BSR/AIAA
S-119-200X

Draft American National Standard

Flight Dynamics Model Exchange Standard

Warning

This document is not an approved AIAA Standard. It is distributed for review and comment. It is subject to change without notice.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Sponsored by

American Institute of Aeronautics and Astronautics

Approved XX Month 200X

American National Standards Institute

Abstract

This is a standard for the interchange of simulation modeling data between facilities. The initial objective is to allow a person with a simulation of a certain type of vehicle or aircraft at facility A to transfer the simulation to facility B in an easy, straightforward manner.

LIBRARY OF CONGRESS CATALOGING DATA WILL BE ADDED HERE BY AIAA STAFF

Published by
American Institute of Aeronautics and Astronautics
1801 Alexander Bell Drive, Reston, VA 20191

Printed in the United States of America

Contents

Foreword

This standard was sponsored and developed by the AIAA Modeling and Simulation Committee on Standards. Mr. Bruce Jackson of NASA Langley conceived DAVE-ML. DAVE-ML is the embodiment of the standard in XML. This document is the data type descriptions for the XML implementation and includes examples of its use. (Annex B)

This implementation was then tested by trial exchange of simulation models between NASA Langley Research Center (Mr. Bruce Jackson), NASA Ames Research Center (Mr. Thomas Alderete and Mr. Bill Cleveland), and the Naval Air Systems Command (Mr. William McNamara and Mr. Brent York). Numerous improvements to the standard resulted from this “testing”.

At the time of approval, the members of the AIAA **Modeling and Simulation CoS** were:

Bruce Hildreth, Chair	Science Applications International Corporation (SAIC)
Bruce Jackson, DAVE-ML Lead	NASA Langley Research Center
Bimal Aponso	NASA Ames Research Center
Jon Berndt	Jacobs
Geoff Brian	Defense Science Technical Organization (DSTO)
Victoria Chung	NASA Langley Research Center
Peter Grant	University of Toronto
Michael Madden	NASA Langley Research Center
Michael Silvestro	Charles Stark Draper Laboratory, Inc.
Jean Slane	Engineering Systems Inc.
Brent York	Indra Systems, Inc.

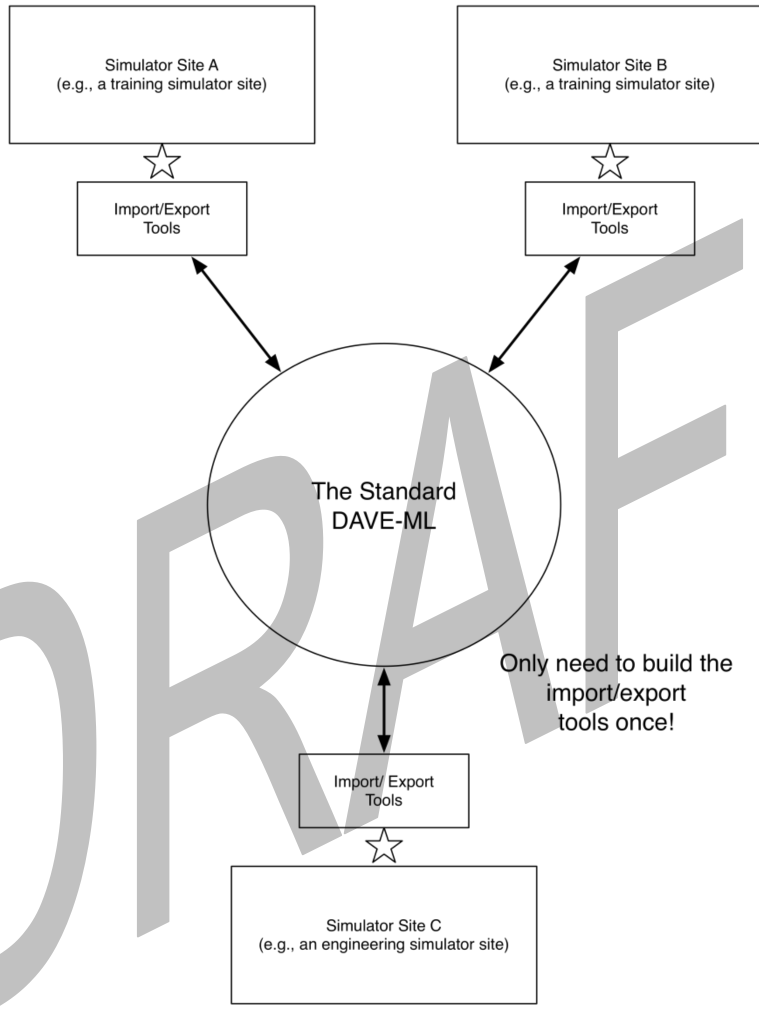
The above consensus body approved this document in Month 200X.

The AIAA Standards Executive Council (VP-Standards Name, Chairman) accepted the document for publication in Month 200X.

The AIAA Standards Procedures dictates that all approved Standards, Recommended Practices, and Guides are advisory only. Their use by anyone engaged in industry or trade is entirely voluntary. There is no agreement to adhere to any AIAA standards publication and no commitment to conform to or be guided by standards reports. In formulating, revising, and approving standards publications, the committees on standards will not consider patents that may apply to the subject matter. Prospective users of the publications are responsible for protecting themselves against liability for infringement of patents or copyright or both.

Introduction

The purpose of this standard is to clearly define the information and format required to exchange air vehicle simulation models between simulation facilities (see the figure below). The standard is implemented in XML and called DAVE-ML.



The Exchange Standard (DAVE-ML) Includes:

— Standard variable name definitions — the purpose of this is to facilitate the transfer of information by using these standard variables as a “common language”. The DAVE-ML standard can be used without using standard variable names, however it will be more difficult because the person exporting the model will have to explicitly define all the variables instead of just a subset unique to the particular model.

— Standard function table definition — this allows easy transfer of non-linear function tables of n dimensions.

— Standard axis system definitions — this is used by the variable names and function tables to clearly define the information being exchanged.

— Standard static math equation representation — for definition of aero model (or other static models) equations. This is implemented using Math-ML.

XML provides a format for the exchange of this information, therefore each organization is required to design import/export tools which comply to the standard one time only.

Use of this standard will result in substantially reduced cost and time necessary to exchange aerospace simulations and model information. Test cases have indicated an order of magnitude reduction in effort to exchange simple models when utilizing this standard. Even greater benefits could be attained for large or complicated models.

Trademarks

The following commercial products that require trademark designation are mentioned in this document. This information is given for the convenience of users of this document and does not constitute an endorsement. Equivalent products may be used if they can be shown to lead to the same results.

Simulink ®

MATLAB®

1 Scope

This standard establishes the definition of the information and format used to exchange air vehicle simulations and validation data between disparate simulation facilities. This standard is not meant to require facilities to change their internal formats or standards. With the concept of an exchange standard, facilities are free to retain their well-known and trusted simulation hardware and software infrastructures. The model is exchanged through the standard, so each facility only needs to create import/export tools to the standard once. These tools can then be used to exchange models with any facility at minimal effort, rather than creating unique import/export tools for every exchange.

The standard includes a detailed convention for defining simulation variables. The purpose of this is to assist in the clarification of the information within the model when it is exchanged between two simulation customers or facilities. Such clarification includes axis systems referenced, units, and sign conventions used. XML is used as the mechanism to facilitate automation of the exchange of the information. Using the definitions in the standard, a list of simulation variable names and axis systems is included. This list of standard variable names further simplifies the exchange of information, but is not required.

2 Tailoring

When viewed from the perspective of a specific program or project context, the requirements defined in this Standard may be tailored to match the actual requirements of the particular program or project. Tailoring of requirements shall be undertaken in consultation with the procuring authority where applicable.

NOTE Tailoring is a process by which individual requirements or specifications, standards, and related documents are evaluated and made applicable to a specific program or project by selection, and in some exceptional cases, modification and addition of requirements in the standards.

The following sections provide further guidance on specific tailoring situations.

2.1 Partial Use of the Standard

2.1.1 General

Each simulation created may not require the implementation of all aspects of this standard. The following guidelines are provided to encourage appropriate use of the standard in a number of example situations.

2.1.2 Creating a New Simulation Environment

This situation calls for use of the complete standard. In this situation it is hoped that the team developing this new simulation would add to the list of standard variables and axis systems.

2.1.3 Creating a New Simulation Model in an Existing Simulation Environment

This situation is defined as creating a new system model (aircraft dynamic model for example) that will run in an existing simulation environment. It is expected that this is the most commonly performed work that will see benefit by application of this standard.

In this case the following tailoring guidelines are applicable. Apply the standard to the new development aspects of the project and all the function tables. Assuming that most or all of the standard variable names and axis systems are applicable to the simulation, use them for the new code developed for the simulation. In the existing simulation environment that is being reused, for example the equations of motion, there is no need to rewrite the code to use the standard variable names or axis systems. However, in most cases the axis systems used in existing simulation environments will be covered in the standard axis system definitions herein (Section 6). Therefore the standard axis systems can easily be referenced when documenting the simulation and interfaces between the new simulation components and those reused.

2.1.4 Creating or Updating a Simulation with a Long Life Expectancy

A pilot training simulator is an excellent example of this type of simulation. This simulation may only be updated every 3-10 years, so at first glance the standard may seem to be less applicable.

In fact the opposite is true. It is because of the infrequent maintenance that the standard is critical. In this case, in each new software update, the original developers (or last updaters) are probably gone, and the update is being done by new personnel. Therefore, software developed under the standard is much easier to understand by the new software team. They would be working with clear variable definitions that they are familiar with. The function table format is understood and the functions themselves better documented. Changes are recorded for the next software update team some years down the road. The axis system definitions are clear.

In simulations with long expected life, use of the state, state derivative and control conventions as part of the naming convention becomes critical as these variables form the core of the model and control of it. It is critical that the personnel modifying the simulation are able to easily find the states, state derivatives and controls.

2.2 New and Reused Software Tailoring Guidance

The longer the expected life of the simulation, the more important the use of the standard becomes. The above tailoring guidelines may be categorized into two common situations; new and reused code.

New simulation code should

— use standard axis system definitions (Section 5) where they coincide with the definitions in the standard;

— use standard variable names (Section 6) to facilitate consistency and simplify documentation requirements;

— apply the convention for states, state derivatives and controls wherever possible; and

— use standard function tables (Section 7) for ALL function tables.

NOTE This facilitates consistency in the data, the documentation of the data, and collaboration with other organizations to improve or debug the data.

Reused simulation code should reference the standard only when convenient to document interfaces with new code.

2.3 Creating New Variable Names and Axis Systems

The standard variable names and axis system definitions are included in the standard to facilitate communication. They provide a “common language” for the exchange. For example, it is not enough to exchange the lift coefficient function. As a minimum, the independent variables used to define the function and their units, sign convention, and reference axis system must be defined. This is facilitated by having standard variable names and axis systems. Of course, new variable names, definitions, and other convenient axis systems may be used to exchange models between simulation facilities. However, in such cases, the exporters and importers must carefully define these variables and axes, otherwise the exchanged model may not produce the desired results. Use of standard variable names and axis systems facilitates the exchange.

This standard includes a methodology for creating new standard variables. Its use is encouraged. Annex C provides the URL for submitting additional standard variable names and axis systems or comments on existing standard variable names and axis systems.

3 Applicable Documents

The following documents contain provisions which, through reference in this text, constitute provisions of this standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

AIAA R-004-1992 *Atmospheric and Space Flight Vehicle Coordinate Systems*

4 Vocabulary

4.1 Acronyms and Abbreviated Terms

a/c	Aircraft
AIAA	American Institute of Aeronautics and Astronautics
ANSI	American National Standards Institute
cg	Center of gravity
DIS	Distributed Interactive Simulation
FE	Flat Earth axis system
GE	Geocentric Earth fixed axis system
MathML	Mathematical Markup Language
SI	Système Internationale d'Unites
w.r.t	with respect to
XML	Extensible Markup Language

4.2 Terms and Definitions

For the purposes of this document, the following terms and definitions apply.

Breakpoint
the value of the independent variable of a given dependent variable, or the X coordinate (or abscissa) of a one dimensional table

Confidence Interval
an estimate of the computed or perceived accuracy of the data

Dependent Variable
the output of a function table

EXAMPLE For $C_L(\alpha,\beta)$, C_L is the dependent variable, also called the output.

Independent Variable
the input(s) to a function table

EXAMPLE For $C_L(\alpha,\beta)$, α and β are independent variables.

One Dimensional Table
a table containing only one independent variable

EXAMPLE $C_L(\alpha)$ is a one dimensional table.

Two Dimensional Table
a table containing two independent variables

EXAMPLE $C_L(\alpha,\beta)$ is a two dimensional table.

Static Equation
a mathematical statement where the output (left hand side) does not have direct dependence (right hand side) on a simulation state

Simulation States (State Derivatives)
in the formulation of a simulation model shown as

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

— x represents a vector of the simulation states.

— $\dot{x}$ represents a vector of the simulation state derivatives.

— u represents a vector of the simulation controls (inputs)

Function Table
the numeral set of data points used to represent non-linear relationships between an independent variable based on (as a function of) one or more independent variables

EXAMPLE $C_L(\alpha,\beta)$ is represented by a function table.

Gridded Table
a multi-dimensional function table in which the independent variable breakpoints do not change for different values of other independent variables

NOTE 1 This is sometimes called an orthogonal table.

NOTE 2 All one-dimensional tables are gridded tables.

Ungridded Table
a multi-dimensional function table in which the independent variable breakpoints change for different values of other independent variables

NOTE This is sometimes called a non-orthogonal table.

5 Standard Simulation Axis Systems

5.1 Background / Philosophy

The axis system definitions discussed herein were taken from existing standards, the *ANSI/AIAA Recommended Practice for Atmospheric and Space Flight Vehicle Coordinate Systems* (ANSI/AIAA R-004-1992) and the *Distributed Interactive Simulation* (DIS Application Protocols, Version 2, IST-CR-90-50, March 1994). AIAA R-004-1992 is based on ISO 1151-1:1988 and ISO 1151-3:1972.

Axis system standards are also reflected in the variable naming convention. When applicable, the axis system is included in the variable name (see Section 6).

5.1.1 Axis System Conventions

In general, ANSI/AIAA R-004-1992 should be referred to as the normative reference for axis system definitions. These axis systems are discussed in Table 1. However, it is important to emphasize the correlation of the AIAA document and the *Distributed Interactive Simulation* (DIS) axis systems. The geocentric earth fixed axis system and body axis coordinate system axis system both are used in DIS and High Level Architecture (HLA) simulations.

5.1.1.1 Geocentric Earth Fixed-Axis System

The Geocentric Earth Fixed-Axis System (Axis System 1.1.3 of the table below) is identical to the DIS "Geocentric Cartesian Coordinate System" (also referred to as "World Coordinate System" in the DIS).

It is a system with both the origin and axis fixed relative to, and rotating with, the earth. The origin is at the center of the earth, the x_G axis being the continuation of the line from the center of the earth through the intersection of the Greenwich Meridian and the Equator, the z_G axis being the mean spin axis of the earth, positive the north, and the y_G axis completing the right hand triad.

All variables in the simulation referenced to this axis system refer to the "GE" for the Geocentric Earth Fixed-Axis System. This axis system is also frequently called "Earth Centered Earth Fixed".

5.1.1.2 Body Axis Coordinate System

Another standard axis system is the Body Axis System (axis system number 1.1.7 in ANSI/AIAA R-004-1992). This is identical to the DIS "Entity Coordinates System".

The body axis system is referred to in the variable names as "Body".

5.1.1.3 Additional Axis Systems

In addition to the axis systems defined in ANSI/AIAA R-004-1992, this standard has added the Flat Earth and Locally Level axis systems. These axis system is are defined only for convenience onfor use in simple simulations and for creating validation data.

The Flat Earth axis system is a fixed, non-rotating, flat earth with no mapping to a round earth coordinate system, therefore, latitude and longitude are meaningless. The purpose of this coordinate system is to allow, if desired, vehicle checkout simulation to be performed in this axis system. This simplifies the use of this standard by the simulation facilities which do not normally use a round or oblate spheroid, rotating earth model.

The Flat Earth reference system is situated on the earth's surface directly under the cg of the vehicle at the initialization of the simulation. The x axis on the local frame points northwards and the y axis points eastward, with the z axis down. The x and y axis are parallel to the plane of the flat earth.

The flat earth axis system is referred to in the variable names as "FE".

The locally level axis system is also a simplified axis system convenient for simulation checkout and validation. The –Z axis passes through the vehicle CG. If a flat earth, the X axis is in the plane of the surface and oriented toward true North. The Y axis is also in the plane of the surface and completes the right hand triad (East). If not at flat earth, the X axis is tangential to the smooth surface of the earth and oriented toward true North in the geometric earth model. The Y axis is tangential to the smooth surface of the earth completing the right hand triad (East).

The locally level axis system is referred to in variable names as "LL".

5.1.1.4 Complete List of Axis Systems

The axis systems that are referenced are taken largely from paragraph 1.1 of ANSI/AIAA R-004-1992. The flat earth and locally level axis systems for atmospheric flight simulation approximation are added to that reference. Table 1 is the comprehensive list of axis systems that may be used.

The first column in Table 1 provides the abbreviation used for each axis system. The axis system may be referenced in a variable name. See Section 6 on the variable naming convention.

Table 1 — Standard axis systems

Reference Abbreviation for Variable Names	R-004-1992 Paragraph Number	Term	Definition	Symbol
EI (Earth centered inertial)	1.1.1	Geocentric inertial axis system (See Appendix D.2 of R-004 for a modification of this system used for launch vehicles.)	An inertial reference system of the FK5 mean equator and equinox of J2000.0 has the origin at the center of the Earth, the X_{Γ}-axis being the continuation of the line from the center of the Earth through the center of the Sun toward the vernal equinox, the Z_{Γ}-axis pointing in the direction of the mean equatorial plane's north pole, and the Y_{Γ}-axis completing the right-hand system. (See Figure 1A in R-004)	$x_I y_I z_I$
Not used, this forms a basis for other definitions	1.1.2	Earth-fixed axis system	A right-hand coordinate system, fixed relative to and rotating with the Earth, with the origin and axes directions chosen as appropriate.	$x_o y_o z_o$
GE (also called ECEF for Earth centered Earth fixed)	1.1.3	Geocentric Earth-fixed axis system	A system with both the origin and axes fixed relative to and rotating with the Earth (1.1.2). The origin is at the center of the Earth, the x_G-axis being the continuation of the line from the center of the Earth through the intersection of the Greenwich meridian and the equator, the z_G-axis being the mean spin axis of the Earth, positive to the north, and the y_G-axis completing the right-hand system. (See Appendix D.3 in R-004-1992)	$x_G y_G z_G$
	1.1.4	Normal Earth-fixed axis system	An Earth-fixed axis system (1.1.2) in which the z_o-axis is oriented according to the downward vertical passing through the origin (from the origin to the nadir). (See Figure 1C in R-004-1992)	$x_o y_o z_o$ ($x_g y_g z_g$ is an acceptable alternative)
VO	1.1.5	Vehicle-carried orbit-defined axis system[a]	A system with the origin fixed in the vehicle, usually the center of mass, in which the z_o-axis is directed from the spacecraft toward the nadir, the y_o-axis is normal to the orbit plane (positive to the right when looking in the direction of the spacecraft velocity), and the x_o-axis completes the right-hand system. (See Figure 1A in R-004-1992)	$x_o y_o z_o$
VE	1.1.6	Vehicle-carried normal Earth axis system[a]	A system in which each axis has the same direction as the corresponding normal Earth-fixed axis, with the origin fixed in the vehicle, usually the center of mass.	$x_o y_o z_o$ ($x_g y_g z_g$ is an acceptable alternative)
Body	1.1.7	Body axis system[a]	A system fixed in the vehicle, with the origin, usually the center or mass, consisting of the	$x_B y_B z_B$

Reference Abbreviation for Variable Names	R-004-1992 Paragraph Number	Term	Definition	Symbol
			following axes:	
		Longitudinal axis	An axis in the reference plane or, if the origin is outside that plane, in the plane through the origin, parallel to the reference plane, and positive forward.[b] In aircraft or missiles, this is normally from the CG forward towards the nose in the vertical plane of symmetry. It is also normally parallel to the waterline of the vehicle.	x_B
		Lateral axis	An axis normal to the reference plane and positive to the right of the x-axis (henceforth, positive to the right).	y_B
		Normal axis	An axis which lies in or parallel to the reference plane, whose positive direction is chosen to complete the orthogonal, right-hand system xyz.	z_B
Wind (for wind axis system)	1.1.8	Air-path system[a]	A system with the origin fixed in the vehicle, usually the center of mass, consistent of the following axes:	$x_w y_w z_w$
		x_a-axis; air-path axis	An axis in the direction of the vehicle velocity relative to the air (1.5.1).	x_w
		y_a-axis; lateral air-path axis; cross-stream axis	An axis normal to the air-path axis and positive to the right.	y_w
		z_a-axis; normal air-path axis	An axis — in the reference plane or, if the origin is outside that plane, parallel to the reference plane, and — normal to the air-path axis. The positive direction of the z_a-axis is chosen so as to complete the orthogonal, right-hand system $x_a y_a z_a$.	z_w
SA (for stability axis system)	1.1.9	Intermediate axis system[a]	A system with the origin fixed in the vehicle, usually the center of mass, consisting of the following axes.	$x_s y_s z_s$
		x_e-axis	The projection of the air-path axis on the reference plane, or, if the origin is outside that lane, on the plane through the origin, parallel to the reference plane.	x_s

Reference Abbreviation for Variable Names	R-004-1992 Paragraph Number	Term	Definition	Symbol
		y_e-axis z_e-axis	An axis normal to the reference plane and positive to the right, coinciding with or parallel to the lateral axis (1.1.7). An axis which coincides with or is parallel to the normal air-path axis so as to complete the orthogonal right-hand system.	y_s z_s
FP	1.1.10	Flight-path axis system[a]	A system with the origin fixed in the vehicle (usually the center of mass) and in which the x_k-axis is in the direction of the flight-path velocity relative to the Earth. The y_k axis is normal to the plane of symmetry and positive to the right. The z_k axis completes the orthogonal right-hand system	$x_k y_k z_k$
AA	1.1.11	Total-angle-of-attack axis system[a] (USA practice: areoballistic axis system.)	A system with the origin fixed in the vehicle, usually the center of mass, in which the x_Γ-axis is coincident with the x-axis in the body axis system (1.1.7). The y_Γ axis is perpendicular to the plane formed by the x_Γ-axis and the velocity vector, positive to the right. The z_Γ axis is formed to complete the orthogonal, right-hand system.	$x_\Gamma y_\Gamma z_\Gamma$
FE		Flat Earth system (not from R-004-1992)	The Flat Earth reference system is situated on the earth's surface directly under the cg of the vehicle at the initialization of the simulation. The x axis on the local frame points northwards and the y axis points eastward, with the z axis down. The x and y axis are parallel to the plane of the flat earth.	$x_{FE} y_{FE\Gamma} z_{FE}$
LL		Locally Level axis system (not from reference R-004-1992)	A vehicle related axis system (1.1.6) with the origin on the smooth surface of the earth and moving with the vehicle. The –Z axis passes through the vehicle CG. The X axis is tangential to the smooth surface of the earth and oriented toward true north in the geometric earth model. The Y axis is tangential to the smooth surface of the earth completing the right hand triad (East).	$x_{LL} y_{LL\Gamma} z_{LL}$

[a] Usually the origins of the axis systems defined in 1.1.5 through 1.1.11 coincide. If that is not the case, it is necessary to distinguish the different origins by appropriate suffixes.

Reference Abbreviation for Variable Names	R-004-1992 Paragraph Number	Term	Definition	Symbol
[b] The reference plane should be a plane of symmetry, or a clearly specified alternative.				

5.2 Summary

This axis system standard should be followed for all future equations of motion. Additionally, it provides the naming convention to properly reference the definitions herein in simulation variable names.

5.3 References

ANSI/AIAA Recommended Practice R-004-1992, *Atmospheric and Space Flight Vehicle Coordinate Systems*, 28 February 1992.

Distributed Interactive Simulation (DIS Application Protocols, Version 2, IST-CR-90-50, March 1994)

6 Standard Simulation Variables

6.1 Background / Philosophy

6.1.1 Rationale for Having Standard Variable Name and Naming Conventions

The standard variable names and axis system definitions are part of the standard to facilitate communication. They provide a "common language" for information exchange. For example, it is not enough to exchange the lift coefficient function. As a minimum the independent variables used to define the function and their units, sign convention, and reference axis system must be defined. This is facilitated by having standard variable names and axis systems.

Therefore, if you exchange models using the standard variables, you don't have to define a variable that is part of the standard, just refer to the standard for the definition of that variable. Additionally, the variable naming convention is presented to allow the list of standard variables to grow as needed by the user community. Hopefully the convention will keep some consistency in the variable names and make them easier for users to interpret.

6.1.2 States and State Derivatives

Long-term maintenance of simulation software used to model the flight dynamics of an airplane is predicated upon identification of the states and controls in the simulation. The importance of this cannot be overstated. States and inputs (controls) are determined by the physics of the problem. Since the physics are immutable the identification of these variables is crucial in software maintenance.

Again, according to physics, all outputs which are used in simulation are derived from states and inputs.

By practice, anything in a simulation of interest is an output. To create an output, for example indicated airspeed, it is necessary to identify the states and inputs. Therefore, if the appropriate law of physics is known, the indicated airspeed may be correctly computed. Too often in simulation modeling these immutable fundamental concepts are forgotten. Approximations are made that commonly create states from outputs. Practically speaking, this is done because the states in the simulation cannot be determined. Since a simulation is an iterative process, it becomes unclear as to what variable is dependent upon what other variable.

Rigorous physics requires that everything is computed from states and inputs.

Now the question becomes which state, that is, state at what time? Again, physics and discrete math require that outputs at any time T are a function of the states and inputs at time T. Integration of the state derivatives at time T results in states at time T plus delta T. State derivatives at time T are functions of the states and inputs at time T. It is crucial that variables at time T are not mixed with variables at time T plus delta T.

Practically speaking, for simulation standards, what this means is that all integrations must be done in a centralized location in each simulation loop, otherwise variables at time T are mixed with variables at time T plus delta T. The simulation industry has violated this mathematical principle for many years in the use of “in-line” difference equations for simulation filters and actuators, etc. This often works “OK” and “no harm is done”. However, what is missed is that software maintenance becomes much more difficult when those states cannot be located because they are embedded—they are strung throughout the code. Therefore, the outputs cannot be properly created. Furthermore, when a modification comes to add a capability or to fix a bug, the key variables required to modify a simulation (again what would the states, state derivatives, and inputs) are impossible to find or inaccessible. Therefore, the fix made to the simulation is less than optimum and possibly creates more errors down the road for the next fix, etc., etc., ad infinitum.

The identification of states and state derivatives is simply for the purpose of encouraging good mathematical fundamentals and to facilitate software maintenance. Therefore this AIAA Simulation Variable standard identifies states and state derivatives as part of the naming convention.

Identification of controls (also called inputs), while a good idea, is very difficult because so many variables are controls and the controls change with the mode of operation of the simulation. As a consequence, identification of controls is optional but should be strongly considered for inclusion in the development of new dynamic simulation models.

6.2 Variable Naming Convention

This clause will discuss the convention and philosophy used for naming simulation variables. This explanation is intended to ensure that new variables defined in the future are consistent with existing variables.

The mixed case variable name convention is used with one exception. The standard uses an underscore to separate the prefix and suffix from the body of the variable name. The standard could also be followed using underscores to separate the parts of the variable names.

The following general rules for naming variables shall be followed.

— Variables shall have meaningful names.

— Mnemonics shall not be used.

— Standard abbreviations are permitted.

— The first word in the variable name (not including the prefix, if any) shall start with a lower case. Distinct words thereafter in variable names shall be capitalized (for example, `angleofAttack_d`).

— Variable names shall not exceed 63 characters in length. Brief, but complete names are most effective.

— Abbreviations are generally all capitals.

6.3 Variable Name Creation Methodology

The suggested method of creating the name is as follows.

— Each name has up to eight components.

— All components are not required to be used because in many cases they do not apply.

— These components are:

1. (prefix)_
2. (variable domain)
3. (specific axis or reference)
4. (axis or reference system)
5. (core name)
6. Of (point on the vehicle)

   NOTE Generally only for positions, velocities and accelerations

7. WRT (reference point or frame)

   NOTE Generally only for positions, velocities and accelerations

8. _(units).

Very rarely, if ever, are all 8 components of a name used.

6.3.1 Prefix

The prefix is used to identify the most important dynamic variables in the simulation, the states and the state derivatives. (See Section 6.1.2)

The prefix shall be separated from the body of the variable by an underscore or as a separate component of a structure.

6.3.1.1 Identification of States and State Derivatives

The states and state derivatives are those variables which make the simulation dynamic and are the key variables in a real time flight simulation. Basically, anything that is integrated (mathematically) is a state derivative. The result of the integration is the state (integration of the state derivative results in the state). This is true for any integration in a simulation. If the user controls all the states, he controls the motion of the simulation. Also, these along with the controls (inputs) are the key variables for validation. All outputs are computed directly or indirectly from states and controls.

The formulation of the equations of motion and the model itself determines what variables are states. This naming convention is not meant to standardize on any variable as a state, just for the simulation engineer to explicitly identify them in the model implementation, making it easier to document and exchange the models.

Examples:

`s_XBodyVelocity_fs_1`	s_ prefix indicates that this variable is a state
`sd_XBodyAcceleration_fs_2`	sd_ prefix indicates that this variable is a state derivative

6.3.1.2 Identification of Controls (optional)

The controls are those variables which provide the pilot/crew or the simulation operator's inputs to the simulation. As with the states and state derivatives, the controls are the key variables for validation. All outputs are computed directly or indirectly from states and controls.

The formulation of the equations of motion and the model itself determines what variables are controls. This naming convention is not meant to standardize on any variable as a control, just for the simulation engineer to explicitly name them, making it easier to document and exchange the models.

Examples:

`c_avgAileronDeflection_d`	c_ prefix indicates that this variable is a control
`c_pilotLongControlPos_r`	c_ prefix indicates that this variable is a control

6.3.2 Variable Source Domain

This represents the domain in which the variable is calculated. In object oriented design, it could logically be the object. The domain is normally not included if it (or the object) is the vehicle or aircraft being simulated, for example, airspeed.

Some domain examples include:

— Aero

— Engine or Thrust

— Controls

— Guidance

— Navigation

— GNC

— Wheel

— Landing Gear

— Hydraulic

— Electrical

— IO (for input/output)

— Motion

— CL or Control Loading

— Radar

— Weapons

— AIM9X (as an example, for the AIM-9X missile)

NOTE Users should add as many domains as needed to clearly identify the variable.

Variable name examples using “aero” and “thrust” include:

— `aeroXBodyForceCoefficient`

— `aeroXBodyForce_lbf`

— `thrustXBodyForce_lbf`

6.3.3 Specific Axis or Reference

This is the specific axis or reference used within the axis system (axis systems are defined in Section 5). If the axis system is included in the name, the specific axis or reference should also be included. For example

— (X, Y, Z), (N, E, D) or (U, V, W) for linear/translational motion,

— (Pitch, Roll, Yaw) or (P, Q, R) for angular motion.

Variable name examples:

`s_rollBodyRate_rs_1` where `Body` is the axis system and `roll` is the specific axis in the body axis system, roll indicating angular motion.

NOTE In this example `rollBodyRate` is designated as a state.

`UBodyTurbulenceVelocity_fs_1` where `Body` is the axis system and `U` is the specific axis in the body axis system, U indicating longitudinal translational motion.

`YGEVelocity_ms_1` where `GE` is the axis system and `Y` is the specific axis, also indicating translational motion.

`ZRunway22VelocityOfLeftWheelWRTTD_fs_1` where `Runway22` is the axis system (user defined) and `Z` is the specific axis, also indicating translational motion. `LeftWheel` is the point on the vehicle and `TD` (touchdown point) is the reference point.

`YBodyAccelOFPilotHead_ms_2` where `Body` is the axis system and `Y` is the specific axis, also indicating translational motion. Design pilot head location is the point on the vehicle.

Alternatively, the specific axis or reference can logically be a vector or an array. When vectors are used, a right handed triad in order (x, y, z) shall be used to avoid confusion.

Example as a vector:

`s_bodyAngularRate_rs_1[3]` where element 1 would be about the X axis (pitch), element 2 would be about the Y axis (roll) and element 3 would be about the Z axis (yaw)

6.3.4 Axis or Reference System

This is the axis or reference system to which the variable is referenced. Table 1 specifies the standard axis system abbreviations that should be used. If no axis system pertains to the variable or the core variable name needs no reference system to be unambiguous (ex. Airspeed) then this part of the variable name may be omitted.

6.3.4.1 Conventions Used

Earth fixed frames and local reference frames by convention use X, Y, Z, Pitch, Roll, and Yaw for axis references. Local reference frame (FE for example) origin and orientation may be user defined. Local reference frames are meant for runway, test range, target reference, navigational aid, etc. coordinate systems. Body fixed frames may use U, V, W, Pitch, Roll, and Yaw for axis references.

6.3.4.2 Variable Name Examples

The following variable names are provided as examples.

— `UBodyVelocity_fs_1 (or XBodyVelocity_fs_1, Body axis system)`

— s_XGEVelocity_fs_1 (in the case where the equations of motion are formulated such that the variable is a state, Geometric Earth axis system)

— XGEVelocity_fs_1 (in the case where the equations of motion are formulated such that the variable is not a state)

— UBodyVelocity_ms_1 (or XBodyVelocity_ms_1)

— VBodyVelocity_fs_1 (or YBodyVelocity_fs_1)

— S_XLLVelocity_fs_1 (Locally Level axis system)

— S_XFEVelocity_fs_1 (Flat Earth axis system)

— pitchBodyRate_rs_1 (or YBodyAngularRate_rs_1)

— rollBodyRate_rs_1 (or XBodyAngularRate_rs_1)

— yawBodyAccel_rs_2 (or ZBodyAngularAccel_rs_1)

Note that the standard encourages U, V, W, pitch, roll yaw for body frames in particular, since that is widely conventional. However, since the overall objective of the standard is to form a framework for clear communication between simulation facilities, the X, Y, Z convention is also acceptable. The appropriate core variable name shall be used to be clear whether the variable is a linear or angular variable.

6.3.5 Core Variable Name

This is the most specific (hence core) name for the variable. All variable names shall include this component of the name. Core variable name examples are as follows.

— velocity

— rate

— accel

— forceCoefficient

— turbulenceVelocity

— angleOfAttack

— angleOfSideslip

— cosineOfAngleOfSideslip

— thrust

— torque

— aileronDeflection (aileron could be considered a domain and deflection the core name)

The following variable names are provided as examples.

— s_rollBodyRate_rs_1

- `XBodyTurbulenceVelocity_fs_1`
- `ZGEVelocity_fs_1`
- `angleOfAttack_r`
- `angleOfSideslip_d`
- `cosineOfAngleOfSideslip`
- `aileronDeflection_d`

6.3.6 Reference Point or location on the vehicle

This component of the name is designed to clarify positions, velocities and accelerations and is normally omitted if the variable is not a position, velocity or acceleration. However, it may be used for any variable if desired. This component describes which point or object on the vehicle is being specified. "`Of`" is used to specify the point or object.

For those who prefer shorter variable names, the standard uses the convention that if the point or location on the vehicle is the center of mass (by convention, center of gravity, or CG) then the reference point may be omitted. However, use of "`OfCG`" is encouraged for clarity.

Reference points may be defined by the user and depend on the object the variable is describing.

Examples of reference points are as follows.

- `OfCG` (CG is the default, so "`OfCG`" may be omitted in any variable name)
- `OfPilot`
- `OfIMU`
- `OfSensor`
- `OfMRC` (for moment reference center)
- `OfPilotEye` (for the pilot eye point)
- `OfRadAlt` (for radar Altimeter)
- `OfTerrain`

The following variable names are provided as examples.

- `UBodyVelocityWRTWind_fs_1` (`OfCG` understood)
- `UBodyVelocityOfCGWRTWind_fs_1` (same meaning as above)
- `UBodyVelocityWRTInertial_fs_1` (inertial velocity of the CG along the X body axis)
- `heightOfCGWRTTerrain_f` (`CG` may be omitted since it is the default)
- `heightOfRadAltWRTTerrain_f`
- `heightOfTerrainWRTSurfaceReference_f`
- `XBodyPositionOfPilotEyeWRTCG_f`

— `longitudeRateOfIMUWRTWGS84_ds_1`

— `longitudeOfIMUWRTWGS84_d`

— `bodyAccelOfPilot_fs_2(3)`

6.3.7 External Reference Frame or Reference Point on the Reference Frame

The external reference frame is generally used in conjunction with “reference point or location on the vehicle” above. It is primarily used in variables describing position, velocities and accelerations. This component defines the external reference frame which the motion is relative to. If the reference frame is rotating or the variable is describing angular motion, this component should define a specific point in the reference frame. Stevens and Lewis (see Section 6.7) may be referred to for a more rigorous definition of “frames”.

The standard uses the convention “`WRT`” to define the frame component of the variable name. For those who prefer shorter variable names, the inertial frame is default, and therefore while use of “`WRTInertial`” is encouraged it may be omitted. Some examples of reference frames are as follows.

— `WRTInertial` (`WRTInertial` is the default and may be omitted)

— `WRTCG` (this is commonly used to clarify definitions of positions)

— `WRTMRC` (moment reference center)

— `WRTWGS84` (world geodetic system 84)

— `WRTTD` (ideal touchdown point)

— `WRTImpact` (the desired weapon impact point)

— `WRTWind` (the instantaneous wind velocity)

— `WRTMeanSL`

The following variable names are provided as examples.

— `UBodyVelocityWRTWind_fs_1` (`OfCG` understood)

— `UBodyVelocityOfCGWRTWind_fs_1` (same meaning as above)

— `UBodyVelocityWRTInertial_fs_1` (inertial velocity of the CG along the X body axis. `WRTInertial` may be omitted since Inertial is the default reference frame)

— `UBodyVelocity_fs_1` (inertial velocity of the CG along the X body axis, same meaning as above)

— `bodyPositionOfPilotEyeWRTCG_f(3)`

— `longitudeOfIMUWRTWGS84_d`

— `longitudeOfCGWRTWGS84_d`

— `bodyPositionOfPilotEyeWRTCG_f(3)`

— `bodyPositionOfCGWRTMRC(3)`

— `ZRunway22VelocityOfLeftWheelWRTTD_fs_1`

— `heightOfRunwayWRTMeanSL_f`

— `UBodyVelocityWRTWind_fs_1`

— `totalVelocityWRTGround_fs_1`

— `GEVelocity_ms_1(3)` (`WRTInertial` is omitted since inertial is the default)

6.3.8 Suffix — Units

The suffix is used to describe the units of the variable. The convention for the suffix is simple and is followed for all variables. This will allow the user, the programmer, and the reader of the code to check for homogeneity of the units and is self-documenting in this respect. Therefore, units shall be included in all variables except variables that are non-dimensional. Including units has the the added advantage of making this standard consistent and acceptable in countries utilizing the international system of units. For example, airspeed is just as acceptable as a standard both for the U.S. system of units and the International system of units.

The standard uses and analogy to exponential notation for the specification of units. A standard expression for feet cubed per second squared (for example) would be f^3s^{-2}. By eliminating the superscript we have f3s-2. However, a compiler would interpret this as subtracting 2 from f3s. Therefore instead of using the negative sign for exponents, we replace it with the underscore. Thus feet cubed per second squared can be represented as f3s_2. Feet per second is fs_1 and feet per second squared is fs_2. Every term in the denominator has an exponent. For example (r/s^2)/(f*lbf) would be expressed as rs_2f_1lbf_1.

Further examples are as follows.

— `trueAirspeed_fs_1` for feet per second (f/s)

— `trueAirspeed_ms_1`for meters per second (m/s)

— `trueAirspeed_nmih_1` for knots (nautical miles per hour)

This standard defines what the variable name for airspeed is, the user defines the units being used. The suffix shall be separated from the body of the variable name by an underscore. The standard unit notations are given in Table 2, SI units and standard abbreviations are included.

Table 2 — Abbreviations used to designate units in standard variable names

Unit	Abbreviation
Time	
hour	h
second	s
minute	min
millisecond	ms
Length	
inch	inch
foot	f
meter	m
nautical mile	nmi
statute mile	smi
kilometer	km
centimeter	cm
millimeter	mm
Force	

Unit	Abbreviation
pound force	lbf
Newton	N
kilogram force	kgf
Mass	
gram	g
kilogram	kg
pound mass	lbm
slug	slug
Plane Angle	
degrees (angular)	d
radians	r
revolution	rev
Temperature	
degrees Rankine	R
degrees Centigrade	C

Unit	Abbreviation
degrees Kelvin	K
Power, energy, work, heat	
British thermal unit	btu
erg	erg
calorie	Cal
joule	Jou
horsepower	Hp
Electrical	
volt direct current	vdc
volt alternating current	vac

Unit	Abbreviation
ampere	A
cycles	cyc
watt	watt
henry	hy
farad	fd
ohm	ohm
Other	
candela (luminous intensity)	cd
mole (amt. of substance)	mol

6.4 Additional Discussion

Very rarely, if ever, are all 8 components of a name used. In the case of `s_rollBodyRate_rs_1` the following 5 components were used:

— prefix [s] indicating that in this formulation of the equations of motion this variable is a state,

— specific axis or reference [`Roll`],

— axis or reference system [`body`],

— core name [`Rate`], and

— units suffix [`rs_1`].

In this case "variable source domain" was omitted because `s_rollBodyRate_rs_1` is a variable defined by the laws of physics and there cannot be a body rate from aerodynamics and a body rate from the moments produced by the engine. If however, the user wanted to have a multi-body simulation, logically the "variable source domain" could be used to discriminate between different elements of the body, or, perhaps more logically, an array or structure would be used to define different elements in a multi-body or flexible structure problem.

The "`Of`" and "`WRT`" were omitted because the variable is describing motion about ("`Of`") the CG and it is relative to ("`WRT`") the inertial frame of reference.

The intent is to provide clear communication when exchanging models, not to force the universal use of these variable names. `s_rollBodyRate_rs_1` is intended to be a clear, brief, unambiguous name for the variable.

6.4.1 Initial Condition Convention

A helpful convention that may be used is adding IC to the end of any variable name, but before the units, to designate that the variable is an initial condition specification. This can be added to virtually any variable, conceptually creating a constant, for example:

— `s_rollBodyRateIC_rs_1`

— `grossWeightIC_kg`

6.4.2 Discarded Conventions and Reasons

One convention considered was to have a prefix for simulation outputs as well as states and controls, but at the present time this has been discarded since the outputs required vary so widely, and there are typically an extremely large number of outputs. Practically speaking, every variable in the model including states, state derivatives and controls (inputs) could be considered an output.

Also considered was eliminating the suffix when the units were one of the standard set, but this concept was discarded since always having the units attached to the variable will help the programmer/engineer have consistent units when they are programming and reduce programming errors due to mixing of the units improperly. It also should noticeably reduce the software maintenance effort after initial development when another software engineer is trying to understand the code to make bug fixes, offer enhancements, or reuse the code.

6.4.3 Relationship with Markup Grammar, DAVE-ML

At present, this variable naming convention is intended to be realized using DAVE-ML grammar of XML (see Section 7). In DAVE-ML, the state/state derivative designation and the units are identified in separate components from the variable names. Thus, including these in a variable name encoded in DAVE-ML would be redundant. .

The best practice is to strip these components (the prefix and suffix) from the variable name when encoding to DAVE-ML, and reinsert them into the variable name if code or model data is generated from the DAVE-ML. Following this convention will have two advantages.

1) Since the DAVE-ML grammar can be used with any variables, for those variables that do not conform to the naming convention and therefore do not have state/state derivative designation or units, DAVE-ML encourages the inclusion of this information which is critical to clear documentation of a model.

2) It allows XML processors to adopt the convention of automatically striping and adding the prefix and suffix to the variable names.

6.5 Standard Variable Name Table Example

Using the conventions discussed above, a set of standard variable names has been created. These are presented in Annex A. An excerpt of Annex A is given in Table 3 for illustrative purposes.

Interpretation of the standard variable name annex is best given by example. Table 3 presents the standard variable defining the Roll Euler Angle, its axis system and positive sign convention (+ = RWD, or right wing down). Four name examples are provided.

— The short name, PHI – the short name is included to accommodate standard variable definitions in legacy compilers with name length restrictions

— One or more full names using the standard units convention — generally one full name with American convention units and one with SI units

NOTE Any suitable units may be used and no attempt is made to include all possible unites in Annex A.

— A description of the variable — when applicable the description should include the axis system in which the variable is defined

— The POSITIVE sign convention of the variable

— Minimum and maximum values of the variable, normally only specified for angles

In addition this example also illustrates the pitch and yaw Euler angles.

Since roll, pitch and yaw may also conveniently be expressed as a vector, the shaded area is the standard definition of the Euler angle vector. Again, `eulerAngle_r(3)` would be the standard vector using radians as the units and is fully compliant with the standard.

The standard allows use of any of the standard set of units.

Table 3 — Standard variable name table example

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Min Value	Max Value
Vehicle Positions and Angles						
ε	EUL(3)	`eulerAngle_d(3)` `eulerAngle_r(3)`	Vector of the roll, pitch, and yaw Euler angles comprised of the elements defined below. LL (locally level) frame.			
Φ	PHI	`rollEulerAngle_d` `rollEulerAngle_r`	Roll Euler Angle, LL frame.	RWD	-180,-π	180, π
θ	THET	`pitchEulerAngle_d` `pitchEulerAngle_r`	Pitch Euler Angle, LL frame	ANU	-90, -π/2	90, π/2
ψ	PSI	`yawEulerAngle_d` `yawEulerAngle_r`	Yaw Euler Angle, LL frame	ANR	-180,-π	180, π

6.6 Summary

While it is strongly recommended that this naming convention be followed for all future variables, the real key to a standard variable name is not the name, but the definition of the name. To exchange information between two or more organizations, the most important factor is not whether a variable is named airspeed or as, but what is the precise, unambiguous definition of the variable (true, indicated, or calibrated airspeed?, etc.), including units and axis system.

Using the standard variable name simply provides a common language and set of definitions within which to facilitate transfer of the model.

The simulation community is encouraged to propose additional standard variable names. Annex C describes the web site used to support this standard. There is an appropriate URL or email address for submitting additional names or for recommending clarification of existing names.

6.7 References

Stevens, Brian L., and Lewis, Frank L., **Aircraft Control and Simulation, Second Edition**. ISBN 978-0-471-37145-8, 2004, New York, J. Wiley and Sons, 2003, p. 3.

7 Standard Simulation Function Table Data Format and XML Implementation of the Standard: DAVE-ML

7.1 Purpose

This section explains the data requirements which a standard function table format must be able to satisfy. It includes the content of the information contained in the table and configuration management of the data in the table. As you will see, the definition of the table format includes data for all these components.

This document also discusses conceptually how the data table should be accessed in an executable program.

The standard is implemented in XML as specified by DAVE-ML, Annex B. Annex C provides links to example programs for loading and looking up data in the XML standard.

7.2 Philosophy

Probably the most immediate benefit of the standard to the simulation discipline is one that defines formats for the interchange of tabular data. Tabular data is used almost universally for non-linear function generation of aerodynamic, engine, atmospheric, and many other model parameters. The simple interchange of such data can greatly improve efficiency in the simulation community.

Most simulation developers and users have addressed this issue locally. In many simulation communities, a family of tools has been built around existing local function table standards. Thus, the intent of this standard is not to obsolete these local standards, but rather to define a format for communication which will allow each site to develop a single format converter to and from their local format. This is an exchange standard. It is hoped that this standard will eventually be adopted for local use as well, but that is not required for the standard to succeed.

7.3 Design Objective

The design objectives of the standard data table format were first and foremost to make a data format that would include all the information about real multi-dimensional data, not just the data values. This notably is the fact that, in the general case of the independent variables for a multi-dimensional table, the independent variables have different numbers of breakpoints, different breakpoints, and different valid ranges. An equally important design objective was to allow the table to contain information on where the data points come from (provenance, via reference), and a confidence interval for the data. Confidence intervals can be used for Monte Carlo simulations and to mathematically combine two different estimates of the same parameter at the same point. Therefore, confidence statistics are extremely valuable when attempting to update a data set (however the user must be careful as not all confidence intervals are equivalent, or even meaningful). Additionally, the table has to be easy to read by the computer and the human being, and be self-documenting as much as possible.

7.4 Standard Function Table Data — An Illustrative Example

Figure 1 presents a fairly standard three-dimensional set of data as is typical of aerodynamic data from flight test or from a wind tunnel. In the example given, lift coefficient is a function of angle-of-attack, Mach number, and a control position. More generally stated, a function output (dependent variable), CLALFA is dependent on three inputs (independent variables), `angleOfAttack_d`, mach, and `avgElevatorDeflection_d`.

Close examination of the example data given will reveal the following characteristics.

1) The number of breakpoints of the independent variables varies for each independent variable. Not only are there a different number of angle-of-attack (`angleOfAttack_d`) breakpoints, but also a different number of Mach number (`mach`) and control position (`avgElevatorDeflection_d`) breakpoints. This standard defines this as an ungridded table. A gridded table is one where the number of breakpoints of a specific independent variable are the same for each of the other independent variables. For example, there are the same number of Mach breakpoints for each angle of attack breakpoint.

2) The values (breakpoints) of the independent variables are different. Again, an ungridded table.

3) The valid ranges of the independent variables are different (ungridded table).

4) The above three differences are not consistent for all data. For example, in the sample table the `angleOfAttack_d`, breakpoints for mach = 0.6 and mach = 0.7 and for delta `SavgElevatorDeflection_d` = -5 are identical.

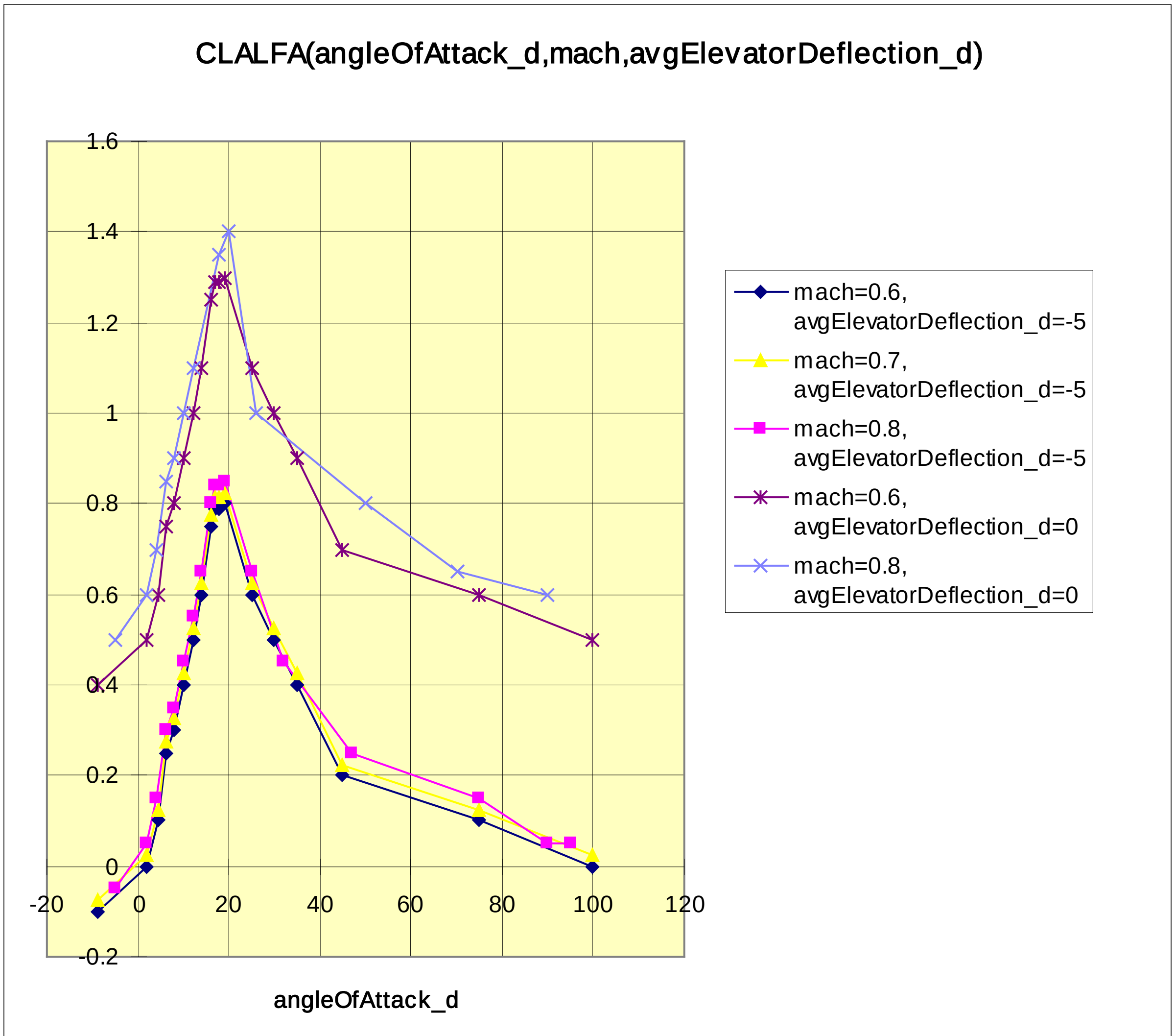


Figure 1 — An illustration of a 3 dimensional function table, CLALFA (`angleOfAttack_d`, `mach`, `avgElevatorDeflection_d`)

For function data there is other information that is of significant importance to the user, without which the data is not very useful. In general this information is as follows.

— Where did the data come from? For example what report?

— How is it defined? For example, is this at a specific altitude? What configuration is it for?

— What are the engineering units of the output (the dependent variable) and the independent variables?

— What is the sign convention of the independent and dependent variables? For example, is the control position positive trailing edge up or trailing edge down? Exactly which control surface is it?

— Who created the table? Not where the data came from, but what person decided that this was the correct data for this table?

— How has it been modified and for what reason?

— How accurate is the data estimated to be? Or, mathematically what is the confidence interval of the data?

— By what method is the data intended to be interpolated? For example, linear interpolation or bi- cubic spline interpolation?

— By what method is the data intended to be extrapolated for data with different ranges?

The standard data format has data elements that contain all of the above information. It has been implemented in XML as seven major elements and is discussed in detail in Annex B. An introduction and overview will be provided here.

Additionally, DAVE-ML also includes the ability to automate static checks of the function data to allow spot checking of the function after it has been exchanged.

7.5 DAVE-ML Major Elements (reference Annex B)

These major elements are provided in the same order as they must be in the XML files. In general, most attributes and sub-elements are optional. In fact, only the *fileHeader* and *variableDef* major elements are required.

The logical flow of information is such that the lower major elements refer upward to information previously defined, in general, so that information (breakpoints, data points, provenance, etc.) that is re-used in more than one function does not need to be repeated.

1) *fileHeader* — the fileHeader contains the file provenance (who created the file and how to contact that person or team), all references and overall description about all the functions in this particular file. The provenance of each particular function refers to the fileHeader.

2) *variableDef* — defines the signals used (variables) to generate the functions, at a minimum, the independent variables (inputs) and the dependent variables (outputs). Additionally, it includes the definition of any intermediate variables used to generate the functions, and defines any calculations that are to be performed (defined as MathML).

3) *breakpointDef* — here, all the breakpoints, or independent variable data points, for gridded tables are defined. One set of breakpoints may be used by many functions. This section does not apply to ungridded tables. They contain their breakpoints within the *ungriddedTableDef* major element. There may be a provenance for the breakpoints, which again may refer to the *fileHeader*.

4) *griddedTableDef* — contains the data points of the function. These data points use the breakpoints defined in the breakpointDef major element. The provenance of each set of data points may be explicitly defined here, and may refer to documents defined in the *fileHeader*.

5) *ungriddedTableDef* — contains the breakpoints and the data points of the ungridded tables. These are specified as sets of breakpoints and data points together and do not refer to the *breakPointDef* major element. As in *griddedTableDef*, the provenance of each set of data points may be explicitly defined here, and may refer to documents defined in the *fileHeader*.

6) *function* — combines the breakpoints with the data points, and defines which independent variables are used as inputs to the functions. This element also includes definition of how the function should be interpolated and extrapolated, and is the definitive element to include provenance on the particular function (where did the data for this function come, who decided this set of data points would be used for this function, etc.). The nonlinear function definition is complete at this point.

7) *checkData* — contains a set of static check cases to verify the functions. It includes an optional tolerance on the outputs. If the checkData element is used, it must include check cases for all outputs in the file (it cannot check some functions and not others).

Annex B contains a detailed description and examples of the data element definitions of the DAVE-ML function table standard. Appendix A of Annex B provides detailed XML element references and descriptions.

7.6 A Simple DAVE-ML Example

The easiest way to understand the standard is through an example. Annex B contains many more examples of the DAVE-ML implementation of the standard.

A simple one dimensional aero table is provided as an example, in this case pitching moment coefficient as a function of angle of attack, Table 4 and Figure 2.

Table 4 — A simple function

`angleOfAttack_d`	0	18	19	20	22	23	25	27	9
`cm(angleOfAttack_d)`	0.1	-0.1	-0.09	-0.08	-0.05	-0.05	-0.07	-0.15	-0

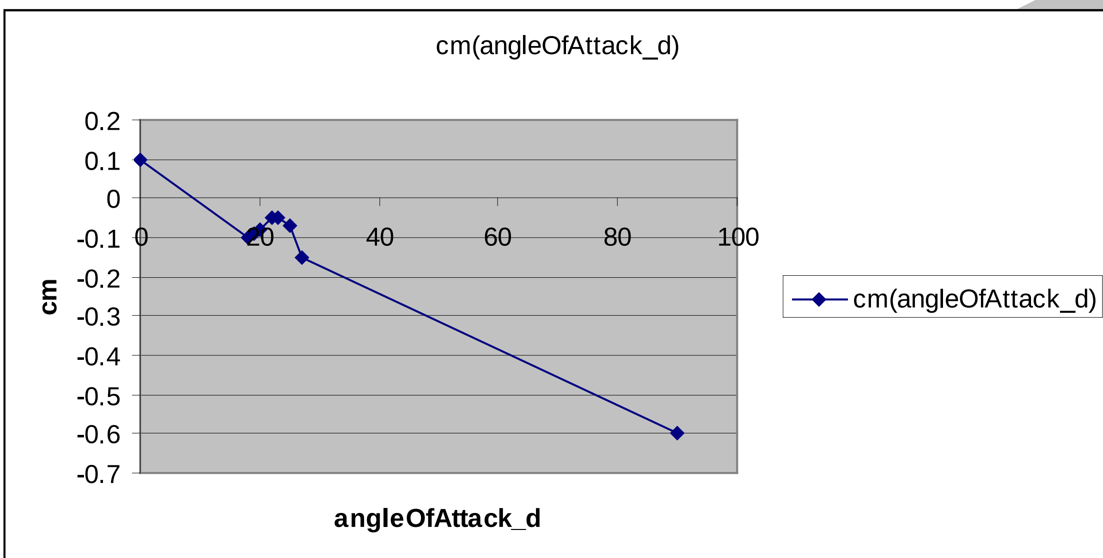


Figure 2 — The $C_m(\alpha)$ function — a simple one dimensional gridded function

The DAVE-ML implementation for this function could be as follows.

```
CmaExample.dml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE DAVEfunc PUBLIC "-//NASA//DTD for Flight Dynamic Models - Functions
2.0//EN" "DAVEfunc.dtd">
<DAVEfunc>

<!--             ==========================                 -->
<!--=============  File Header Components  ===============  -->
<!--             ==========================                 -->
<fileHeader>

  <!--  This is an example of the file header components of the
        derivative of Cm as a function of angle of attack.  It must
        remembered that all fileheader components of all functions
        in the file must be grouped together into one file header
        area.

         Also note that there is not much information in this header,
         Mainly because it is mean to be a simple example.  In
         reality, probably the most important information is the
```

```
         author, the reference and the modification record, because
         these data describe where the data came from and if it has
         been changed (and how).  See annex B for more complete
         examples.
 -->

  <author name="Bruce Hildreth" org="SAIC" email="bruce.hildreth@saic.com"/>
  <fileCreationDate date="2006-03-18"/>
  <description>
    This is made up data to use as an example of a simple gridded function.
  </description>
  <reference refID="BLHRpt1" author="Joe Smith"
           title="A Generic Aircraft Simulation Model (does not really
exist)"
           accession="ISBN 1-2345-678-9" date="2004-01-01"/>

 <!-- no modifications so far, so we don't need a modificationRecord yet -->

</fileHeader>

<!--             ==================================                -->
<!--=============  Variable Definition Components  ===============  -->
<!--             ==================================                -->


 <!-- Input variable -->

 <variableDef name="Angle of attack" varID="angleOfAttack_d" units="deg" >
   <isStdAIAA/> <!-- Indicates that this variable is a standard
                     variable, which is why the author omitted
                     description and sign convention
                     and any other info. (it certainly could
                     be included here)                              -->
 </variableDef>

 <!-- Output (function value) -->

 <variableDef name="Pitching moment coefficient due to angle of attack"
              varID="CmAlfa" units="nondimensional" sign="+ANU">
   <description>
     The derivative of total pitching moment with respect to
     angle of attack.
  </description>
 </variableDef>

 <!--             =============================                -->
 <!--=============  Breakpoint Definition Set  ===============  -->
 <!--             =============================                -->

 <breakpointDef bpID="angleOfAttack_d_bp1">

   <!--
     Note that the bpID can be any name for the breakpoints.  The
     author here chose to use a name related to the independent
```

```
    variable that is expected to be used to look up the function.  In
    fact, if this set of breakpoints were shared by many functions
    and different independent variables would be used to look up the
    function, then the bpID of "angleOfAttack_d_BP1" would be
    misleading and a more generic name like "AOA" would probably be
    better.
  -->

  <description>
    Angle of attack breakpoint set for CmAlfa, CdAlfa, and ClAlfa
  </description>

  <bpVals>    <!--  Always comma separated values  -->
    0, 18, 19, 20, 22, 23, 25, 27, 90
  </bpVals>

</breakpointDef>

<!--             =============================                --> 
<!--=============  Gridded Table Definition   =============== -->
<!--             =============================                -->

<griddedTableDef gtID="CmAlfa_Table1">
  <description>
    The derivate of Cm wrt fuselage AOA in degrees
  </description>

  <provenance>
    <author name="Jake Smith" org="AlCorp"/>
    <functionCreationDate date="2006-12-31"/>
    <documentRef refID="BLHRpt1" />  <!--  This points back to the Header,
                                          which provides the information
                                          about BLHRpt1. -->
  </provenance>

  <breakpointRefs>
    <bpRef bpID="angleOfAttack_d_bp1" />
  </breakpointRefs>

  <uncertainty effect="percentage">
    <normalPDF numSigmas="3">
      <bounds>12</bounds>
    </normalPDF>
    <!--  This means that the 3 sigma confidence is +-12% on the Data. -->
  </uncertainty>

  <dataTable>     <!--  Always comma separated values -->
    0.1,-0.1,-0.09, -.08, -0.05, -0.05, -0.07, -0.15, -0.6
  </dataTable>

</griddedTableDef>

<!--             ========================                -->
<!--=============  Function Definition   =============== -->
```

```
<!--             ========================                --> 

<!-- The function definition ties together input and output variables
     to table definitions. This allows a level of abstraction such
     that the table, with it's breakpoint definitions, can be reused
     by several functions (such as left and right aileron or multiple
     thruster effect tables).
-->

<function name="Cm_alpha_func">
  <description>
    Variation of pitching moment coefficient with angle of attack (example)
  </description>
  <independentVarRef varID="angleOfAttack_d"/>
  <dependentVarRef varID="CmAlfa"/>
  <functionDefn>
    <griddedTableRef gtID="CmAlfa_Table1"/>
  </functionDefn>
</function>

<!--             =====================                --> 
<!--=============  Check Data Cases   =============== --> 
<!--             =====================                --> 

<!-- Checkcase data provides automatic verification of the model by
     specifying the tolerance in output values for a given set of
     input values. One 'staticShot' is required per input/output
     mapping; in this case for a single input, single output model,
     we have a single input signal and a single output signal in each
     test point.
-->

<checkData>
  <staticShot name="case 1">
    <checkInputs>
      <signal>
      <varID>angleOfAttack_d</varID>
      <signalValue> 0.</signalValue>
      </signal>
    </checkInputs>
    <checkOutputs>
      <signal>
      <varID>CmAlfa</varID>
      <signalValue>0.01</signalValue>
      <tol>0.00001</tol>
      </signal>
    </checkOutputs>
  </staticShot>
  <staticShot name="case 2">
    <checkInputs>
      <signal>
      <varID>angleOfAttack_d</varID>
      <signalValue> 5.</signalValue>
      </signal>
```

```
    </checkInputs>
    <checkOutputs>
      <signal>
      <varID>CmAlfa</varID>
      <signalValue>0.04444</signalValue>
      <tol>0.00001</tol>
      </signal>
    </checkOutputs>
  </staticShot>
  <staticShot name="case 3">
    <checkInputs>
      <signal>
      <varID>angleOfAttack_d</varID>
      <signalValue>10.</signalValue>
      </signal>
    </checkInputs>
    <checkOutputs>
      <signal>
      <varID>CmAlfa</varID>
      <signalValue>-0.01111</signalValue>
      <tol>0.00001</tol>
      </signal>
    </checkOutputs>
  </staticShot>
  <staticShot name="case 4">
    <checkInputs>
      <signal>
      <varID>angleOfAttack_d</varID>
      <signalValue>15.</signalValue>
      </signal>
    </checkInputs>
    <checkOutputs>
      <signal>
      <varID>CmAlfa</varID>
      <signalValue>-0.06667</signalValue>
      <tol>0.00001</tol>
      </signal>
    </checkOutputs>
  </staticShot>
  <staticShot name="case 5">
    <checkInputs>
      <signal>
      <varID>angleOfAttack_d</varID>
      <signalValue>20.</signalValue>
      </signal>
    </checkInputs>
    <checkOutputs>
      <signal>
      <varID>CmAlfa</varID>
      <signalValue>-0.08</signalValue>
      <tol>0.00001</tol>
      </signal>
    </checkOutputs>
  </staticShot>
```

```
   <staticShot name="case 6">
     <checkInputs>
       <signal>
       <varID>angleOfAttack_d</varID>
       <signalValue>25.</signalValue>
       </signal>
     </checkInputs>
     <checkOutputs>
       <signal>
       <varID>CmAlfa</varID>
       <signalValue>-0.07</signalValue>
       <tol>0.00001</tol>
       </signal>
     </checkOutputs>
   </staticShot>
   <staticShot name="case 7">
     <checkInputs>
       <signal>
       <varID>angleOfAttack_d</varID>
       <signalValue>50.</signalValue>
       </signal>
     </checkInputs>
     <checkOutputs>
       <signal>
       <varID>CmAlfa</varID>
       <signalValue>-0.31429</signalValue>
       <tol>0.00001</tol>
       </signal>
     </checkOutputs>
   </staticShot>
 </checkData>
</DAVEfunc>
```


While the above seems incredibly long for a function with only 9 data points, keep in mind it also includes many instructional comments and optional, but very important information, such as units and where the data came from (provenance). Also, a very large complex function would only be expanded by the additional data points. The definitions and provenance information included with the function would probably not change much.

In the minimum, the same data can be represented as shown.

shorter_cma_example.dml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?> <!DOCTYPE DAVEfunc
PUBLIC "-//NASA//DTD for Flight Dynamic Models - Functions 2.0//EN"
"DAVEfunc.dtd"> <DAVEfunc>
     <fileHeader>
         <author name="Bruce Hildreth" org="SAIC"/>
         <fileCreationDate date="2006-03-18"/>
     </fileHeader>
     <variableDef name="Angle of attack" varID="angleOfAttack_d"
units=""/>
     <variableDef name="CMalpha" varID="CmAlfa" units=""/>
     <breakpointDef bpID="angleOfAttack_d_bp1">
```

```
        <bpVals> 0, 18, 19, 20, 22, 23, 25, 27, 90 </bpVals>
    </breakpointDef>
    <griddedTableDef gtID="CmAlfa_Table1">
        <breakpointRefs>
            <bpRef bpID="angleOfAttack_d_bp1"/>
        </breakpointRefs>
        <dataTable> 0.1,-0.1,-0.09, -.08, -0.05, -0.05, -0.07, -0.15, -0.6
</dataTable>
    </griddedTableDef>
    <function name="Cm_alpha_func">
        <independentVarRef varID="angleOfAttack_d"/>
        <dependentVarRef varID="CmAlfa"/>
        <functionDefn>
            <griddedTableRef gtID="CmAlfa_Table1"/>
        </functionDefn>
    </function>
</DAVEfunc>
```

7.7 Summary

The DAVE-ML embodiment of the standard truly enables nearly effortless transfer of simulation aerodynamics models between simulation facilities or architectures. The addition of the Math-ML allows the formulation of algebraic equations, aero or engine model coefficient buildup equations, for example, to be included as data in the model. DAVE-ML is also suitable for use of transfer of tabular functions and supporting algebraic equations for any type of data, not just simulation models.

While the above paragraphs explained the concepts implemented in DAVE-ML, Annex B is the authoritative normal for this standard. It provides much more detail and examples on how to easily build a DAVE-ML compliant simulation. Annex C provides reference to the DAVE-ML web site that includes tools to facilitate using DAVE-ML based models in you particular simulation.

8 Future Work

The AIAA Modeling and Simulation Technical Committee plans to continue its efforts in facilitation of the exchange of simulations and models throughout the user community. Comments and suggestions on this expansion are welcomed on the simulation standards discussion group. Visit http://daveml.nasa.gov for submittal information. The following sections describe the two tasks of primary interest.

8.1 Time History Information

The immediate task that is being pursued is the transfer of validation data between facilities. This is for the purpose of sending time response validation data when a model is exchanged.

The approach being taken is to adopt a flight test data standard. This has the advantage of using an existing standard and facilitating the use of flight test data to validate a simulation. Lockheed Martin has an existing internal standard that they have released for use by the community. It is implemented in hierarchal data format (HDF) and has been adopted by the JSF community and other programs. It is the Modeling and Simulation Technical Committees intent to adopt this for the transfer of simulation validation data. Some work will be required to define the data elements that are required for the validation of a simulation. This is expected to be a subset of the data elements that comprise flight test data.

8.2 Dynamic Element Specification

The addition of the specification of dynamics (e.g. continuous and discrete states) is being considered to expand the scope of the standard. This expansion would allow more of the domain of a flight vehicle model (flight controls as a good example) to be exchanged in a non-proprietary, facility-neutral way.

9 Conclusion

This is a standard for the purpose of facilitating the exchange of simulation models between users. This purpose cannot be emphasized enough. It is not meant to enforce any standard simulation architecture. DAVE-ML provides the mechanism for exchange of the modeling data and equations; the standard variables and axis systems provide a common language to facilitate effective communication. The standard is also valuable for documenting a model, since the names and axis system definitions are clearly documented for the user.

A model can be DAVE-ML compliant without using any standard names or axis systems, but the exchange of such a model between users will be more difficult, since clear definitions will have to be exchanged also.

It is the earnest desire of the authors of this standard that the user community will employ the current standard for aerodynamic models, continue to suggest improvements to the standard, and develop tools to enhance the standard. Visit http://daveml.nasa.gov for information on how to be part of this effort and/or submit change or improvement recommendations.

Annex A Standard Variable Names (Normative)

A.1 General

The table in this annex is meant to contain simulation variables that are independent of the particular vehicle type being simulated. These variables are tailored towards aircraft simulation. Visit http://DaveML.nasa.gov to suggest additional variables or changes to the existing list

A.2 Table Explanation

Interpretation of the standard variable name table is best given by example. In general the table has 7 columns. These are described below using the `rollEulerAngle` as an example:

1) The symbol for that variable, Φ

2) The short name, PHI

3) One of more full names using the standard units conventions — generally, one full name with American convention units and one with SI units.

   NOTE Any suitable units may be used. In the example for `rollEulerAngle` both the _d for degrees and the `_r` for radians are given. The "Full Variable Name" column does not necessarily provide all acceptable units for each variable.

4) A description of the variable, if applicable should always specify the axis system.

5) The POSITIVE sign convention of the variable — RWD indicates that positive `rollEulerAngle` is right wing down

6) Minimum value, normally only specified for angles

7) Maximum values of the variable, normally only specified for angles

This example also illustrates the pitch and yaw Euler angles.

Some variables may be used to represent variables referenced to more than one axis system. In this case the axis system is specified as xx and any axis system reference (refer to the body of this standard) may be substituted for the xx. For example, `YxxVelocity_fs_1` may represent:

— `YEIVelocity_fs_1` for the EI axis system - Earth centered Inertial (also know as geocentric inertial) axis system

— `YECEFVelocity_fs_1` for the ECEF axis system - Earth centered Earth Fixed (also known as Geocentric Earth [GE] axis system, `YGEVelocity_fs_1` is the same as `YECEFVelocity_fs_1`)

— `YVOVelocity_fs_1` for the VO axis system - Vehicle carried, Orbit defined axis system

Since roll, pitch and yaw may also conveniently be expressed as a vector, the shaded area is the standard definition of the Euler angle vector. Again, `eulerAngle_r[3]` would be the standard vector using radians as the units and is fully compliant with the standard.

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Min Value	Max Value
ε	EUL[3]	`eulerAngle_d[3]` `eulerAngle_r[3]`	Vector of the roll, pitch, and yaw Euler angles comprised of the elements defined below. LL (locally level) frame.			
Φ	PHI	`rollEulerAngle_d` `rollEulerAngle_r`	Roll Euler Angle, LL frame.	RWD	-180,-π	180, π
θ	THET	`pitchEulerAngle_d` `pitchEulerAngle_r`	Pitch Euler Angle, LL frame	ANU	-90, -π/2	90, π/2
ψ	PSI	`yawEulerAngle_d` `yawEulerAngle_r`	Yaw Euler Angle, LL frame	ANR	-180,-π	180, π

The variable name table below does not specify which variables are states, state derivatives, inputs or initial conditions. These specifications may be added to any appropriate variable. See the body of this standard.

A.3 Standard Variable Name Tables

Table A.1 — Vehicle Positions and Angles

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Intial Value	Min Value	Max Value
$\underline{\varepsilon}$	EUL	eulerAngle_d[3] eulerAngle_r[3]	Vector of the roll, pitch, and yaw Euler angles defined below. LL (locally level) frame.				
Φ	PHI	rollEulerAngle_d rollEulerAngle_r	Roll Euler Angle, LL frame.	RWD	From vehicle trim	-180	180
θ	THET	pitchEulerAngle_d pitchEulerAngle_r	Pitch Euler Angle, LL frame	ANU	From vehicle trim	-90	90
ψ	PSI	yawEulerAngle_d yawEulerAngle_r	Yaw Euler Angle, LL frame	ANR	From vehicle trim	-180	180
sin Φ	SPHI	rollEulerAngleSine	Sine Of Euler Roll Angle	RWD		-1.0	1.0
cos Φ	CPHI	rollEulerAngleCosine	Cosine Of Euler Roll Angle	RWD		-1.0	1.0
sin θ	STHT	pitchEulerAngleSine	Sine Of Euler Pitch Angle	ANU		-1.0	1.0
cos θ	CTHT	pitchEulerAngleCosine	Cosine Of Euler Pitch Angle	ANU		-1.0	1.0
sin ψ	SPSI	yawEulerAngleSine	Sine Of Euler Yaw Angle	ANR		-1.0	1.0
cos ψ	CPSI	yawEulerAngleCosine	Cosine Of Euler Yaw Angle	ANR		-1.0	1.0
$\underline{T}_{FE/B}$		FEToBodyT[3,3]	The FE to Body transformation matrix composed of the elements defined below				
$T_{FE/B}(1,1)$	T11	FEToBodyT11	CTHT*CPSI (FE To B) axis transformation element				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Intial Value	Min Value	Max Value
$T_{FE/B}(2,1)$	T21	FEToBodyT21	SPHI*STHT*CPSI - CPHI*SPSI (FE To B) axis transformation element				
$T_{FE/B}(3,1)$	T31	FEToBodyT31	CPHI*STHT*CPSI + SPHI*SPSI (FE to B) axis transformation element				
$T_{FE/B}(1,2)$	T12	FEToBodyT12	CTHT*SPSI (FE to B) axis transformation element				
$T_{FE/B}(2,2)$	T22	FEToBodyT22	SPHI*STHT*SPSI + CPHI*CPSI (FE to B) axis transformation element				
$T_{FE/B}(3,2)$	T32	FEToBodyT32	CPHI*STHT*SPSI - SPHI*CPSI (FE to B) axis transformation element				
$T_{FE/B}(1,3)$	T13	FEToBodyT13	-STHT (FE to B) axis transformation element				
$T_{FE/B}(2,3)$	T23	FEToBodyT23	SPHI*CTHT (FE to B) axis transformation element				
$T_{FE/B}(3,3)$	T33	FEToBodyT33	CPHI*CTHT (FE to B) axis transformation element				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Intial Value	Min Value	Max Value
γ_V	GAMV	flightPathAngle_r flightPathAngle_d	Flight Path Angle Above Horizon	ANU		-p/2 -90	p/2 90
γ_H	GAMH	flightPathAzimuth_r flightPathAzimuth_d	Flight Path Angle In Horizon Plane, from North	CWFN		-p -180	p 180
h	ALT	altitudeMSL_f altitudeMSL_m	Geometric altitude of vehicle altimeter above Mean Sea Level	UP			
	XLON	longitudeWRTzzz_r longitudeWRTzzz_d	Longitude of Vehicle CG with respect to the zzz reference frame.	WEST			
	XLAT	latitudeWRTzzz_r latitudeWRTzzz_d	Latitude of Vehicle CG with respect to the zzz reference frame.	NORTH			
	XLONIMU	longitudeOfIMUWRTzzz_r longitudeOfIMUWRTzzz_d	Longitude of Vehicle IMU with respect to the zzz reference frame.	NORTH			
	XLATIMU	latitudeOfIMUWRTzzz_r latitudeOfIMUWRTzzz_d	Latitude of Vehicle IMU with respect to the zzz reference frame.	NORTH			
EXAMPLE							
		longitudeOfIMUWRTWGS84_d latitudeOfIMUWRTWGS84_d	Longitude and latitude of the vehicle IMU in the World Grid System 1984 reference frame				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Intial Value	Min Value	Max Value
	HGT_RWY	runwayHeightAboveSL_ft runwayHeightAboveSL_m	Height Of Runway w.r.t. mean Sea Level	Above			
		General Definition xxPositionOfyyyWRTzzz_f[3] xxPositionOfyyyWRTzzz_m[3] For Example: xxPosition_f[3] is the same as xxPositionOfCG_f[3]	General Definition Vector of positions of yyy with respect to zzz (a user defined reference point or frame) in the xx axis system. The lengths of xx, yyy, zzz are not restricted to 2 and 3 characters respectively. The axis system, xx, must always be defined. If the yyy is not defined the definition defaults to the vehicle cg. If the zzz is not defined the reference point defaults to the origin of the axis system. Comprised of the three components as defined below.				
	XCG	XxxPositionOfyyyWRTzzz_f XxxPositionOfyyyWRTzzz_m or XxxPosition_f	X position of yyy with respect to zzz (a user defined reference point) in the xx axis system. Defaults to th CG and origin of the axis system.	(yyy –zzz)			
	YCG	YxxPositionOfyyyWRTzzz_f YxxPositionOfyyyWRTzzz_m or YxxPosition_f	Y position of yyy with respect to zzz (a user defined reference point) in the xx axis system. Defaults to th CG and origin of the axis system.	(yyy –zzz)			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Intial Value	Min Value	Max Value
	ZCG	ZxxPositionOfyyyWRTzzz_f ZxxPositionOfyyyWRTzzz_m or ZXxxPosition_f	Z position of yyy with respect to zzz (a user defined reference point) in the xx axis system. Defaults to the CG and origin of the axis system.	(yyy –zzz)			
		General Definition xxPositionOfMRCWRTzzz_f[3] xxPositionOfMRCWRTzzz_m[3] Example xxPositionOfMRC_f[3]	General Definition Vector of positions of the moment reference center (MRC) with respect to zzz (a user defined reference point) in the xx axis system. The lengths of xx, yyy, zzz are not restricted to 2 and 3 characters respectively. The moment reference center is sometimes more convenient to locate a vehicle since the moment reference center is fixed in the vehicle, but the CG moves. zzz may be defaulted to the origin of the axis system. Comprised of the three components as defined below.				
	XREF	XxxPositionOfMRCWRTzzz_f XxxPositionOfMRCWRTzzz_m	X position of the moment reference center (MRC) with respect to zzz in the xx axis system.	XxxPositionOfMRCWRTzzz_f XxxPositionOfMRCWRTzzz_m			
	YREF	YxxPositionOfMRCWRTzzz_f YxxPositionOfMRCWRTzzz_m	Y position of the moment reference center (MRC) with respect to zzz in the xx axis system.	YxxPositionOfMRCWRTzzz_f YxxPositionOfMRCWRTzzz_m			
	ZREF	ZxxPositionOfMRCWRTzzz_f ZxxPositionOfMRCWRTzzz_m	Z position of the moment reference center (MRC) with respect to zzz in the xx axis system.	ZxxPositionOfMRCWRTzzz_f ZxxPositionOfMRCWRTzzz_m			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Intial Value	Min Value	Max Value
		bodyPositionOfPilotEyeWRTCG_f[3] bodyPositionOfPilotEyeWRTCG_f[3]	Vector of positions of the pilot's eye with respect to the CG in the body axis system. Comprised of the three components as defined below.				
	XPLT2CG	XBodyPositionOfPilotEyeWRTCG_f XBodyPositionOfPilotEyeWRTCG_f	X position of pilot eye point w.r.t. CG, in the body axis system	Eye FWD of CG			
	YPLT2CG	YBodyPositionOfPilotEyeWRTCG_f YBodyPositionOfPilotEyeWRTCG_f	Y position of pilot eye point w.r.t. CG, in the body axis system	Eye Right of the CG			
	ZPLT2CG	ZbodyPositionOfPilotEyeWRTCG_f ZbodyPositionOfPilotEyeWRTCG_f	Z position of pilot eye point w.r.t. CG, in the body axis system	Eye below CG			
		EXAMPLE					
		Runway22Position_f[3] indicates position of the CG with respect to the origin of the Runway22 axis system Runway22PositionOfFwdLeftMainWheelWRTTD_f[3] indicates position of the forward left main wheel with respect to the touchdown point in the Runway 22 axis system NOTE All are user defined	Vector of positions of the vehicle CG relative to the Runway 22 (a user defined axis system) touchdown reference point. Comprised of the three components as defined below.				
	XCGTD	XRunway22PositionOfCGWRTTD _f XRunway22PositionOfCGWRTTDD_m	CG X-position w.r.t. Runway touchdown point in the specified (Runway22) axis system.	CG Down the runway from the reference point			
	YCGTD	YRunway22PositionOfCGWRTTD _f YRunway22PositionOfCGWRTTD_m	CG Y-position w.r.t. Runway touchdown point in the specified (Runway22) axis system.	CG to the right of the reference point			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Intial Value	Min Value	Max Value
	ZCGTD	ZRunway22PositionOfCGWRTTD _f ZRunway22PositionOfCGWRTTD _m	CG Z-position w.r.t. Runway touchdown point in the specified (Runway22) axis system(this variable is normally negative)	CG below the TD point			
	RE	smoothEarthRadius_f smoothEarthRadius_m	Radius of Earth (center to smooth surface which is mean sea level), round earth model or oblate spheroid under the aircraft.				
	RALT	heightOfCGWRTTerrain_f heightOfCGWRTTerrain _m	Height of the aircraft CG above the terrain	NSG			
	HTERRAIN	heightOfTerrainWRTSurfaceReference_f heightOfTerrainWRTSurfaceReference_m	Height of the terrain under the A/C CG. It is the terrain height above the smooth surface of of the earth, regardless whether a flat, round or oblate spheroid model is used.				

Table A.2 — Vehicle velocities and angular rates

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\underline{\omega}_B$	OMB	bodyAngularRate_rs_1[3] bodyAngularRate_ds_1[3]	Vector of body axis angular rates comprised of the three components as defined below. Motion is always wrt the inertial frame unless otherwise specified.				
p_B	PB	rollBodyRate_rs_1 rollBodyRate_ds_1	Vehicle roll velocity, body axis system	RWD			
q_B	QB	pitchBodyRate_rs_1 pitchBodyRate_ds_1	Vehicle pitch velocity, body axis system	ANU			
r_B	RB	yawBodyRate_rs_1 yawBodyRate_ds_1	Vehicle yaw velocity, body axis system	ANR			
$\dot{\underline{\xi}}$	EULD	eulerAngleRate_ds_1[3] eulerAngleRate_rs_1[3]	Vector of the roll, pitch, and yaw Euler angle rates defined below. LL (locally level) axis system				
$\dot{\phi}$	PHID	rollEulerAngleRate_rs_1	Euler roll rate, LL axis system	RWD			
$\dot{\theta}$	THETD	pitchEulerAngleRate_rs_1	Euler pitch rate, LL axis system	ANU			
$\dot{\psi}$	PSID	yawEulerAngleRate_rs_1	Euler yaw rate, LL axis system	ANR			
		General Definition XxxVelocityOfyyyWRTzzz_fs_1 XxxVelocityOfyyyWRTzzz_ms_1 YxxVelocityOfyyyWRTzzz_fs_1 YxxVelocityOfyyyWRTzzz_ms_1 ZxxVelocityOfyyyWRTzzz_fs_1 ZxxVelocityOfyyyWRTzzz_ms_1	General expression for velocities along the X, Y and Z axes of the xx coordinate system. yyy indicates the reference point on the vehicle and the Ofyyy may be omitted if it is the CG. zzz represents the frame that the vehicle is moving with repect to and the WRTzzz may be omitted if it is the inertial frame. So XFEVelocity_fs_1 is the inertial velocity of the vehicle CG along the X axis of the FE coordinate system and is the short version of XFEVelocityOfCGWRTInetial_fs_1.				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\underline{V}_B$	VELB	bodyVelocityWRTWind_fs_1[3] bodyVelocityWRTWind_ms_1[3] can also be expressed as: bodyVelocityOfCGWRTWind_fs_1[3]	Vector of body axis velocities of the cg with respect to the instantaneous wind comprised of the three components as defined below.				
u_B	*UB*	UBodyVelocityWRTWind_fs_1 UBodyVelocityWRTWind_ms_1	X-velocity Body axis system.	FWD			
v_B	*VB*	VBodyVelocityWRTWind_fs_1 VBodyVelocityWRTWind_ms_1	Y-velocity Body axis system	RT			
w_B	WB	WBodyVelocityWRTWind_fs_1 WBodyVelocityWRTWind_ms_1	Z-velocity Body axis system	DWN			
$\underline{V}_{B_I}$	VELB	bodyVelocity_fs_1[3] bodyVelocity_ms_1[3] can also be expressed as: bodyVelocityOfCGWRTInertial_fs_1[3]	Vector of body axis inertial translational velocities of the cg comprised of the three components as defined below.				
u_{B_I}	UBI	UBodyVelocity_fs_1 UBodyVelocity_ms_1	X-velocity Body axis system.	FWD			
v_{B_I}	VBI	VBodyVelocity_fs_1 VBodyVelocity_ms_1	Y-velocity Body axis system	RT			
w_{B_I}	WBI	WBodyVelocity_fs_1 WBodyVelocity_ms_1	Z-velocity Body axis system	DWN			
$\underline{V}_{FE}$	VELFE	FEVelocity_fs_1(3) FEVelocity_ms_1(3)	Vector of Flat Earth (FE) axis translational velocities comprised of the three components as defined below.				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
V_N	VNFE	`NfeVelocity_fs_1` `NfeVelocity_ms_1`	Northward Velocity Over Flat Earth (FE) axis system [flat, non-rotating earth]	NORTH			
V_E	VEFE	`EfeVelocity_fs_1` `EfeVelocity_ms_1`	Eastward Velocity Over Flat Earth (FE) axis system [flat, non-rotating earth]	EAST			
V_D	VDFE	`DfeVelocity_fs_1` `DfeVelocity_ms_1`	Downward Velocity Toward Earth Ctr,.(FE) axis system [flat, non-rotating earth]	DOWN			
$\underline{V}_{xx}$	VELxx	`xxVelocity_fs_1(3)` `xxVelocity_ms_1(3)`	Vector of vehicle cg inertial translational velocities in the specified xx axis system comprised of the three components as defined below.				
$V_{X_{xx}}$	VXxx	`XxxVelocity_fs_1` `XxxVelocity_ms_1`	X component of velocity with respect to the inertial reference frame in the specified (xx) axis system				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$V_{Y_{xx}}$	VYxx	YxxVelocity_fs_1 YxxVelocity_ms_1	Y component of velocity with respect to the inertial reference frame in the specified (xx) axis system				
$V_{Z_{xx}}$	VZxx	ZxxVelocity_fs_1 ZxxVelocity_ms_1	Z component of velocity with respect to the inertial reference frame in the specified (xx) axis system				
EXAMPLES							
		XGEVelocity_fs_1	X inertial velocity in the geocentric earth (GE) axis system in ft/sec				
		ZRunway22VelocityOfFwdLeftWheelWRTTD_fs_1	Z axis velocity of the “forward left wheel” (user defined) in the “runway22” (user defined) coordinate system in f/s	Down			
V_T	VTzzz	TotalVelocityWRTzzz_fs_1 TotalVelocityWRTzzz_ms_1	Total Velocity with respect to the reference frame zzz				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
V_G	VG	TotalVelocityWRTGround_fs_1 TotalVelocityWRTGround_ms_1 GroundSpeed_fs_1 GroundSpeed_ms_1	Vehicle velocity with respect to the ground under the vehicle cg				
M_N	XMACH	mach	Mach Number of the vehicle				
		xxVelocityWRTWind_fs_1[3] xxVelocityWRTWind_ms_1[3] or xxVelocityOfCGWRTWind_fs_1[3]	Vector of translational velocities of the CG wrt the instantaneous wind in the specified (xx) axis system comprised of the three components as defined below.				
	VXRWxx	XxxVelocityWRTWind_fs_1 XxxVelocityWRTWind_ms_1	X Relative Velocity of the CG with respect to the instantaneous wind in the xx axis system.	(CG velocity – wind velocity)			
	VYRWxx	YxxVelocityWRTWind_fs_1 YxxVelocityWRTWind_ms_1	Y Relative Velocity of the CG with respect to the instantaneous wind in the xx axis system.	(CG velocity – wind velocity)			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	VZRWxx	ZxxVelocityWRTWind_fs_1 ZxxVelocityWRTWind_ms_1	Z Relative Velocity of the CG with respect to the instantaneous wind in the xx axis system.	(CG velocity – wind velocity)			
$\dot{h}$	ALTD	AltitudeRate_fs_1 AltitudeRate_ms_1	Geometric altitude time rate of change.	DOWN			
	XLOND	longitudeRateWRTzzz_r longitudeRateWRTzzz_d	Rate of change of longitude of Vehicle CG with respect to the zzz reference frame.	WEST			
	XLATD	latitudeRateWRTzzz_r latitudeRateWRTzzz_d	Rate of change of latitude of Vehicle CG with respect to the zzz reference frame.	NORTH			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	XLONDIMU	longitudeRateOfIMUWRTzzz_r longitudeRateOfIMUWRTzzz_d	Rate of change of longitude of Vehicle IMU with respect to the zzz reference frame.	WEST			
	XLATDIMU	latitudeRateOfIMUWRTzzz_r latitudeRateOfIMUWRTzzz_d	Rate of change of latitude of Vehicle IMU with respect to the zzz reference frame.	NORTH			
EXAMPLE							
		longitudeRateOfIMUWRTWGS84_ds_1 latitudeRateOfIMUWRTWGS84_ds_1	Rate of change of longitude and latitude of the vehicle IMU in the World Grid System 1984 reference frame				
p_s	PS	rollSARate_rs_1 rollSARate_ds_1	Roll about the X axis in the SA (stability) axis system, also known as stability axis roll rate.	RWD			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
r_s	RS	`yawSARate _rs_1` `yawSARate_ds_1`	Yaw about the Z axis in the SA (stability) axis system, also known as the Stability Axis yaw rate	ANR			

Table A.3 — Vehicle linear and angular accelerations

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\underline{\omega}_B$	OMBD	`bodyAngularAccel_rs_2[3]` `bodyAngularAccel_ds_2[3]`	Vector of body axis angular accelerations comprised of the three components as defined below.				
p_B	PBD	`rollBodyAccel_rs_2` `rollBodyAccel_ds_2`	Vehicle Roll Acceleration, Body axis system	RWD			
q_B	QBD	`pitchBodyAccel_rs_2` `pitchBodyAccel_ds_2`	Vehicle Pitch Accel, Body axis system	ANU			
r_B	RBD	`yawBodyAccel_rs_2` `yawBodyAccel_ds_2`	Vehicle Yaw Acceleration, Body axis system	ANR			
		`bodyAccel_fs_2[3]` `bodyAccel_ms_2[3]`	Vector of accelerations of the cg of the vehicle wrt the interital frame in the body axis system. Therefore does not include the gravity vector. Comprised of the three components as defined below.				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\dot{u}_B$	UBD	UBodyAccel_fs_2 UBodyAccel_ms_2	Lonngitudinal acceleration (along the X-body axis)	FWD			
$\dot{v}_B$	VBD	VBodyAccel_fs_2 VBodyAccel_ms_2	Right Sideward Acceleration, (along the Y Body axis)	RT			
$\dot{w}_B$	WBD	WBodyAccel_fs_2 WBodyAccel_ms_2	Downward Acceleration, Body axis	DOWN			
$\dot{v}_T$	VTD	totalAccel_fs_2 totalAccel_ms_2	Rate of change of total velocity of the CG in the inertial frame				
$\underline{\dot{V}}_{xx}$		xxAccel_fs_2 xxAccel_ms_2	Vector of vehicle cg inertial translational accelerations in the specified (xx) axis system comprised of the three components as defined below.				
$\dot{V}_{X\,xx}$	VXD	XxxAccel_fs_2 XxxAccel_ms_2	Acceleration along the X axis				
$\dot{V}_{Y\,xx}$	VYD	YxxAccel_fs_2 YxxAccel_ms_2	Acceleration along the Y axis				
$\dot{V}_{Z\,xx}$	VZD	ZxxZAccel_fs_2 ZxxAccel_ms_2	Acceleration along the Z axis				
$\underline{\dot{V}}_{FE}$		FEAccel_fs_2 FEAccel_ms_2	Vector of vehicle cg translational accelerations in the FE (Flat Earth) axis system comprised of the three components as defined below.				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\dot{V}_{X_{FE}}$	VND	NFEAccel_fs_2 NFEAccel_ms_2	North Acceleration Over flat earth	NORTH			
$\dot{V}_{Y_{FE}}$	VED	EFEAccel_fs_2 EFEAccel_ms_2	East Acceleration Over flat earth	EAST			
$\dot{V}_{Z_{FE}}$	VDD	DFEZAccel_fs_2 DFEAccel_ms_2	Down Acceleration Toward flat earth surface	DOWN			
		bodyAccelSensedOfCG_fs_2[3] bodyAccelSensedOfCG_ms_2[3]	Vector of accelerations sensed at the cg (including the effects of the gravity vector) in the body axis system. Comprised of the three components as defined below.				
	AX	XBodyAccelSensedOfCG_fs_2 XBodyAccelSensedOfCG _ms_2	X Acceleration Of A/c C.g. (body axis) Includes the gravity vector.	FWD			
	AY	YBodyAccelSensedOfCG _fs_2 YBodyAccelSensedOfCG _ms_2	Y Acceleration Of A/c C.g. (body axis) Includes the gravity vector.	RT			
	AZ	ZBodyAccelSensedOfCG _fs_2 ZBodyAccelSensedOfCG _ms_2	Z Acceleration Of A/c C.g. (body axis) Includes the gravity vector.	DOWN			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		bodyAccelOfPilot_fs_2[3] bodyAccelOfPilot_ms_2[3]	Vector of accelerations at the pilot reference point, in the body axis system, comprised of the three components as defined below.				
	AXP	XBodyAccelOfPilot_fs_2 XBodyAccelOfPilot _ms_2	X Acceleration Of Pilot reference point (body axis)	FWD			
	AYP	YBodyAccelOfPilot _fs_2 YBodyAccelOfPilot _ms_2	Y Acceleration Of Pilot reference point(body axis)	RT			
	AZP	ZBodyAccelOfPilot _fs_2 ZBodyAccelOfPilot _ms_2	Z Acceleration Of Pilot reference point(body axis)	DOWN			
	G	localGravity_fs_2 localGravity_ms_2	Acceleration Due To Gravity (at the vehicle altitude)	DOWN			

Table A.4 — Vehicle air data

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
α	ALFA	angleOfAttack_d angleOfAttack_r	Angle Of Attack, Body axis	ANU		$-\pi$,-180	$+\pi$,+180
β	BETA	angleOfSideslip_d angleOfSideslip_r	Sideslip Angle, Body axis	ANL		$-\pi$,-180	$+\pi$,+180

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\dot{\alpha}$	ALFD	angleOfAttackRate_rs_1	Angle Of Attack Rate, Body axis	ANU			
$\dot{\beta}$	BETD	angleOfSideslipRate_rs_1	Sideslip Angle Rate	ANL			
$\sin\alpha$	SALPH	sineAngleOfAttack	Sine Of Angle Of Attack	ANU		-1.0	1.0
$\cos\alpha$	CALPH	cosineAngleOfAttack	Cosine Of Angle Of Attack	ANU		-1.0	1.0
$\sin\beta$	SBETA	sineAngleOfSideslip	Sine Of Sideslip Angle	ANL		-1.0	1.0
$\cos\beta$	CBETA	cosineAngleOfSideslip	Cosine Of Sideslip Angle	ANL		-1.0	1.0
V_{CAL}	VCAL	calibratedAirspeed_nmih_1	Calibrated Air Speed, knots	FWD			
V_{EQ}	VEQ	equivalentAirspeed_nmih_1	Equivalent Air Speed	FWD			
V_{IND}	VCAL	indicatedAirspeed_nmih_1	Calibrated Air Speed,	FWD			
V_{RW}	VRW	trueAirspeed_fs_1 trueAirspeed_ms_1 trueAirspeed_ nmih_1	Vehicle Velocity relative to the local wind (true airspeed)	FWD			
$\bar{q}$	QBAR	dynamicPressure_lbff_2 dynamicPressure_Nm_2	Dynamic Pressure	NSC			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\overline{q}_c$	QBARC	impactPressure_lbff_2 impactPressure_Nm_2	Impact Pressure	NSC			
ρ	RHO	airDensity_lbmf_3 airDensity_kgpm_3	Air Density, At Altitude of the vehicle	NSC			
	DENALT	densityAltitude_f densityAltitude_f	Density altitude				
a	SOUND	speedOfSound_fs_2 speedOfSound_ms_2	Velocity Of Sound At Altitude of the vehicle	NSC			
T_{TOT_R}	TR	totalTempRatio_C totalTempRatio_K	Total Temperature Ratio	NSC			
P_{TOT_R}	PR	totalPressureRatio_C totalPressureRatio_K	Total Pressure Ratio	NSC			
T_{AMB}	TAMB	ambientTemperature_C ambientTemperature_K	Ambient Temperature at altitude	NSC			
P_{AMB}	PAMB	ambientPressure_lbff_2 ambientPressure_Nm_2	Ambient Pressure at altitude	NSC			
P_{AMB_R}	PAMBR	ambientPressureRatio	Ratio Of ambient pressure at altitude to sea level ambient pressure	NSC			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
T_{AMB_R}	TAMBR	`ambientTemperatureRatio`	Ratio Of ambient temperature at altitude to sea level ambient temp.	NSC			
T_{TOT}	TTOT	`totalTemp_C` `totalTemp_K`	Total Temperature at altitude	NSC			
P_{TOT}	PTOT	`totalPressure_lbff_2` `totalPressure_Nm_2`	Total Pressure at altitude	NSC			

	TAMB_R	`ambientTemperatureAtAlt_K`
		`ambientTemperatureAtAlt_R`
		`ambientTemperatureAtAlt_C`
			Ambient temperature, at the altitude of the CG				

	TTOT_R	`totalTemperatureAtAlt_K`
		`totalTemperatureAtAlt_R`
		`totalTemperatureAtAlt_C`
			Total temperature at the altitude of the CG				

	ALT_SET	`InstrumentAltimeterSetting_inchMercury`	Cockpit Altimeter setting (Kohlsman window)	29.92 is standard day			
	P_ALT	`PressureAltitude_f` `PressureAltitude_m`	Pressure altitude at the CG				
	RHO_SL	`seaLevelAirDensity_lbfpf3`	Air density at sea level				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	TAMB_SL	seaLevelAmbientTemp_K seaLevelAmbientTemp_R seaLevelAmbientTemp_C	Ambient temperature at mean sea level				
	PAMB_SL	seaLevelAmbientPressure_lbff2 seaLevelAmbientPressure_Nm2	Ambient pressure at sea level				

Table A.5 — Atmospheric disturbances and turbulence

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	WIND_SPEED	steadyStateWindVelocity_fs_1 steadyStateWindVelocity_ms_1	Total velocity of steady wind				
	WIND_DIRECTION	steadyStateWindDirection_d	Steady wind heading (blowing FROM true North)	Wind blowing from			
$\underline{V}_{B_{Turb}}$	VELBT	bodyTurbulenceVelocity_fs_1[3] bodyTurbulenceVelocity_ms_1[3]	Vector of body axis translational turbulence velocities comprised of the three components as defined below.				
$u_{B_{Turb}}$	UBTURB	UbodyTurbulenceVelocity_fs_1 UbodyTurbulenceVelocity_ms_1	X-velocity Turb. Component, Body axis	FWD			
$v_{B_{Turb}}$	VBTURB	VbodyTurbulenceVelocity_fs_1 VbodyTurbulenceVelocity_ms_1	Y-velocity Turb. Component, Bodyaxis	RT			
$w_{B_{Turb}}$	WBTURB	WbodyTurbulenceVelocity_fs_1 WbodyTurbulenceVelocity_ms_1	Z-velocity Turb. Component, Body axis	DWN			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
$\underline{V}_{W_{xx}}$	VWxx	xxWindVelocity_fs_1S[3} xxWindVelocity_ms_1[3]	Vector of wind velocities velocities in the specified (xx) axis system comprised of the three components as defined below. Only applies to earth fixed axis systems.				
W_N	VNWxx	NxxWindVelocity_fs_1 NxxWindVelocity_ms_1	North component of wind velocity in xx axis system	To the North			
W_E	VEWxx	ExxWindVelocity_fs_1 YxxWindVelocity_ms_1	East component Of wind velocity in xx axis system.	To the East			
W_D	VDWxx	ExxWindVelocity_fs_1 ExxWindVelocity_ms_1	Down Component Of Wind Velocity in xx axis system.	To Downward			
$W_{T_{xx}}$	VTWxx	xxTotalwindVelocity_fs_1 xxTotalwindVelocity_ms_1	Total Wind Velocity, in xx axis system.	NSC			
		netWindVel_fs_1[3] netWindVel_ms_1[3]	Vector of the net wind velocities impinging on the vehicle. Comprised of the three components as defined below.				
	VTWN	netWindVelFromNorth_fs_1 netWindVelFromNorth_ms_1	Net wind velocity from North. Net wind is the steady state winds plus any turbulences and shears.	From the North			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	VTWE	netWindVelFromEast_fs_1 netWindVelFromEast_ms_1	Net wind velocity from East. Net wind is the steady state winds plus any turbulences and shears.	From the East			
	VTWD	netWindVelFromBelow_fs_1 netWindVelFromBelow_ms_1	Net wind velocity from below. Net wind is the steady state winds plus any turbulences and shears.	From below			
		turbulence_fs_1[3] turbulence_ms_1[3]	Vector of the wind turbulence velocities impinging on the vehicle. Comprised of the three components as defined below.				
	VNTURB	turbulenceFromNorth_fs_1 turbulenceFromNorth_ms_1	North component of turbulence	From the North			
	VETURB	turbulenceFromEast_fs_1 turbulenceFromEast_ms_1	East component of turbulence	From the East			
	VDTURB	turbulenceFromBelow_fs_1 turbulenceFromBelow_ms_1	Vertical component of turbulence	From below			
		bodyAngularTurbulence_ds_1[3] bodyAngularTurbulence_rs_1[3]	Vector of angular turbulence velocities comprised of the three components as defined below. Body axis system.				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	PTURB	rollBodyTurbulenceRate_ds_1 rollBodyTurbulenceRate_rs_1	Body axis roll turbulence	The turbulence would move the aircraft right wing down			
	QTURB	pitchBodyTurbulenceRate_ds_1 pitchBodyTurbulenceRate_rs_1	Body axis pitch turbulence	The turbulence would move the aircraft nose up			
	RTURB	yawBodyTurbulenceRate_ds_1 yawBodyTurbulenceRate_rs_1	Body axis yaw turbulence	The turbulence would move the aircraft nose right			

Table A.6 — Vehicle physical characteristics

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
I		bodyMomentOfInertia_slugf2[3,3] bodyMomentOfInertia_kgm2[3,3]	Matrix of the total moments of inertia of the vehicle. This is wrt the CG and includes everything in or attached to the vehicle (stores, passengers, crew, fuel, etc.). It is comprised of the components below. I_{XX} $-I_{XY}$ $-I_{ZX}$ $-I_{XY}$ I_{YY} $-I_{YZ}$ $-I_{ZX}$ $-I_{YZ}$ I_{ZZ}				
I_{xx}	XIXX	bodyXXMomentOfInertia_slugf2 bodyXXMomentOfInertia_kgm2	Vehicle Roll Moment Of Inertia about Cg, body axis system	NSC			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
I_{xx}	XIYY	bodyYYMomentOfInertia_slugf2 bodyYYMomentOfInertia_kgm2	Vehicle Pitch Moment Of Inertia about Cg, body axis system	NSC			
I_{zz}	XIZZ	bodyZZMomentOfInertia_slugf2 bodyZZMomentOfInertia_kgm2	Vehicle Yaw Moment Of Inertia about Cg, body axis system	NSC			
I_{xz}	XIZX	bodyZXProductOfInertia_slugf2 bodyZXProductOfInertia_kgm2	Vehicle ZX Cross Product Of Inertia about Cg, body axis system	NSC			
I_{xy}	XIXY	bodyXYProductOfInertia_slugf2 bodyXYProductOfInertia_kgm2	Vehicle XYy Cross Product Of Inertia about Cg, body axis system	NSC			
I_{yz}	XIYZ	bodyYZProductOfInertia_slugf2 bodyZProductOfInertia_kgm2	Vehicle YZ Cross Product Of Inertia about Cg, body axis system	NSC			
		bodyPositionOfCG_f[3] bodyPositionOfCG_f _m[3]	Vector of the CG position of the vehicle in the body axis system. Comprised of the three components as defined below.				
	XCGREF	XBodyPositionOfCG_f XBodyPositionOfCG_m	C.g. Position w/r/t L.e. Of the mean aerodynamic chord	CG AFT of LEMAC			
	YCGREF	YBodyPositionOfCG_f YBodyPositionOfCG_m	C.g. Position w/r/t the centerline of the vehicle	CG Right of the a/c centerline			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	ZCGREF	ZBodyPositionOfCG_f ZBodyPositionOfCG_m	C.g. Position w/r/t the waterline reference of the vehicle (usually WL 0, see ZBodyWaterline_)	CG below the a/c waterline reference			
		bodyPositionOfCGWRTMRC_f[3] bodyPositionOfCGWRTMRC_m[3]	Vector of the distance from the Moment Reference center to the CG position of the vehicle in the body axis system. Comprised of the three components as defined below.				
ΔX_{cg}	DXCG	XPositionOfCGWRTMRC_f XPositionOfCGWRTMRC _m	Cg Displacement From the aerodynamic force and moment reference center, + is CG fwd of the Moment Reference Center (MRC). The MRC is the reference point that the aero model forces and moments act upon the vehicle.	FWD			
ΔY_{cg}	DYCG	YPositionOfCGWRTMRC _f YPositionOfCGWRTMRC _m	Cg Displacement From the aerodynamic force and moment reference center, + is CG to the right of the ARC	RT			
ΔZ_{cg}	DZCG	ZPositionOfCGWRTMRC_f ZPositionOfCGWRTMRC_m	Cg Displacement From the aerodynamic force and moment reference center, + is CG below the the ARC	DWN			
		bodyPositionOfMRC_f[3] bodyPositionOfMRC_m[3]	Vector of the location of the moment reference center (MRC) of the vehicle in the body axis system. Comprised of the three components as defined below.				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	XMRC	XBodyPositionOfMRC_f XBodyPositionOfMRC_m	X MRC Position w.r.t L.e. Of the mean aerodynamic chord	MRC AFT of LEMAC			
	YMRC	YBodyPositionOfMRC_f YBodyPositionOfMRC_m	Y MRC Position w.r.t. the centerline of the vehicle	MRC Right of the a/c centerline			
	ZMRC	ZBodyPositionOfMRC_f ZBodyPositionOfMRC_m	Z MRC Position w.r.t. the waterline reference of the vehicle (usually WL 0, see ZBodyWaterlinePositio n_)	MRC below the a/c waterline reference			
	ZWL	ZBodyWaterlinePosition_f ZBodyWaterlinePosition_m	The waterline (vertical) reference position on the a/c body. This is a constant used to locate the vertical cg and MRC postion to the vehicle. Waterline reference position is normally 0 but does not have to be.	NSC			
M	XMASS	totalMass_slug totalMass_kg	Total mass of vehicle (including fuel, crew, cargo, stores, passengers, etc.)	NSC			
W	WEIGHT	grossWeight_lbf grossWeight_N	Vehicle gross weight (mass*gravity), including all fuel, occupants, stores, etc.	NSC			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
A	AREA	`referenceWingArea_f2` `referenceWingArea_m2`	Reference wing area	NSC			
b	SPAN	`referenceWingSpan_f` `referenceWingSpan_m`	Reference wing span	NSC			
c	CHORD	`referenceWingChord_f` `referenceWingChord_m`	Mean aerodynamic chord (reference wing chord)	NSC			
		`engineMomentOfInertia_slugf2` `engineMomentOfInertia_kgm2`	Matrix of the moments of inertia of the rotating engine, for an engine with a propeller, includes the propeller and drive train. This convention is for rotation of the engine about the X axis. For a propeller driven aircraft it is for rotaton of the propeller about the X axis. For multi-engine vehicles is for one engine. It is comprised of the components below. I_{EXX} $-I_{EXY}$ $-I_{EZX}$ $-I_{EXY}$ I_{EYY} $-I_{EYZ}$ $-I_{EZX}$ $-I_{EYZ}$ I_{EZZ}				
I_{Exx}	IEXX	`engineXXMomentOfInertia_slugf2` `engineXXMomentOfInertia_kgm2`	Moment of inertia about the X-axis Of Rotating Eng, for an engine with a propeller, includes the propeller. This is w.r.t. the rotational axis of the engine				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
I_{EYY}	IEYY	engineYYMomentOfInertia_slugf2 engineYYMomentOfInertia_kgm2	Moment of inertia about the Y-axis Of Rotating Eng, for an engine with a propeller, includes the propeller. This is w.r.t. the rotational axis of the engine				
I_{EZZ}	IEZZ	engineZZMomentOfInertia_slugf2 engineZZMomentOfInertia_kgm2	Moment of inertia about the Z-axis Of Rotating Eng, for an engine with a propeller, includes the propeller. This is w.r.t. the rotational axis of the engine				
I_{EXZ}	IEXZ	engineXZProductOfInertia_slugf2 engineXZProductOfInertia_kgm2	Product of inertia about the XZ-axis Of Rotating Eng, for an engine with the propeller, includes the propeller This is w.r.t. the rotational axis of the engine				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
I_{EXY}	IEXY	engineXYProductOfInertia_slugf2 engineXYProductOfInertia_kgm2 (engine_xy_product_of_inertia_slugf2)	Product of inertia about the XY-axis Of Rotating Eng, for an engine with the propeller, includes the propeller This is w.r.t. the rotational axis of the engine				
I_{EYZ}	IEYZ	engineYZProductOfInertia_slugf2 engineYZProductOfInertia_kgm2 (engine_yz_product_of_inertia_slugf2)	Product of inertia about the YZ-axis Of Rotating Eng, for an engine with the propeller, includes the propeller This is w.r.t. the rotational axis of the engine				
		fuelInTank_lbm[number of fuel tanks] fuelInTank_kg[number of fuel tanks]	Vector of fuel weight by tank. Each vehicle tank is normally numbered and the vector should be ordered according to fuel tank number. In the absence of tank numbering the convention of port to starboard, upper to lower, then front to rear should be used.				

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		fuelTankCentroid_f[number of fuel tanks,3] fuelTankCentroid_m[number of fuel tanks,3]	Matrix used to locate the centoids of the fuel tanks. Each vehicle tank is normally numbered and the matrix should be ordered according to fuel tank number. The second component is the x, y and z moment arms from the moment reference center to the tank centroid in the body axis. In the absence of tank numbering the convention of port to starboard, upper to lower, then front to rear should be used.	Tank centroid behind, right, and below the moment reference center.			

Table A.7 — Vehicle control position

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		pilotLongControlPos_d pilotLongControlPos_r	Longitudinal control position of the pilot.	AFT			
		pilotLatControlPos_d pilotLongControlPos_r	Lateral control position of the pilot.	RT			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		pilotPedalControlPos_d pilotPedalControlPos_r	Net Directional control position of the pilot. Normally, Right pedal – left pedal.	Pedal in or clockwise twist of a sidestick			
		pilotRightPedalControlPos_d pilotRightPedalControlPos_r	Right Directional control position of the pilot.	Pedal in.			
		pilotLeftPedalControlPos_d pilotLeftPedalControlPos_r	Left Directional control position of the pilot.	Pedal in.			
		pilotCollectiveControlPos_d pilotCollectiveControlPos_r	Pilot collective control position.	UP			
		pilotAvgThrottleControlPos_d pilotAvgThrottleControlPos_r	Average pilot throttle control position.	FWD			
		pilotThrottleControlPos_d [number of engines] pilotThrottleControlPos_r [number of engines]	Individual pilot throttle control positions. Order is outboard port (left) to outboard starboard.	FWD			
		copilotLongControlPos_d copilotLongControlPos_r	Longitundal control position of the copilot.	AFT			
		copilotLatControlPos_d copilotLongControlPos_r	Lateral control position of the copilot.	RT			
		copilotPedalControlPos_d copilotPedalControlPos_r	Net Directional control position of the copilot. Nornally, Right pedal – left pedal.	Pedal in or clockwise twist of a sidestick.			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		copilotRightPedalControlPos_d copilotRightPedalControlPos_r	Right Directional control position of the copilot.	Pedal in.			
		copilotLeftPedalControlPos_d copilotLeftPedalControlPos_r	Left Directional control position of the copilot.	Pedal in.			
		copilotCollectiveControlPos_d copilotCollectiveControlPos_r	Copilot collective control position.	UP			
		copilotAvgThrottleControlPos_d copilotAvgThrottleControlPos_r	Average copilot throttle control position.	FWD			
		copilotThrottleControlPos_d [number of engines] copilotThrottleControlPos_r [number of engines]	Individual copilot throttle control positions. Order is outboard port (left) to outboard starboard.	FWD			
		avgThrottleControlPos_d avgThrottleControlPos_r	Average pilot and copilot throttle control position.	FWD			
		throttleControlPos_d[number of engines] throttleControlPos_r[number of engines]	Individual throttle control position (pilot and copilot average). Order is outboard port (left) to outboard starboard.	FWD			
		avgPropControlPos_d avgPropControlPos_r	Average pilot and copilot propeller blade pitch control position.	FWD			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		propControlPos_d[number of engines] propControlPos_r[number of engines]	Individual propeller blade pitch control position. Order is outboard port (left) to outboard starboard.	FWD			
		trailingEdgeFlapDeflection[number of leading edge flap control surfaces]	Vector of trailing edge flap positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	LED			
		avgTrailingEdgeFlapDeflection_d	Trailing edge flap deflection. Average for all trailing edge flap surfaces.	TED			
		differentialTrailingEdgeFlapDeflection_d	Measure of roll control due to trailing edge flap deflection differences in vehicles with multiple control surfaces, usually (left deflections-right deflections)	RWD control			
		leadingEdgeFlapDeflection[number of leading edge flap control surfaces]	Vector of leadng edge flap positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	LED			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		avgLeadingEdgeFlapDeflection_d	Leading edge flap/slat deflection. Average for all deflected leading edge flap/slat surfaces.	LED			
		differentialLeadingEdgeFlapDeflection_d	Measure of roll control due to leading edge flap deflection differences in vehicles with multiple control surfaces, usually (left deflections-right deflections)	RWD control			
		spoilerDeflection[number of spoiler control surfaces]	Vector of spoiler control positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	TEU			
		avgSpoilerDeflection_d	Spoiler deflection. Average for all deflected spoilers	TEU			
		differentialSpoilerDeflection_d	Measure of roll control due to spoiler deflection differences in vehicles with multiple control surfaces, usually (right deflections-left deflections)	RWD control			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		aileronDeflection[number of aileron control surfaces]	Vector of aileron control positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	TEU			
		avgAileronDeflection	Differential aileron deflection, right-left	Right aileron TEU			
		rudderDeflection_d[number of rudder control surfaces]	Vector of rudder control positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	TEL			
		avgRudderDeflection_d	Average rudder deflection	TEL			
		differentialRudderDeflection_d	Measure of yaw control due to rudder deflection differences in vehicles with multiple control surfaces, usually (right deflections-left deflections)				
		rudderTabDeflection_d[number of rudder tab control surfaces]	Vector of rudder tab control positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	TEL			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		avgRudderTabDeflection_d	Average rudder tab deflection	TEL			
		differentialRudderTabDeflection_d	Measure of yaw control due to rudder tab deflection differences in vehicles with multiple control surfaces, usually (right deflections-left deflections)				
		elevatorDeflection_d[number of elevator control surfaces]	Vector of elevator (or stabilizer/stabilator) control positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	TEU			
		avgElevatorDeflection_d	Average elevator (or stabilizer/stabilator) deflection	TEU			
		differentialElevatorDeflection_d	Measure of roll control due to elevator (or stabilizer/stabilator) deflection differences in vehicles with multiple control surfaces, usually (right deflections-left deflections)	Right control TEU			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		elevatorTabDeflection_d[number of elevator tab control surfaces]	Vector of elevator (or stabilizer/stabilator) tab control positions, one for each surface deflected. Order is outboard port (left) to outboard starboard.	TEU			
		avgElevatorTabDeflection_d	Average elevator (or stabilizer/stabilator) tab deflection	TEU			
		differentialElevatorTabDeflection_d	Measure of roll control due to elevator (or stabilizer/stabilator) tab deflection differences in vehicles with multiple control surfaces, usually (right deflections-left deflections)	Right control TEU			
		canardDeflection_d[number of canard control surfaces]	Vector of canard control positions, one for each surface. Order is outboard port (left) to outboard starboard.	TED			
		avgCanardDeflection_d	Average canard deflection	TED			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		differentialCanardDeflection_d	Measure of roll control due to canard deflection differences in vehicles with multiple control surfaces, usually (right deflections-left deflections)	Right control TED			
		canardTabDeflection_d[number of canard tab control surfaces]	Vector of canard tab control positions, one for each surface. Order is outboard port (left) to outboard starboard.	TED			
		avgCanardTabDeflection_d	Average canard tab deflection	TED			
		differentialCanardTabDeflection_d	Measure of roll control due to canard tab deflection differences in vehicles with multiple control surfaces, usually (right deflections-left deflections)	Right control TED			
		speedbrakeDeflection_d	Speedbrake deflection	Extended			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
		`landingGearPosition[number of landing gear struts]`	Vector of landing gear positions, one for each strut. Order is outboard port (left) to outboard starboard.	0= up and locked 1= full extension with no weight on wheels			
		`landingGearWeightOnWheels_lbf [number of landing gear struts]` `landingGearWeightOnWheels_kg {number of landing gear struts]`	Vector of landing gear weight on wheels, one for each strut. Order is outboard port (left) to outboard starboard.				
		`landingGearWheelSpeed_rs_1[number of landing gear struts, number of trucks, number of wheels per truck]`	Array of landing gear wheel speeds by strut, one for each strut. Order of struts is outboard port (left) strut, to outboard starboard. Order of trucks is front to rear. Order of wheels on each truck is port to starboard.				

Table A.8 — Vehicle aerodynamic characteristics

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
C_L	CL	`totalCoefficientOfLift`	Coefficient Of Lift, Total, includes effects of stores	UP			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
C_D	CD	`totalCoefficientOfDrag`	Coefficient Of Drag, Total, includes effects of stores	AFT			
		`aeroBodyForceCoefficient[3]`	Vector of total aerodynamic force coefficients in the body axis system, comprised of the three components as defined below.				
C_X	CX	`aeroXBodyForceCoefficient`	X-body Force Coefficient due to aerodynamic loads, includes stores (Body axis)	FWD			
C_Y	CY	`aeroYBodyForceCoefficient`	Y-body Force Coefficient due to aerodynamic loads, includes stores (Body axis)	RT			
C_Z	CZ	`aeroZBodyForceCoefficient`	Z-body Force Coefficient due to aerodynamic loads, includes stores (Body axis)	DOWN			
		`aeroBodyForce_lbf[3]` `aeroBodyForce_N[3]`	Vector of total aerodynamic forces in the body axis system, including stores. Comprised of the three components as defined below.				
F_{AX}	FAX	`aeroXBodyForce_lbf` `aeroXBodyForce_N`	Total X-body Force due to aerodynamic loads, includes stores (Body axis)	FWD			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
F_{AY}	FAY	aeroYBodyForce_lbf aeroYBodyForce_N	Total Y-body Force due to aerodynamic loads, includes stores (Body axis)	RT			
F_{AZ}	FAZ	aeroZBodyForce_lbf aeroZBodyForce_N	Total Z-body Force due to aerodynamic loads, includes stores (Body axis)	DOWN			
		thrustBodyForce_lbf[3] thrustBodyForce_N[3]	Vector of total net propulsion system forces in the body axis system (includes installion losses, inlet efficieny and propeller efficiency). Comprised of the three components as defined below.				
F_{EX}	FEX	thrustXBodyForce_lbf thrustXBodyForce_N	Total net engine thrust Force, X-body axis	FWD			
F_{EY}	FEY	thrustYBodyForce_lbf thrustYBodyForce_N	Total net engine thrust Force , Y-body axis	RT			
F_{EZ}	FEZ	thrustZBodyForce_lbf thrustZBodyForce_N	Total net engine thrust Force, Z-body axis	DOWN			
		gearBodyForce_lbf[3] gearBodyForce_N[3]	Vector of total landing gear ground reaction forces in the body axis system. Does NOT include aerodynamic forces on the landing gear which are included in `aeroBodyForce` defined above. Comprised of the three components as defined below.				
F_{GX}	FGX	gearXBodyForce_lbf gearXBodyForce_N	Total landing gear ground reaction force, X-body axis	FWD			
F_{GY}	FGY	gearYBodyForce_lbf gearYBodyForce_N	Total landing gear ground reaction force, Y-body axis	RT			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
F_{GZ}	FGZ	gearZBodyForce_lbf gearZBodyForce_N	Total landing gear ground reaction force, Z-body axis	DOWN			
		totalBodyForce_lbf[3] totalBodyForce_N[3]	Vector of total forces in the body axis system. Includes all forces exerted upon the aircraft. Comprised of the three components as defined below.				
F_{xTOT}	FX	totalXBodyForce_lbf totalXBodyForce_N	Total Forces On a/c, X-body axis	FWD			
F_{yTOT}	FY	totalYBodyForce_lbf totalYBodyForce_N	Total Forces On a/c, Y-body axis	RT			
F_{zTOT}	FZ	totalZBodyForce_lbf totalZBodyForce_N	Total Forces On a/c, Z-body axis	DOWN			
		aeroBodyMomentCoefficient[3]	Vector of total aerodynamic moment coefficients in the body axis system, including stores. Comprised of the three components as defined below.				
C_l	CLL	aeroRollBodyMomentCoefficient	Total Aerodynamic Rolling Moment Coefficient including stores. Moment about the X-body axis	RWD			
C_m	CLM	aeroPitchBodyMomentCoefficient	Total Aerodynamic Pitching Moment Coefficient, including stores. Moment about the Y-body axis	ANU			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
C_n	CLN	`aeroYawBodyMomentCoefficient`	Total Aerodynamic yawing Moment Coefficient, including stores. Moment about the Z-body axis	ANR			
		`aeroBodyMoment_flbf[3]` `aeroBodyMoment_Nm[3]`	Vector of total aerodynamic moments in the body axis system, including stores. Referenced to the moment reference center. Comprised of the three components as defined below.				
L_A	TAL	`aeroRollBodyMoment_flbf` `aeroRollBodyMoment_Nm`	Total Aerodynamic Rolling moment (including attached stores), about the X-body axis	RWD			
M_A	TAM	`aeroPitchBodyMoment_flbf` `aeroPitchBodyMoment_Nm`	Total Aerodynamic pitching moment (including attached stores), about the Y-body axis	ANU			
N_A	TAN	`aeroYawBodyMoment_flbf` `aeroYawBodyMoment_Nm`	Total Aerodynamic yawing moment (including attached stores), about the Z-body axis	ANR			
		`thrustBodyMoment_flbf[3]` `thrustBodyMoment_Nm[3]`	Vector of total net propulsion system moments in the body axis system (includes installation losses, inlet efficiency and propeller efficiency). Referenced to the moment reference center. Comprised of the three components as defined below.				
L_E	TEL	`thrustRollBodyMoment_flbf` `thrustRollBodyMoment_Nm`	Total Engine Rolling Moment, about the X-body axis	RWD			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
M_E	TEM	thrustPitchBodyMoment_flbf thrustPitchBodyMoment_Nm [thrust_body_pitch_moment_flbf]	Total Engine pitching Moment, about the Y-body axis	ANU			
N_E	TEN	thrustYawBodyMoment_flbf thrustYawBodyMoment_Nm	Total Engine yawing Moment, about the X-body axis	ANR			
		landingGearBodyMoment_flbf[3] landingGearBodyMoment_Nm[3]	Vector of total landing gear ground reaction moments in the body axis system. Referenced to the moment reference center. Does NOT include aerodynamic moments on the landing gear which are included in `aeroBodyMoment` defined above. Comprised of the three components as defined below.				
L_G	TGL	landingGearRollBodyMoment_flbf landingGearRollBodyMoment_Nm	Total Landing Gear Rolling Moment, about the X-body axis	RWD			
M_G	TGM	landingGearPitchBodyMoment_flbf landingGearPitchBodyMoment_Nm	Total Landing gear Pitch Moment, about the Y-body axis	ANU			
N_G	TGN	landingGearYawBodyMoment_flbf landingGearYawBodyMoment_Nm	Total Landing Gear Yawing Moment, about the Z-body axis	ANR			
		totalBodyMoment_flbf[3] totalBodyMoment_Nm[3]	Vector of total moments in the body axis system. Referenced to the moment reference center. Includes all moments exerted upon the aircraft. Comprised of the three components as defined below.				
L_{TOT}	TTL	totalRollBodyMoment_flbf totalRollBodyMoment_Nm	Total Rolling Moment, about the X-body axis	RWD			
M_{TOT}	TTM	totalPitchBodyMoment_flbf totalPitchBodyMoment_Nm	Total Pitching Moment, about the Y-body axis	ANU			

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
N_{TOT}	TTN	totalYawBodyMoment_flbf totalYawBodyMoment_Nms	Total Yawing Moment, about the Z-body axis	ANR			

Table A.9 — Simulation control parameters

Symbol	Short Name	Full Variable Name	Description	Sign Convention	Initial Value	Min Value	Max Value
	TIME	simTime_s simTime_s [sim_time_s]	Time Since Start Of Operate Mode	NSC			
		deltaTime_s [number of different integration step sizes]	Vector of Integration step sizes				

Annex B Dynamics Aerospace Vehicle Exchange Markup Language (DAVE-ML) Reference (Normative)

For the latest update to the DAVE-ML Reference document, please see http://daveml.nasa.gov.

Annex C DAVE-ML Website (Informative)

The “official” DAVE-ML site is http://daveml.nasa.gov/. This link contains all DAV-ML documentation and links and information on DAVE-ML tools and applications. Additional information is available at http://www.aiaa.org/