Trajectory Modeling in the Flight Path System

One form of the relative flight path coordinates can be defined as follows;

$$\begin{aligned} r &= \text{ geocentric radius} \\ V &= \text{ speed} \\ \gamma &= \text{ flight path angle} \\ \delta &= \text{ geocentric declination} \\ \lambda &= \text{ geographic longitude } (+\text{ east}) \\ \psi &= \text{ flight azimuth } (+\text{ clockwise from north}) \end{aligned}$$

Please note the sign and direction convention.

The first order equations of motion in this flight path coordinate system are as follows:

Geocentric radius

$$\dot{r} = \frac{dr}{dt} = v \sin \gamma$$

Longitude

$$\dot{\lambda} = \frac{d\lambda}{dt} = v \frac{\cos \gamma \sin \psi}{r \cos \delta}$$

Geocentric declination

$$\dot{\delta} = \frac{d\delta}{dt} = v \frac{\cos \gamma \cos \psi}{r}$$

Speed

$$\dot{V} = \frac{dV}{dt} = \frac{(T \cos \alpha - D)}{m} - g \sin \gamma + \omega_e^2 r \cos \delta \left(\sin \gamma \cos \delta - \sin \delta \cos \gamma \cos \psi \right)$$

Flight path angle

$$\begin{aligned} \dot{\gamma} = \frac{d\gamma}{dt} = \frac{V}{r} \cos \gamma + \left(\frac{T \sin \alpha + L}{mV} \right) \cos \beta - \frac{g \cos \gamma}{V} \\ + 2\omega_e \sin \psi \cos \delta + \omega_e^2 \frac{r}{V} \cos \delta \left(\cos \psi \sin \gamma \sin \delta + \cos \gamma \cos \delta \right) \end{aligned}$$

Flight azimuth

$$\begin{aligned} \dot{\psi} = \frac{d\psi}{dt} = \frac{V}{r} \tan \delta \sin \psi \cos \gamma + \left(\frac{T \sin \alpha + L}{mV \cos \gamma} \right) \cos \beta \\ + 2\omega_e \left(\sin \delta - \cos \psi \cos \delta \tan \gamma \right) + \frac{r}{V \cos \gamma} \omega_e^2 \sin \psi \cos \delta \sin \delta \end{aligned}$$

where

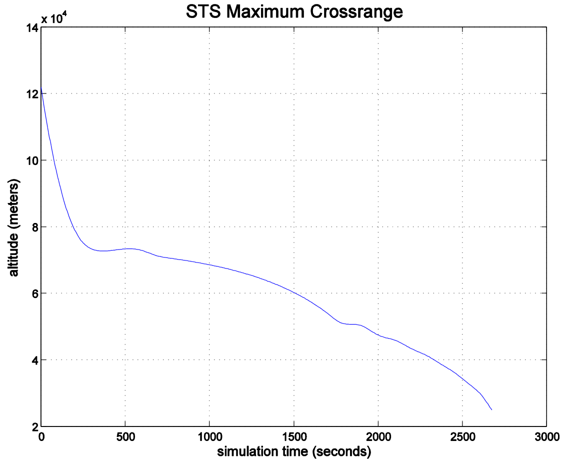
$$
\begin{aligned}
r &= \text{geocentric radius} \\
V &= \text{speed} \\
\gamma &= \text{flight path angle} \\
\delta &= \text{geocentric declination} \\
\lambda &= \text{longitude } (+\text{ east}) \\
\psi &= \text{flight azimuth } (+\text{ clockwise from north}) \\
\beta &= \text{bank angle } (+\text{ for a right turn}) \\
\alpha &= \text{angle of attack} \\
\omega_e &= \text{Earth inertial rotation rate} \\
g &= \text{Earth acceleration of gravity } = \mu/r^2 \\
\mu &= \text{Earth gravitational constant} \\
L &= \text{aerodynamic lift force } = \frac{1}{2}\rho V^2 C_L S \\
D &= \text{aerodynamic drag force } = \frac{1}{2}\rho V^2 C_D S \\
T &= \text{propulsive thrust} \\
m &= \text{spacecraft mass} \\
C_L &= \text{lift coefficient (non-dimensional)} \\
C_D &= \text{drag coefficient (non-dimensional)} \\
S &= \text{aerodynamic reference area} \\
\rho &= \text{atmospheric density}
\end{aligned}
$$

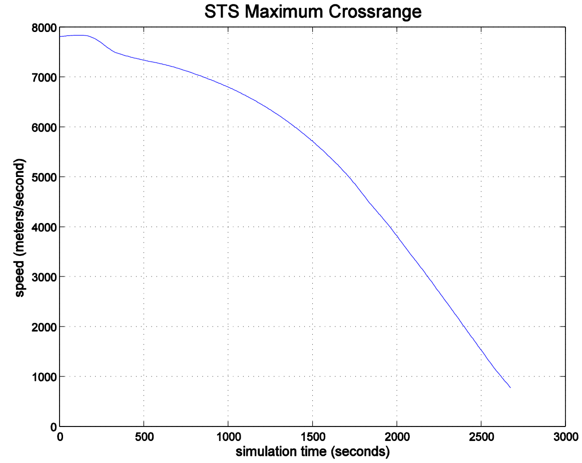
The following describes a MATLAB script named `demo_fpeqm` which demonstrates how to model a trajectory in the flight path system. This example flies an STS maximum cross range re-entry trajectory using angle-of-attack and bank information extracted from a trajectory optimization program.

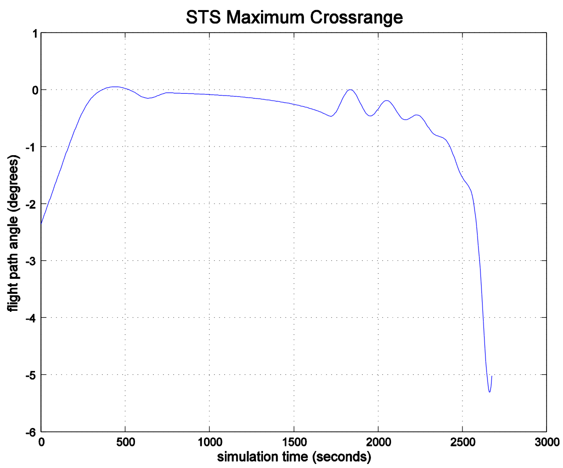
The following is the program output created by this script along with several graphic displays of flight parameters.

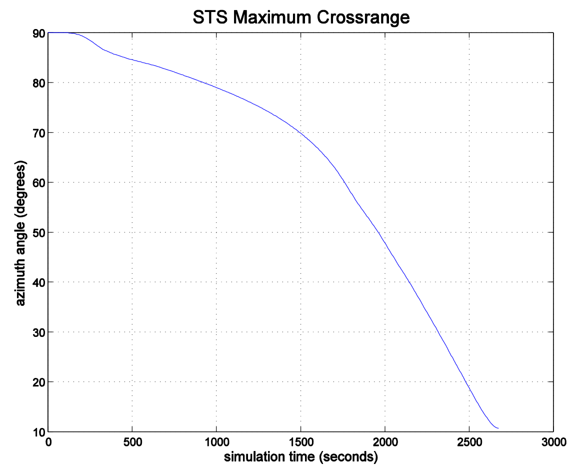
```
program demo_fpeqm

initial flight path coordinates
-------------------------------

altitude               121920.0000 meters
velocity                 7802.8800 meters/second
declination                 0.0000 degrees
longitude                   0.0000 degrees
azimuth                    90.0000 degrees
flight path angle          -2.3978 degrees


final flight path coordinates
-----------------------------
```

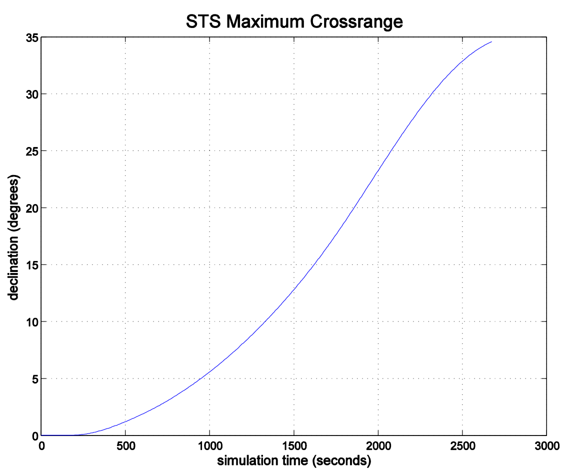
```
altitude               24518.4877 meters
velocity                 770.5555 meters/second
declination               34.5880 degrees
longitude                120.7278 degrees
azimuth                   10.6927 degrees
flight path angle         -5.0200 degrees
```

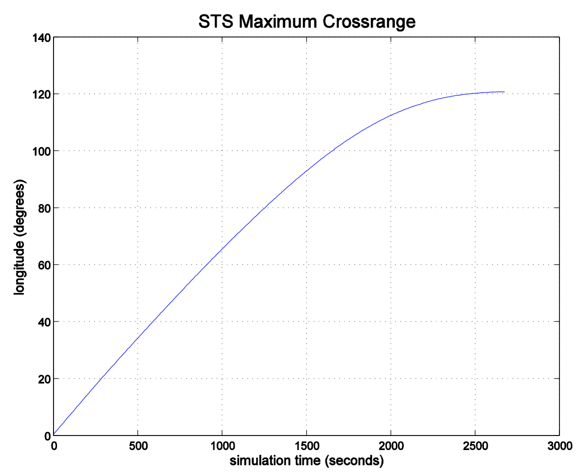












The following is a summary of the major MATLAB support functions for this script.

fpeqms.m – flight path equations of motion

This MATLAB function computes the first-order form of the equations of motion in the flight path coordinate system. These equations include the effects of aerodynamic lift and drag, and gravity due to a spherical Earth.

The syntax of this MATLAB function is

```
function ydot = fpeqms(t, y)

% flight path equations of motion

% includes lift, drag, spherical Earth
% gravity and Earth rotation

% input

% state variables

%  y(1) = altitude (kilometers)
%  y(2) = longitude (radians)
%  y(3) = geocentric declination (radians)
%  y(4) = relative velocity (kilometers/second)
%  y(5) = relative flight path angle (radians)
%  y(6) = relative azimuth (radians)

% output

%  ydot = equations of motion

% global

%  req   = Earth equatorial radius (kilometers)
%  mu    = Earth gravitational constant (km**3/sec**2)
%  omega = Earth inertial rotation rate (radians/second)
%  mass  = vehicle mass (kilograms)
%  sref  = aerodynamic reference area (cubic kilometers)
```

us76.m – 1976 U.S. Standard Atmosphere – analytic algorithm

This MATLAB function calculates atmospheric density using an analytic implementation of the 1976 U. S. Standard Atmosphere. It is based on the classic "U.S. Standard Atmosphere, 1976" document published by NOAA (NOAA-S/T 76-1562).

This function requires initialization the first time it is called. The following statement in the main MATLAB script will accomplish this:

```
iatmos = 1;
```

This variable should also be placed in a `global` statement at the beginning of the main script which calls this function. After the first call, the function will set this value to `0`.

The syntax of this MATLAB function is

```
function [atmtmp, atmdns, atmprs, atmsos] = us76 (alt)

% U.S. 1976 standard atmosphere

% input

%  alt = altitude (kilometers)

% output

%  atmtmp = temperature (degrees Kelvin)
%  atmdns = density (kilograms/cubic meter)
%  atmprs = pressure (newton/square meter)
%  atmsos = speed of sound (meters/second)
```

atmos76.dat – 1976 U. S. Standard atmosphere data file

This ASCII data file contains density values based the 1976 U. S. Standard atmosphere. The unit of this data is kilograms/cubic kilometer, data is valid from 0 to 1000 kilometers, and the altitude interval is 500 meters. The following are the first few lines of data in this file.

```
.1225000E+10
.1167273E+10
.1111660E+10
.1058104E+10
.1006554E+10
.9569545E+09
.9092544E+09
.8634021E+09
.8193470E+09
.7770388E+09
.7364289E+09
.6974689E+09
.6601116E+09
.6243101E+09
.5900187E+09
```

Relationship between inertial and relative flight path coordinates

The following are several useful equations that summarize the relationships between inertial and relative flight path coordinates.

$$v_r \sin\gamma_r = v_i \sin\gamma_i$$

$$v_r \cos\gamma_r \cos\psi_r = v_i \cos\gamma_i \cos\psi_i$$

$$v_r \cos\gamma_r \sin\psi_r + \omega_e r \cos\delta = v_i \cos\gamma_i \sin\psi_i$$

where the *r* subscript denotes relative coordinates and the *i* subscript inertial coordinates.

The inertial speed can also be computed from the following expression

$$v_i = \sqrt{v^2 + 2v\,r\omega\cos\gamma\sin\psi\cos\delta + r^2\omega^2\cos^2\delta}$$

The inertial flight path angle can be computed from

$$\cos\gamma_i = \sqrt{\frac{v^2\cos^2\gamma + 2vr\omega\cos\gamma\cos\psi\cos\delta + r^2\omega^2\cos^2\delta}{v^2 + 2vr\omega\cos\gamma\cos\psi\cos\delta + r^2\omega^2\cos^2\delta}}$$

The inertial azimuth can be computed from

$$\cos\psi_i = \frac{v\cos\gamma\cos\psi + r\omega\cos\delta}{\sqrt{v^2\cos^2\gamma + 2vr\omega\cos\gamma\cos\psi\cos\delta + r^2\omega^2\cos^2\delta}}$$

where all coordinates on the right-hand-side of these equations are relative to a rotating Earth.