FLIGHT PATH COORDINATES

This section describes several MATLAB functions that can be used to convert between relative (Earth-fixed) flight path coordinates and inertial position and velocity vectors. The relative flight path coordinates used in these functions are

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

Please note the sign and direction convention.

The 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.

eci2fpc1.m – convert ECI state vector to flight path coordinates

This MATLAB function convert an ECI state vector consisting of a position and velocity vector to relative flight path coordinates.

The syntax of this MATLAB function is

```
function fpc = eci2fpc1(gast, reci, veci)

% convert inertial state vector to flight path coordinates

% input

%  gast = greenwich apparent sidereal time (radians)
%  reci = inertial position vector (kilometers)
%  veci = inertial velocity vector (kilometers/second)

% output

%  fpc(1) = east longitude (radians)
%  fpc(2) = geocentric declination (radians)
%  fpc(3) = flight path angle (radians)
%  fpc(4) = azimuth (radians)
%  fpc(5) = position magnitude (kilometers)
%  fpc(6) = velocity magnitude (kilometers/second)
```

The transformation of an Earth-centered inertial (ECI) position vector $\mathbf{r}_{ECI}$ to an Earth-centered fixed (ECF) position vector $\mathbf{r}_{ECF}$ is given by the following vector-matrix operation

$$\mathbf{r}_{ECF} = [\mathbf{T}]\mathbf{r}_{ECI}$$

where the elements of the transformation matrix $[\mathbf{T}]$ are given by

$$[\mathbf{T}] = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and θ is the Greenwich apparent sidereal time at the moment of interest. Greenwich sidereal time is given by the following expression:

$$\theta = \theta_{g0} + \omega_e t$$

where θ_{g0} is the Greenwich sidereal time at 0 hours UT, ω_e is the inertial rotation rate of the Earth, and t is the elapsed time since 0 hours UT.

Finally, the flight path coordinates are determined from the following set of equations

$$r = \sqrt{r_{ECF}^2 + r_{ECF_y}^2 + r_{ECF}^2}$$

$$v = \sqrt{v_{ECF}^2 + v_{ECF_y}^2 + v_{ECF_z}^2}$$

$$\lambda = \tan^{-1}\left(r_{ECF_y}, r_{ECF_x}\right)$$

$$\delta = \sin^{-1}\left(\frac{r_{ECF_z}}{\left|\mathbf{r}_{ECF}\right|}\right)$$

$$\gamma = \sin^{-1}\left(-\frac{v_{R_z}}{\left|\mathbf{v}_R\right|}\right)$$

$$\psi = \tan^{-1}\left[v_{R_y}, v_{R_x}\right]$$

where

$$\mathbf{v}_R = \begin{bmatrix} -\sin\delta\cos\lambda & -\sin\delta\sin\lambda & \cos\delta \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\delta\cos\lambda & -\cos\delta\sin\lambda & -\sin\delta \end{bmatrix} \mathbf{v}_{ECF}$$

Please note that the two argument inverse tangent calculation is a four quadrant operation.

eci2fpc2.m – convert ECI state vector to flight path coordinates

This MATLAB function computes the relative flight path coordinates by first converting the ECI state vector to ECF coordinates and then to flight path coordinates. The following is the source code that performs these calculations.

```
% compute ecf state vector

[recf, vecf] = eci2ecf(gast, reci, veci);

% compute relative flight path coordinates

fpc = rv2fpc(recf, vecf);
```

The syntax of this MATLAB function is

```
function fpc = eci2fpc2(gast, reci, veci)

% convert inertial state vector to relative flight path coordinates

% input

%  gast = greenwich apparent sidereal time (radians)
%  reci = inertial position vector (kilometers)
%  veci = inertial velocity vector (kilometers/second)
```

```
% output

%  fpc(1) = east longitude (radians)
%  fpc(2) = geocentric declination (radians)
%  fpc(3) = flight path angle (radians)
%  fpc(4) = azimuth (radians)
%  fpc(5) = position magnitude (kilometers)
%  fpc(6) = velocity magnitude (kilometers/second)
```

The syntax of the MATLAB function that converts the state vector to flight path coordinates is

```
function fpc = rv2fpc (r, v)

% transform from cartesian coordinates
% to flight path coordinates

% input

%  r = eci or ecf position vector
%  v = eci or ecf velocity vector

% output

%  fpc(1) = geocentric right ascension or east longitude (radians)
%  fpc(2) = geocentric declination (radians)
%  fpc(3) = flight path angle (radians)
%  fpc(4) = azimuth (radians)
%  fpc(5) = position magnitude (kilometers)
%  fpc(6) = velocity magnitude (kilometers/second)

% NOTE: if ECI input, output is inertial
%       if ECF input, output is Earth relative
```

This software suite contains a MATLAB script named `demo_eci2fpc` that demonstrates how to interact with both of these functions. The input required by this script is "hardwired" within the source code. The following is the output created by this script.

```
eci2fpc1 function
=================

UTC julian date          2458337.83361944


inertial coordinates
--------------------

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 -5.86479273288000e+003  -1.78173078828000e+003  -2.15629990858000e+003  +6.49769095119867e+003

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 +4.92973713131000e-001  -7.31828662424000e+000  +8.19193017342000e+000  +1.09958202132700e+001


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

east longitude             121.00000000  degrees
```

```
geocentric declination      -19.38148629  degrees

flight path angle            -6.20000000  degrees

relative azimuth             38.98298719  degrees

position magnitude         6497.69095120  kilometers

velocity magnitude           10.71074956  kilometers/second

geodetic coordinates
--------------------

geodetic latitude           -19.50000000  degrees

geodetic altitude           121.92000000  kilometers

eci2fpc2 function
=================

UTC julian date           2458337.83361944

inertial coordinates
--------------------

        rx (km)                 ry (km)                 rz (km)                 rmag (km)
 -5.86479273288000e+003  -1.78173078828000e+003  -2.15629990858000e+003  +6.49769095119867e+003

        vx (kps)                vy (kps)                vz (kps)                vmag (kps)
 +4.92973713131000e-001  -7.31828662424000e+000  +8.19193017342000e+000  +1.09958202132700e+001

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

east longitude              121.00000000  degrees

geocentric declination      -19.38148629  degrees

flight path angle            -6.20000000  degrees

relative azimuth             38.98298719  degrees

position magnitude         6497.69095120  kilometers

velocity magnitude           10.71074956  kilometers/second

geodetic coordinates
--------------------

geodetic latitude           -19.50000000  degrees

geodetic altitude           121.92000000  kilometers
```

fpc2eci.m – convert flight path coordinates to ECI state vector

This MATLAB function transforms relative flight path coordinates to an ECI state vector (position and velocity vectors).

The syntax of this MATLAB function is

```
function [reci, veci] = fpc2eci(gst, fpc)
```

```
% transform relative flight path coordinates to inertial state vector

% input

%  gst    = greenwich apparent sidereal time (radians)
%  fpc(1) = east longitude (radians)
%  fpc(2) = geocentric declination (radians)
%  fpc(3) = relative flight path angle (radians)
%  fpc(4) = relative azimuth (radians)
%  fpc(5) = position magnitude (kilometers)
%  fpc(6) = relative velocity magnitude (kilometers/second)

% output

%  reci = inertial position vector (kilometers)
%  veci = inertial velocity vector (kilometers/second)
```

This function performs this conversion by first calculating the ECF state vector and then the ECI state vector using the following MATLAB source code:

```
% compute ecf state vector

[recf, vecf] = fpc2ecf(fpc);

% compute eci state vector

[reci, veci] = ecf2eci(gst, recf, vecf);
```