example assumes a straightforward linear dependence.) As these angles become large a nonlinear region may be entered where forces and moments can be significantly reduced or even result in a change of sign (that is, a change in fin deflection intended to increase  $\alpha$  may not increase the lift force, but may actually reduce it). The effect could be loss of control of the vehicle. These aerodynamic angles are computed as

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \quad \text{angle of attack}$$

$$\beta = \tan^{-1}\left(\frac{v}{u}\right) \quad \text{side slip angle}$$

## Sensed Accelerations

Accelerometers can be used for both navigation and control. An accelerometer is a force sensor, that is, it senses the applied external forces. An accelerometer oriented vertically on a table top actually has its case accelerating upward. The table applies a force equal and opposite to that of gravity; in effect accelerating the case upward. Similarly, an accelerometer attached to the vertical axis of an airframe flying at a constant altitude thinks it is accelerating upward at 1 g even though there is no change in altitude. Accelerometers (for sensors located at the center of gravity) sense the applied external forces including gravity

$$a_x = \frac{F_x}{m} + g_x$$

$$a_y = \frac{F_y}{m} + g_y$$

$$a_z = \frac{F_z}{m} + g_z$$

If the accelerometers are located off the center of gravity additional terms must be added to these equations. These terms account for the accelerations induced by the rotational rates and the distance from the center of gravity.

## Local-Level Velocity and Position

The body-fixed accelerations  $\dot{u}$ ,  $\dot{v}$ , and  $\dot{w}$  are numerically integrated to produce the body-fixed velocities  $u$ ,  $v$ , and  $w$ ; these are then transformed into the local-level coordinate system to keep track of the body velocity in the local-level reference frame

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix} = \begin{bmatrix} e_{11} & e_{21} & e_{31} \\ e_{12} & e_{22} & e_{32} \\ e_{13} & e_{23} & e_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

The three body velocity components in the local-level frame are individually numerically integrated to track the body position

$$x_b = \int \dot{x}_b dt$$

$$y_b = \int \dot{y}_b dt$$

$$z_b = \int \dot{z}_b dt$$

The total velocity of the vehicle is computed from the body components

$$V = \sqrt{\dot{x}_b^2 + \dot{y}_b^2 + \dot{z}_b^2}$$

## Environmental Model

Many environmental models exist; most are implemented with table look-ups or nonlinear equations for the density of air as a function of both altitude and temperature. As noted previously, the dynamic pressure is calculated from the air density and vehicle speed

$$\bar{q} = \frac{1}{2} \rho V^2$$

Some simulations require the speed of sound for calculation of the aerodynamic coefficients. A model for the speed of sound is

$$\text{Speed of Sound} = \sqrt{\gamma R g \text{ } ^\circ K}$$

where

$\gamma$	= 1.4	ratio of specific heats
$R$	= 29.4	universal gas constant, m <sup>2</sup> /s <sup>2</sup> °K
$g$	= 9.8	gravity, m/s <sup>2</sup>
$^\circ K$		air temperature in Kelvin

## **Graphical Simulation**

With the basic elements of a 6-DOF identified, we are ready to bring them together in a simulation. A graphical simulation such as VisSim offers many unique advantages. First and foremost is that a block diagram is significantly easier to comprehend than the line code of a higher order computer language. Block diagrams are also easier to recall and reuse; easier to develop; easier to modify and verify. In a nutshell it is far easier to visualize and comprehend graphical block diagrams than line code.

Good simulation organization and documentation also enhance readability and comprehension as well as intent. VisSim has some nice features that, if used regularly, help in this regard. VisSim has global variable capability. Global variables are available to the entire simulation without interconnecting “wires,” are easily found in a large simulation with a search feature, and are non-redundant in that VisSim prohibits duplicate variable assignments.

## **Simulation Organization**

Organization of the blocks is up to the discretion of the simulation’s designer. A rule of thumb is to make grouping or compounding of blocks reasonable from both a functional standpoint and from a size viewpoint. That is, the individual blocks that are compounded should have a meaningful functional relationship. Organization is enhanced by keeping the number of blocks to some reasonable level of appearance on the computer’s monitor. If a screen gets too crowded, the intent or functionality of the compound block being simulated becomes lost.

The name or title of the compound block should bear a direct relationship to its contents. The relationship can sometimes be expressed by a mathematical equation or by a bitmap image. The bitmap image is a great way to supplement the visual appearance of the simulation and should be used to promote comprehension of the blocks that it contains.

Documentation is also left to the discretion of the designer. Good rules to follow are to adequately label units on variables at their source, and provide reminders as to the meaning or usage of the variables elsewhere in the simulation, especially if there are many compound sub-blocks. VisSim provides two forms of annotation: labels and comments. Routine use of these promotes usability for both the designer and subsequent users.

## **6-DOF Simulation Implementation in VisSim**

Once the equations of motion and quaternions have been implemented in a graphical simulation, they rarely, if ever, have to be modified. They form the core of every 6-DOF simulation. Modifications must obviously be made to represent the aerodynamics and mass properties unique to other newly simulated vehicles. Other obvious modifications are the initial flight conditions and hardware specific to the new vehicle.

Each vehicle typically has unique fin actuators and inertial sensors, and a unique control system and command structure to execute maneuvers. These features are based on performance requirements and are usually included in the simulation after a significant design effort. The design cycle, especially of the control system or autopilot, can be shortened for all vehicles (especially those with nonlinear aerodynamics) by linearization features provided by the graphical simulation tools. VisSim offers the capability to obtain a linear model either in state space or transfer function form by simply selecting an input and an output in the block diagram and selecting the “linearize” option from a pull-down menu.

The linear model helps the designer to specify detailed hardware requirements and to develop the control laws for the vehicle. The models of the hardware and control laws are then implemented in the simulation. Their ability to stabilize and maneuver the vehicle is easily assessed by executing more and more aggressive maneuvers until the limit of performance is achieved. The assessment



is enhanced in VisSim by `display`, `light`, `meter`, `plot`, and `stripChart` blocks. These can be attached to any point in a diagram at any time during the simulation. These visual presentations, updated as the simulation progresses, are immensely helpful in both debugging and performance assessment.

VisSim also has the capability to stop a simulation, change a design parameter and continue the simulation. This feature allows the designer to execute “what if” scenarios on-line. If a response to a command is too slow or too fast, gain changes can be made instantaneously and their effect quickly evaluated.

## Simulation Example

The graphical implementation of our small projectile will now be illustrated. The projectile dynamics are implemented in the Vehicle compound block, shown in Figure 1. This block, shown in Figure 8, contains the Equations of Motion (EOM), the Aerodynamic Forces and Moments, the Stability Derivatives and the Accelerations for input to the accelerometer models (not simulated here).

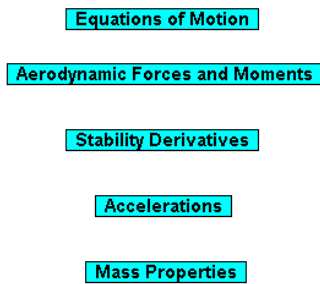


Figure 8. Vehicle compound block

The EOM shown in Figure 9, contains the six basic equations as well as the calculation of velocity and position in the local-level coordinate system. A bitmap image showing the coordinate system reference is included as part of the simulation to enhance user comprehension.

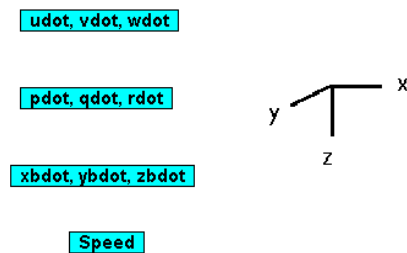


Figure 9. Equations of motion

The  $\dot{u}$ ,  $\dot{v}$ , and  $\dot{w}$  compound block shown in Figure 10, contains the three translational equations. The components of gravity in the body axes are supplied as inputs to each equation.

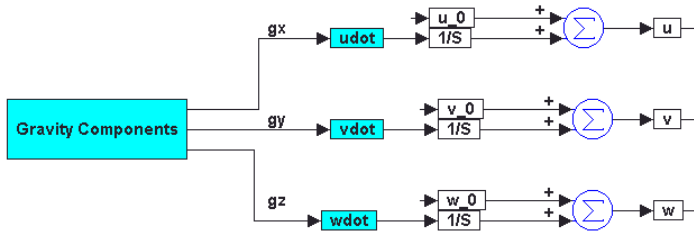


Figure 10. Translational acceleration equations of motion

The implementation of the  $\dot{u}$  equation is given as an example in Figure 11.

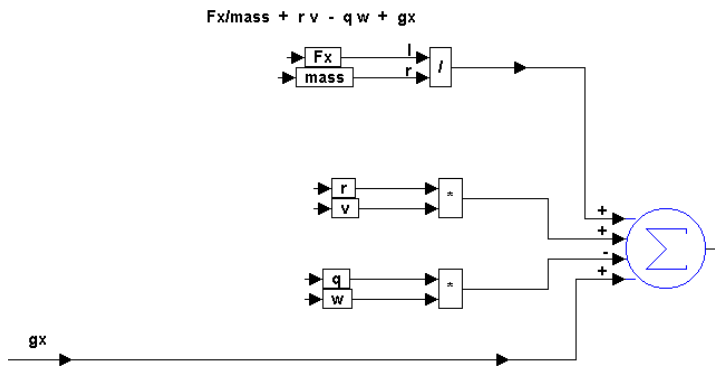


Figure 11.  $\dot{u}$  example equation of motion

Note that the input is labeled and that there is a label for the implemented equation. A similar compound block contains the equations for  $\dot{p}$ ,  $\dot{q}$ , and  $\dot{r}$ .

Figure 12 shows the Aerodynamic Forces and Moments compound block. Note that terms used in multiple equations (that is,  $\bar{q}$ ,  $S$ , and  $d$ ), are assigned as local variables (as indicated by the leading colon) thereby minimizing the number of wires on the screen and simplifying the presentation.

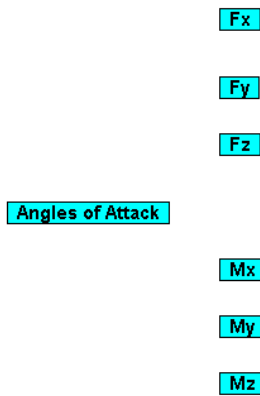


Figure 12. Aerodynamic forces and moments compound block

The fin actuator for our small projectile is represented by a linear second order transfer function, shown in Figure 13. The net roll fin deflection is represented, for convenience in this example, by an equivalent single fin whose angular displacement is the net roll deflection of all four fins. The bandwidth is 100 Hz; the damping ratio is 1.0.

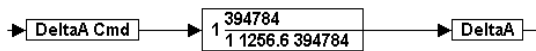


Figure 13. Fin actuator transfer function

The roll autopilot (pitch and yaw are neither implemented nor exercised in this example) employs rate and attitude feedback, as shown in Figure 14. The roll autopilot is only capable of responding to roll attitude commands.

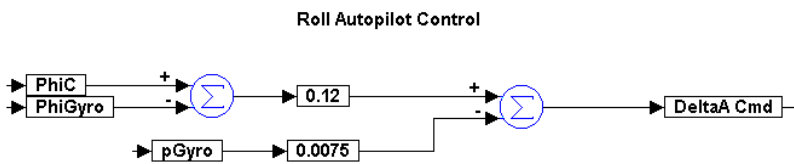


Figure 14. Autopilot compound block showing roll axis only

The initial conditions are set for the trajectory in the Inputs compound block. Here the autopilot step size, initial position, velocity, attitude, body rate, are set along with the environment and vehicle mass properties. For example, Figure 15 shows the initial attitude set in degrees, converted to radians, and recomputed for the four required initial quaternion values.

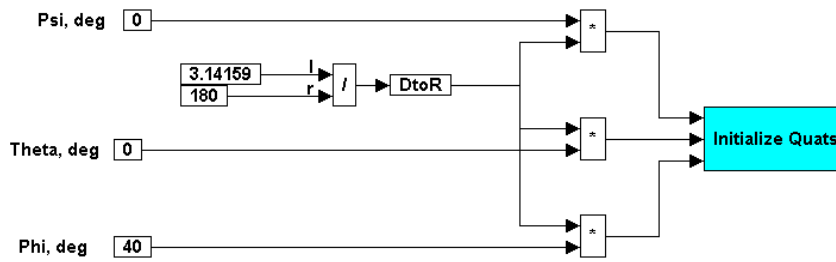


Figure 15. Quaternion initialization from Euler angles

Some results from the simulation of our fictitious vehicle are now briefly explored. A vertical plane trajectory is given in Figure 16 for the entire flight.

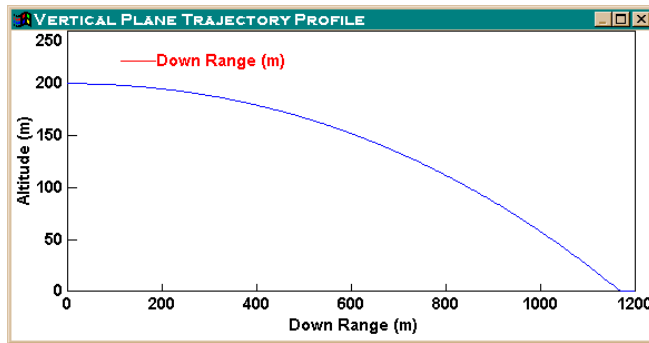


Figure 16. Vertical plane trajectory profile

The vehicle travels nearly 1200 m downrange before impacting the ground. Speed decreases from its initial value of 200 m/sec to about 165 m/sec due to drag, as shown in Figure 17.

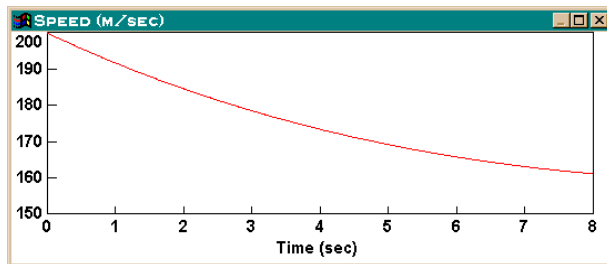


Figure 17. Decrease in speed due to drag

The angle of attack remains small throughout the flight and is shown in Figure 18.

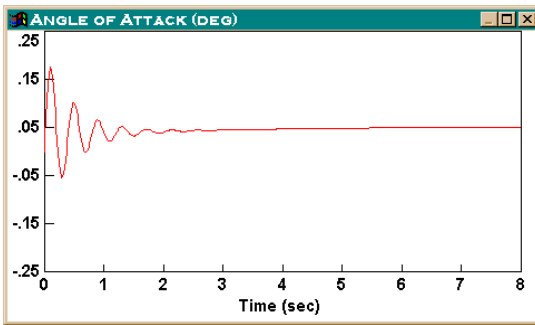


Figure 18. Angle of attack initial transient

The roll axis autopilot executes two maneuvers. The first maneuver immediately commands the vehicle to roll from its initial roll attitude of  $40^\circ$  to  $0^\circ$ . Then, at 0.5 second into the flight, a  $20^\circ$  roll maneuver is commanded. Figure 19 shows that the roll attitude response of the vehicle follows the commands.

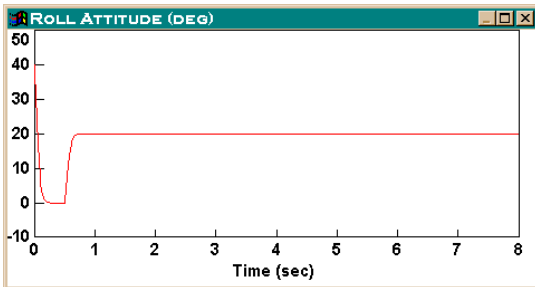


Figure 19. Induced roll maneuver response

The effective net aileron, shown in Figure 20, does not deflect more than  $\pm 5^\circ$  to affect these two maneuvers. If all four fins splay to control roll, then the simulation predicts that each individual fin requires  $\pm 1.25^\circ$  of travel.

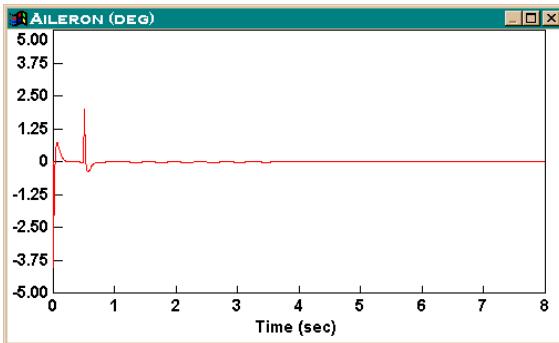


Figure 20. Aileron deflection resulting from roll maneuvers

The simulation integration technique is chosen among several options provided by VisSim. Once the appropriate integration algorithm is selected, the integration step size must be selected. Good rules of thumb are to select a step size at least five times faster than required by the highest bandwidth model and to select an integer sub-multiple of any sample rates used in the control system. A simple check on the adequacy of the step size is to cut it in half and rerun the simulation comparing the results. If the differences are comparatively minor, the larger integration step size is usually adequate.

A convenient feature in VisSim lets the user stop the simulation before the final time specified during simulation set up has been reached. When the input to the `stop` block equals 1, the simulation halts. The feature allows the simulation to be halted when the vehicle hits the ground.

## Simulation Enhancements

Once a basic simulation has been developed rapid growth and model development usually follow. Typical additions fall into two categories: more detailed models and additional models.

In the category of more detailed models the first order of business is usually the fin actuation system. Electric motors and gear trains are common for most small air vehicles. Additional details added to models typically include gear train backlash, motor inductance, resistance and back emf. Sometimes external torques are produced on the fins because the aerodynamic force is not located at the center of fin rotation. These hinge moments can create large opposing torque loads on the fin motor. Simulation of these loads aids in sizing the fin actuation system.

Autopilot performance can degrade because inertial sensors have too low a dynamic bandwidth or, in the case of digital sensors, too large a time delay (latency) in providing data. A typical model of an inertial analog sensor may include a second order transfer function that represents its dynamic response or equivalent linear bandwidth. Rate sensor model details can include sensor bias, acceleration sensitivity and scale factor error. Modeling these errors also helps select the right performance level (and cost) of the instruments needed to perform the flight controls.

More details may be required to faithfully represent the airframe itself. The linear region of operation of many vehicles is usually quite small (below 5° to 10° angle of attack). Maximum performance is achieved by staying within the linear region usually present near the flight aerodynamic trim condition. The aerodynamic representation of the modeled vehicle becomes highly nonlinear; it may require two, three, or even four-dimensional table look-ups for accurate characterization.

The category of additional functionality may include other improvements to more faithfully represent the vehicle or environment being modeled. These functions may include wind, both steady and gusty, a rotating earth model, a non-flat earth model, and on-board simulation of navigation processing algorithms. Two are briefly introduced here: a rotating earth and wind.

The differential equation of quaternions given above that includes Earth rate is [2]

$$\frac{d\bar{q}(t)}{dt} = \frac{1}{2} \bar{q}(t) \bar{\omega} - \frac{1}{2} \bar{\Omega} \bar{q}(t)$$

where the quaternion Earth rate in “four-vector” form is

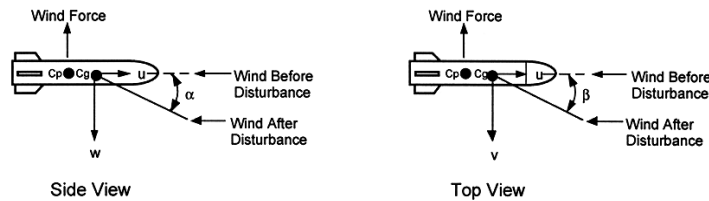
$$\bar{\Omega} = \begin{bmatrix} 0 & -\Omega_x & -\Omega_y & -\Omega_z \\ \Omega_x & 0 & -\Omega_z & \Omega_y \\ \Omega_y & \Omega_z & 0 & -\Omega_x \\ \Omega_z & -\Omega_y & \Omega_x & 0 \end{bmatrix}$$

$\Omega_x$ ,  $\Omega_y$ , and  $\Omega_z$  are the components of Earth rate,  $\Omega_E$ . These components are

$$\begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix} = \begin{bmatrix} (\Omega_E + \dot{\Lambda}) \cos \lambda \\ -\dot{\lambda} \\ -(\Omega_E + \dot{\Lambda}) \sin \lambda \end{bmatrix}$$

where  $\Omega_E = 360^\circ/\text{day} = 0.0000727$  rad/sec,  $\Lambda$  is the vehicle longitude, and  $\lambda$  is the vehicle latitude.

The effect of wind is shown in Figure 21 for the body y-axis ( $M_y$  moment) and z-axis ( $M_z$  moment).



For center of gravity forward of Cp the moment about y-axis (out of paper) is negative. Thus  $Cm_\alpha$  is negative.

Moment about z-axis (into paper) is positive. Thus  $Cn_\beta$  is positive.

Note: Wind force acts at center of pressure (Cp)

Figure 21. Explanation of signs for aerodynamic moments

A head wind with a component from the negative z direction causes an aerodynamic force in the negative z direction that acts at the center of pressure (cp). Most vehicles have the center of gravity (cg) forward of the center of pressure (cp) for stability. A wind force at the cp then causes a negative moment,  $M_y$ , about the y-axis. This negative moment is a restoring moment. Figure 21 also shows a wind from the positive y direction. This wind causes a positive moment,  $M_z$ , about the z-axis. This positive moment is a restoring moment. For a cg forward of a cp,  $Cm_\alpha$  is negative and  $Cn_\beta$  is positive.

## Monte Carlo Simulations

Monte Carlo simulations predict performance for real life systems having random signals or conditions. A Monte Carlo simulation typically simulates a trajectory over and over again with varying random inputs or events. The inputs can be random initial conditions such as variations in launch velocity or random noise that simulates real life inertial instruments such as a noisy rate gyro. Random events may be a one-time wind gust or a hardware failure.

VisSim has Monte Carlo capability by allowing random number generation and a random number seed to be set prior to the start of a simulation. A random seed can be specified for each Monte Carlo bundle. Thus, the same sequence of pseudo-random numbers will be generated for the same seed. Different seeds produce different sequences of pseudo-random numbers. The same seed used over and over allows the designer to evaluate the sensitivity to individual design parameter changes. VisSim also allows the collection of statistics at the end of each individual run in the Monte Carlo bundle. An example of the purpose of a Monte Carlo simulation is to determine the

impact dispersion of a projectile due to random tip-off rates at the end of a gun barrel or to variations in propellant strength.

VisSim allows the importation of Dynamic Link Libraries (DLLs). The DLLs are particularly useful for functions requiring matrix algebra. Guidance laws, Kalman filter target trackers, and end of run statistical processing functions have all been incorporated into 6-DOF simulations with VisSim. The functions are first coded in FORTRAN, C, or PASCAL, then compiled and linked into VisSim with a `userFunction` block. DLL outputs can either be viewed with any of the standard VisSim signal consumer block, or written to a file directly from the DLL for later post run analyses.

Graphical 6-DOF simulations are especially useful to the autopilot designer. A standard, classical solution to the control problem is to design a control law based on a linearized model of the plant (in this case the vehicle being simulated). This approach usually works well on paper. In real life electronic amplifiers saturate, actuators have slew rate limits, rate sensors do not have infinitesimal sensitivity, A/D and D/A converters have a finite number of bits, and computers have computational processing delays. The effects of these nonlinearities are easily evaluated visually by displaying, as the simulation is running, a plot of the key performance variables. VisSim allows the simulation to be stopped, scales on plots to be readjusted, and the simulation to continue. If the performance during simulation degrades or becomes unacceptable, the simulation can be immediately stopped with changes made to the control laws. Graphical simulation such as VisSim allows rapid prototyping of control laws.

## Summary

Simulation of air vehicles brings many advantages to the development process, not the least of which is cost reduction. We have examined the essential elements that constitute a 6-DOF simulation including coordinate systems and transformations, equations of motion, and the forces and moments acting on the vehicle. The essential elements were then assembled into a simulation. A specific, yet generic, example of a small projectile was presented to highlight the advantages of a graphical simulation such as Visual Solutions's VisSim graphical simulation tool.

## References

- Savage, Paul G., "Strapdown System Algorithms," AGARD Lecture Series No. 113, ADVANCES IN STRAPDOWN INERTIAL SYSTEMS, May, 1984.
- Shibata, Minoru, "Error Analysis Strapdown Inertial Navigation Using Quaternions," AIAA J. Guidance, May-June, 1986.