PSU-AESS
Notes on coordinates, units, and notation

T.Brandon

Created Oct. 27 1999

Coordinate systems

There is as yet no great solution to the problem of consistent, efficient, robust, and unambiguous notation, The philosophy expressed here is meant to be pragmatic for our circumstances. In the field of aerodynamics there are many notational conventions most of which adhere to a common subset of symbols. I feel it's in our interest to maintain at least superficial compatibility with the core convention. Which really means that we should avoid using a letter for something when in the literature it's consistently used for something else, especially when the subject makes it likely that the symbols would occur in the same expressions.

To this end i present a list of common symbols with their meanings and units. This is just a suggestion list, although a possibly nice thing to do would be to set up a common list for people working on software and modeling to refer to and update with their own symbols. The idea being to keep everyone as consistent as possible, and to facilitate a group understanding of everyone's individual contributions, and also to avoid mistakes that arise from basic misunderstandings.

Further, i propose a system of coordinates and a notation for the same to be used for navigation and control of our rocket, and possibly for its mechanical design and construction. This is also a preliminary proposal whose purpose is to immediately get those people who are working on theoretical and design issues to adopt a common notation, hopefully to gain benefits similar to those mentioned above. So here it is.

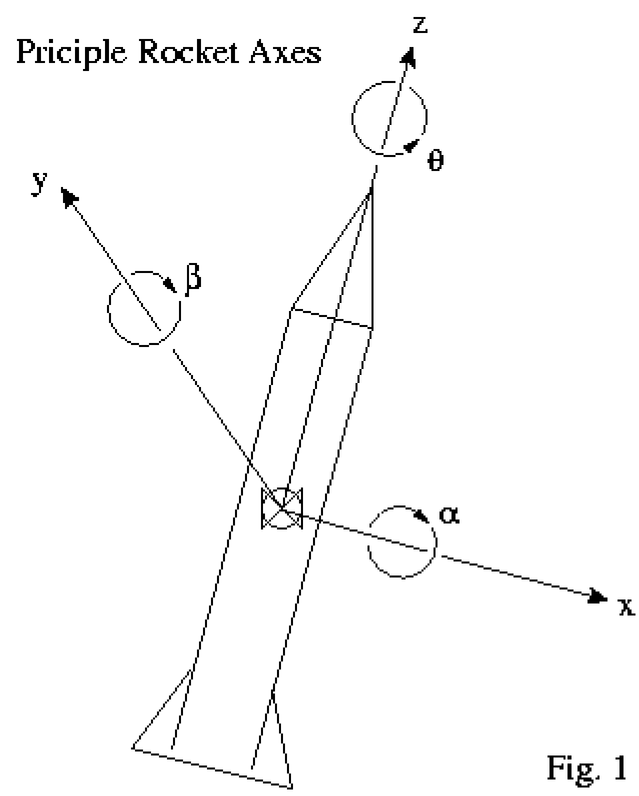

Fig. 1

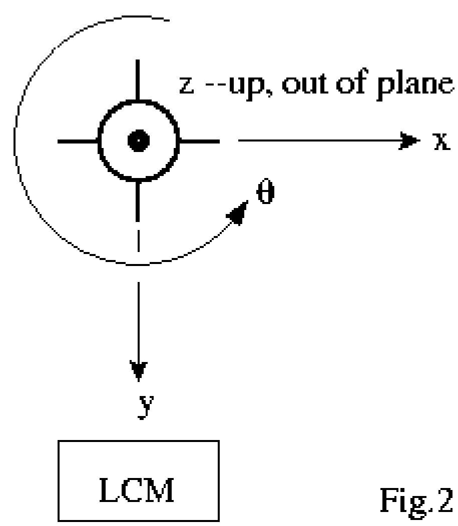

Fig.2

In Figure 1 we see the six principle coordinates. The major achievement of this arrangement is that the angular coordinates follow the right-hand rule about the axes. Also a positive rotation about the x -axis causes a positive projection of the z-axis onto the y-axis, and likewise for y. Figure 2 shows an overhead view of the rocket in launch configuration. This is relevant for ground tracking purposes. Note that the alpha coordinate represents the angle of attack, as is conventional, by extension beta would be the yaw, or if you prefer, the other angle of attack.

Example symbols

Here are some recommended symbols and MKS units

Symbol	Meaning	Unit

Position

x	Cross-range position	m
y	Down-range position	m
z	Altitude	m
alpha	Down-range attitude	radians
beta	Cross-range attitude	radians
theta	Roll angle	radians

Velocity

U	Free stream velocity (z-axis)	m/s
u	velocity difference along the z-axis	m/s
v	velocity along the y-axis	m/s
w	velocity along the x-axis	m/s

Aerodynamics

Re	Reynolds number	dimensionless
Pr	Prandtl number	dimensionless
t	time	s
m	mass	kg
M	Mach number	dimensionless
a	Sonic velocity	m/s
Cd	Drag coefficient	dimensionless
Cf	Skin friction coefficient	dimensionless
Cp	Pressure coefficient	dimensionless
phi	Potential function	varies

Thermodynamics etc.

T	Temperature	K
p	Pressure	kg/m^2 = Pa
E	Energy	kg m^2/s^2 = J
S	Entropy	J/K
h	Enthalpy	J/kg
Q	Heat	J
q	Heat per unit mass	J/kg
rho	Density	kg/m^3
Cp	Specific Heat @ constant pressure	J/(kg K)
Cv	Specific Heat @ constant volume	J/(kg K)
gamma	ratio Cp/Cv	dimensionless
R	Universal gas constant	J/(kg K)
mu	shear viscosity	(N s)/m^2 = kg/(m s)
nu	kinematic viscosity	m^2/s
kappa	heat conductivity	J/(K m s)

Considerations for limited character sets and computers

Another problem which arises is the transliteration of various scripts. Mathematical notation is very free in the use of symbols, while many computer codes allow only alphanumeric names and the underscore. A suggested style is to approach the native notation as closely as possible. For instance Re for R-sub-e. While forms that have no easy representation can be spelled out, "gamma" for instance, is preferred over substituting a simple "g". A useful convention is to write pi when a lowercase pi is meant and Pi when an uppercase is intended.

Further, in writing computer programs, often, many separate sets of similar symbols need to be maintained simultaneously. The ideal solution may be to use scoping rules to avoid conflict and ambiguity but when this is not convenient, adding contextual information to a symbol resolves the problem. As a simple example, suppose that the rocket's x-coordinate is computed with data exclusively from the rocket and then augmented with ground information and recomputed from within the same computer program. The variables might be named thus:

x_rocket
x_ground

In general i suggest the convention of starting with the most specific information and proceeding to the more general. So if the flight computer's IMU system has a redundant x-axis accelerometer, while, say, a payload IMU is operating aboard the same rocket, a computer code might refer to the signal as something like:

ax_secondary_FC_IMU   (Because IMU_FC is too gross)

Notice that some data-hiding scheme is probably better than using very long names, but this is still a workable convention. Also note the casually-slick technique of using ax to represent the second derivative of position, vx would be velocity. This kind of abbreviation is desirable because it is close enough to common usage to be readily understandable, while its compactness reduces the chances for error. In general being longer than you have to is bad, but being too short is worse.