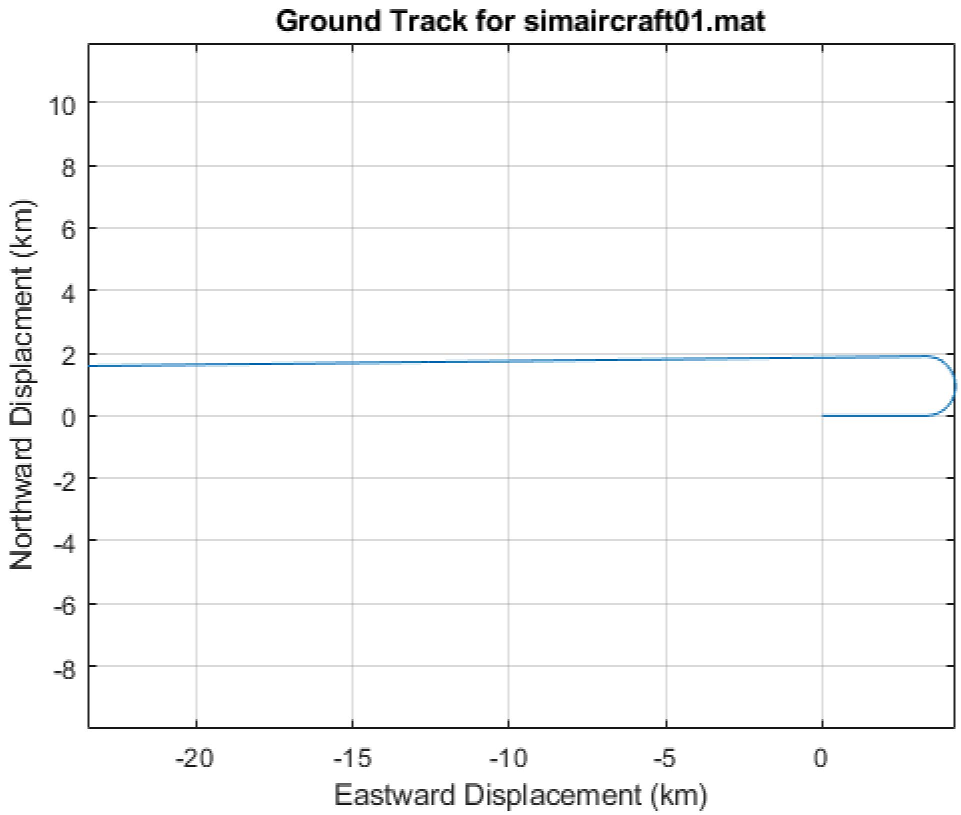
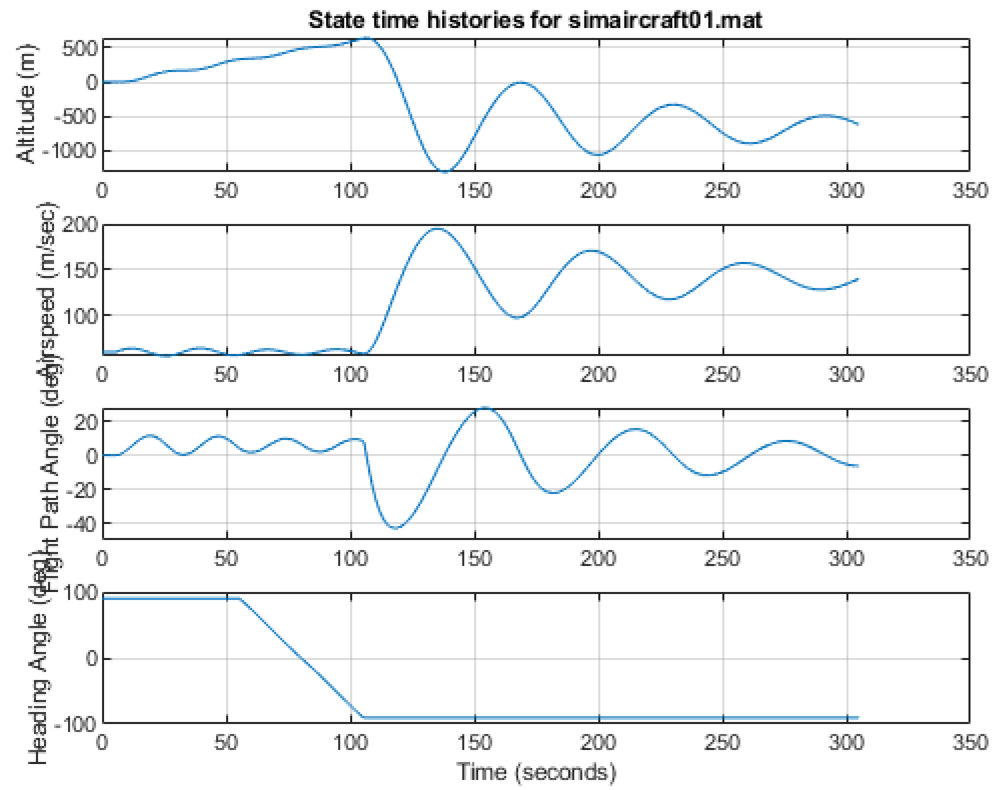
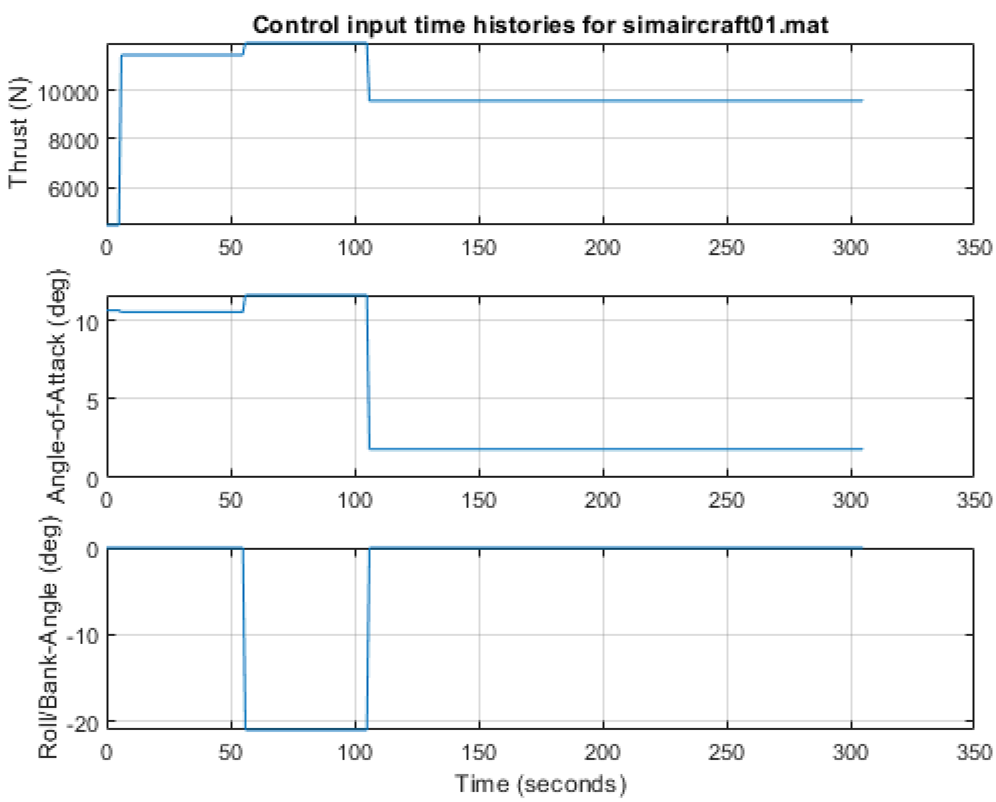
```
 1.0e+04 *

 0.158647266765228
-2.346841979945498
 0.061558912737074
 0.013988616814899
-0.000011110309497
-0.000158233102664
```



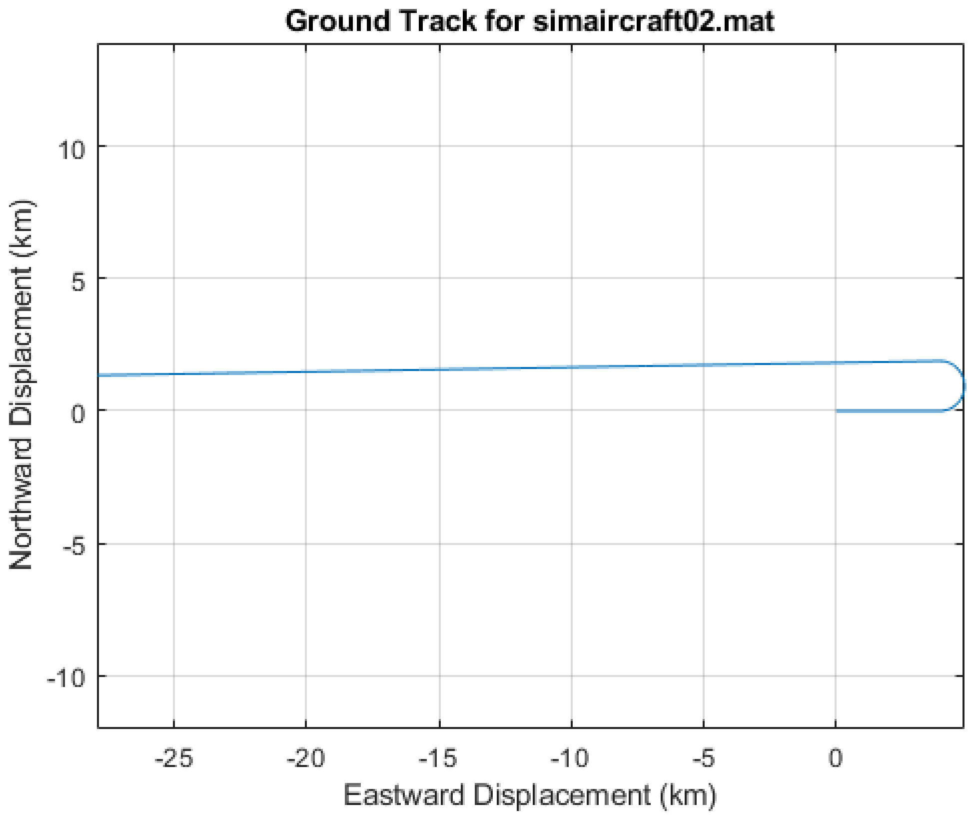


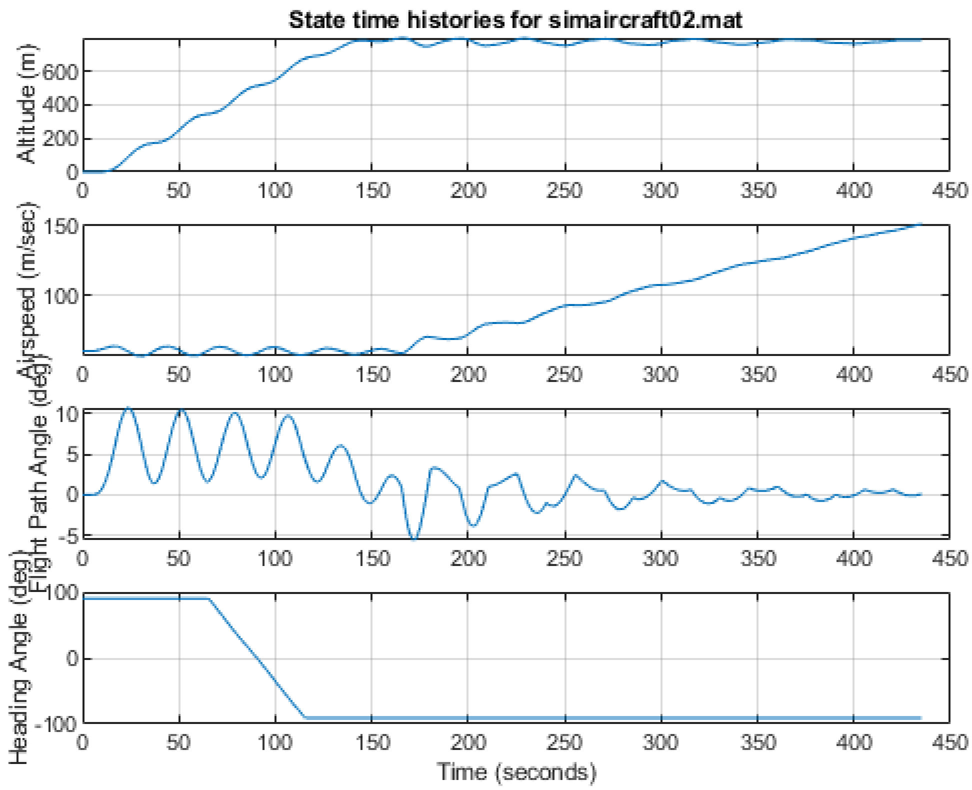
State time histories for simaircraft01.mat
Altitude (m)
Airspeed (m/sec)
Flight Path Angle (deg)
Heading Angle (deg)
Time (seconds)

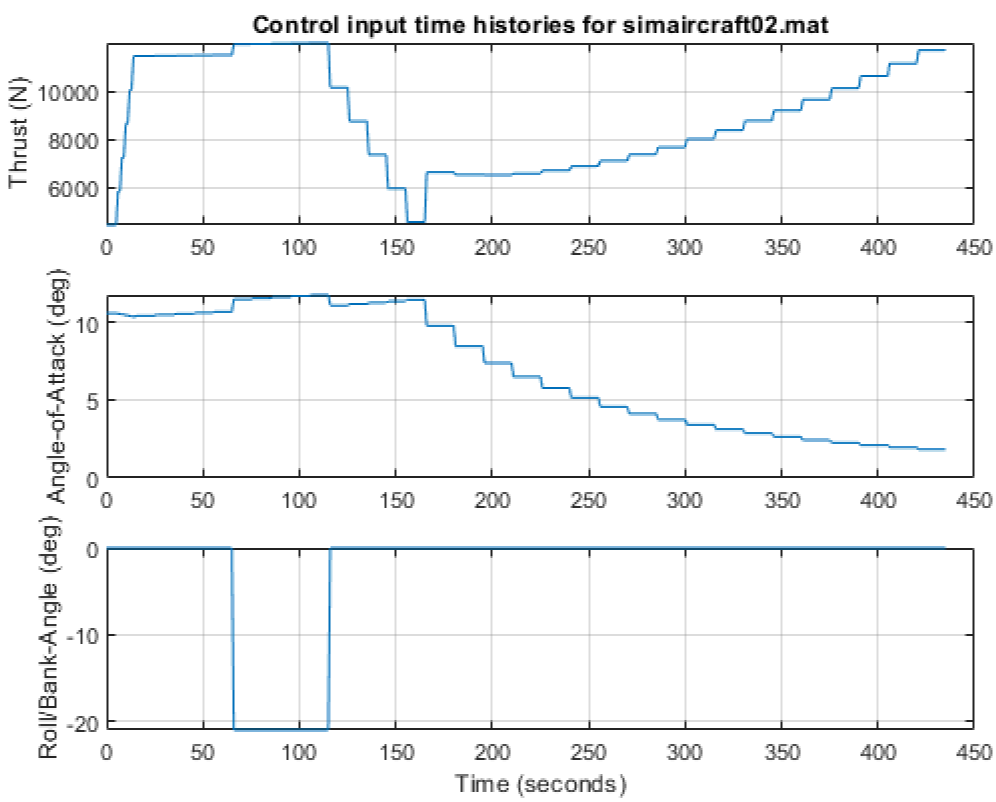


Control input time histories for simaircraft01.mat
Thrust (N)
Angle-of-Attack (deg)
Roll/Bank-Angle (deg)
Time (seconds)

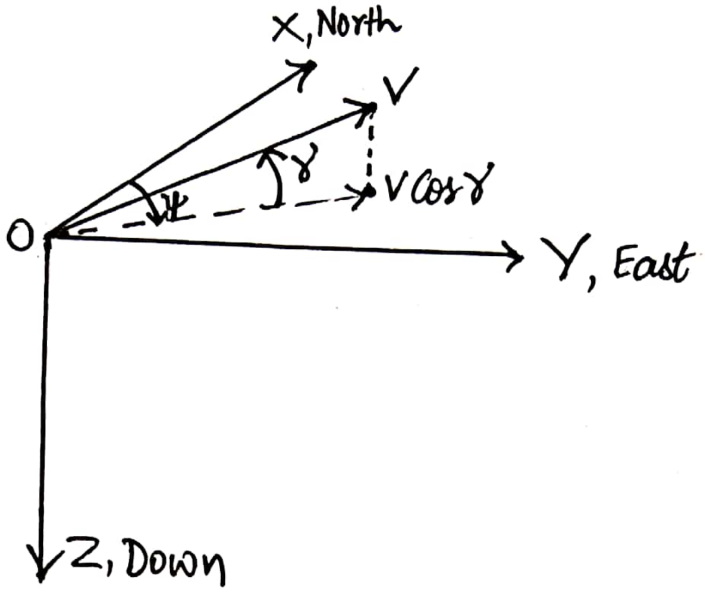
```
 1.0e+04 *

 0.134544732984597
-2.792795344349624
-0.078497666765006
 0.015137558915019
 0.000000207783870
-0.000158784097156
```




State time histories for simaircraft02.mat
Altitude (m)
Airspeed (m/sec)
Flight Path Angle (deg)
Heading Angle (deg)
Time (seconds)


Control input time histories for simaircraft02.mat
Thrust (N)
Angle-of-Attack (deg)
Roll/Bank-Angle (deg)
Time (seconds)

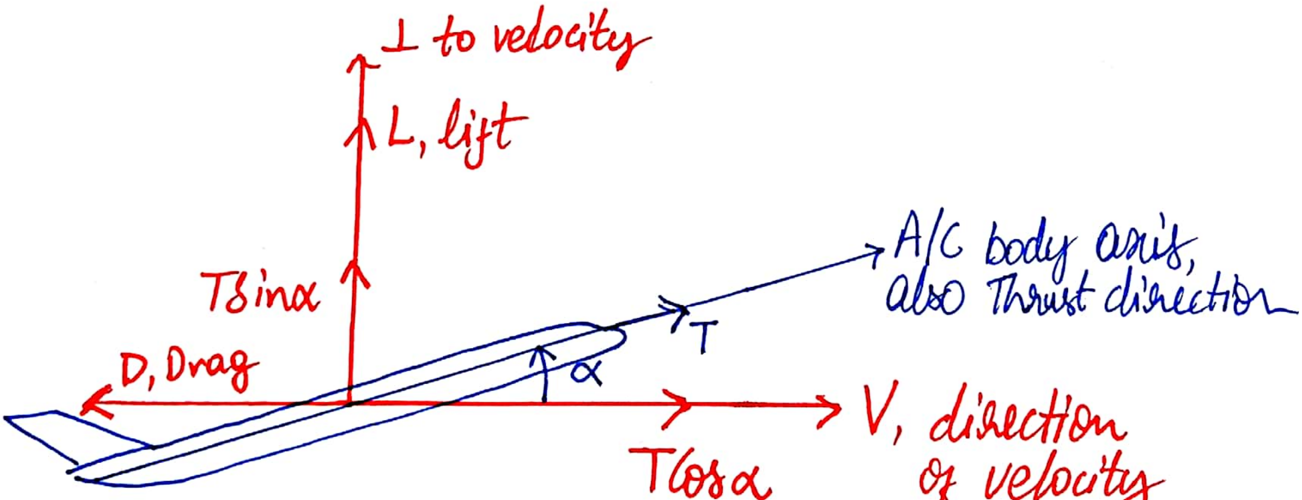
③ The Velocity(V), γ (Elevation), ψ (heading) defined in the North-East-down co-ord Sys as follows:



$$\dot{X} = V \cos\gamma \cos\psi$$
$$\dot{Y} = V \cos\gamma \sin\psi$$
$$\dot{Z} = -V \sin\gamma$$

Kinematic Equations.

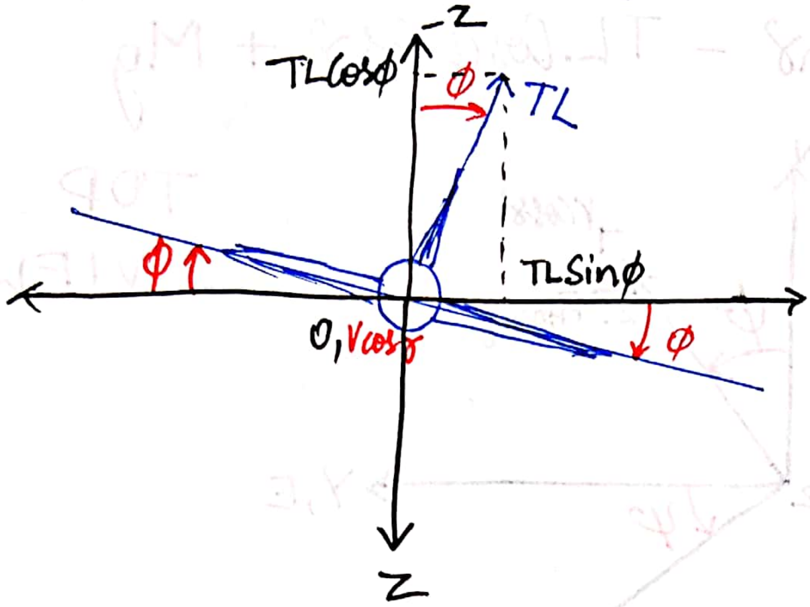
Resolution the thrust vectos along & perpendicular to the velocity direction



definitions:

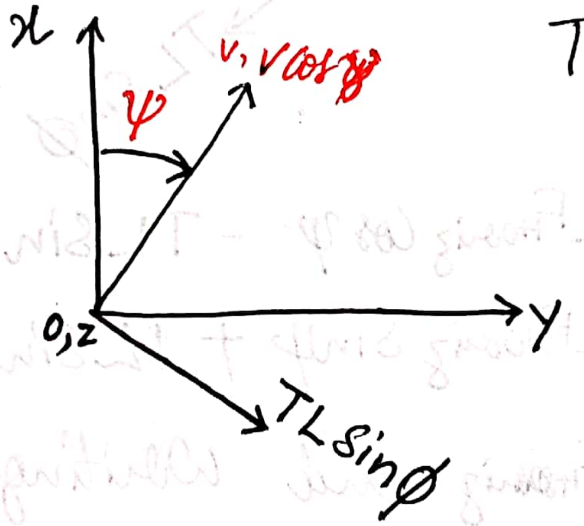
$$TL \triangleq T\sin\alpha + L \quad (\perp \text{ to velocity})$$
$$TD \triangleq T\cos\alpha - D \quad (\text{along the velocity})$$

Impact of A/C bank angle about V-axis, ϕ



LOOKING FROM TAIL of A/C

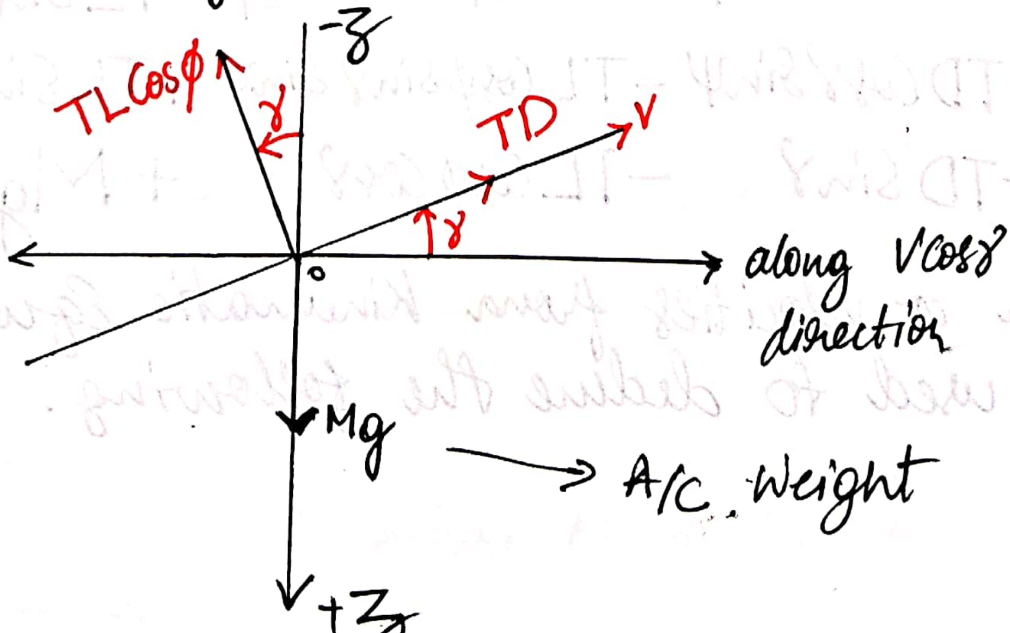
Horizontal XY plane



TOP VIEW

The convention of ϕ is; increasing $\dot{\psi}$ with a positive ϕ

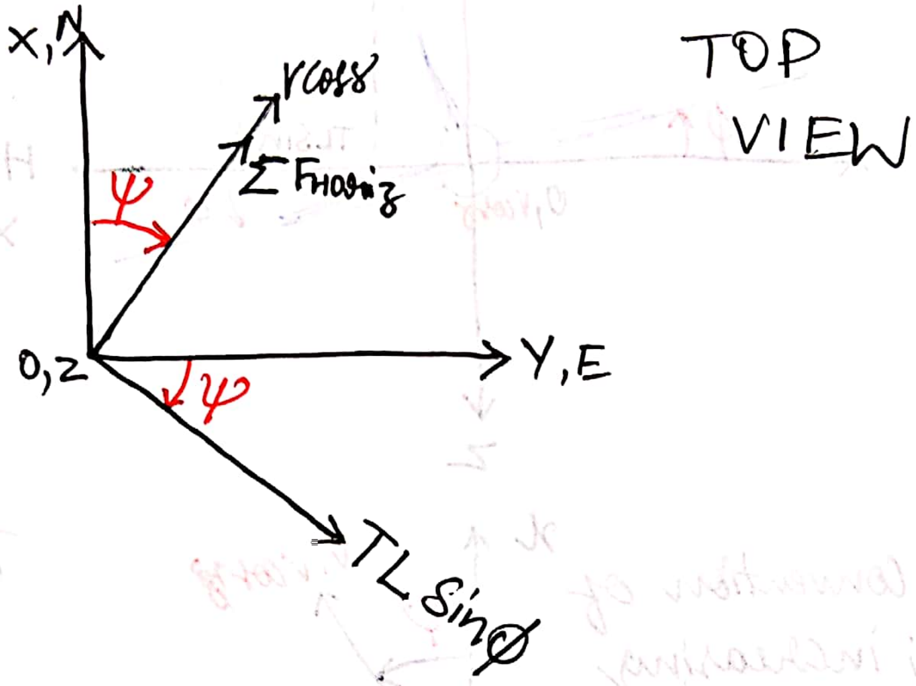
The force are now resolved 1st in the plane containing O, V, V Cos γ



→ A/C. Weight

$$\sum F_{Horiz} = TD\cos\gamma - TL\cos\phi\sin\gamma$$

$$\sum F_z = -TD\sin\gamma - TL\cos\phi\cos\gamma + Mg$$



$$\sum F_x = \sum F_{Horiz}\cos\psi - TL\sin\phi\sin\psi$$

$$\sum F_y = \sum F_{Horiz}\sin\psi + TL\sin\phi\cos\psi$$

Expanding $\sum F_{Horiz}$ and writing all three components

$$\sum F_x = TD\cos\gamma\cos\psi - TL\cos\phi\sin\gamma\cos\psi - TL\sin\phi\sin\psi$$

$$\sum F_Y = TD\cos\gamma\sin\psi - TL\cos\phi\sin\gamma\sin\psi + TL\sin\phi\cos\psi$$

$$\sum F_z = -TD\sin\gamma - TL\cos\phi\cos\gamma + Mg$$

Definition of velocities from Kinematic Equation Can be used to deduce the following.

$$\dot{X}^2+\dot{Y}^2+\dot{Z}^2 = V^2 \quad \text{—— ①}$$

$$\sin\gamma = \frac{-\dot{Z}}{V} \quad \text{—— ②}$$

$$\sin^2\gamma = \frac{\dot{Z}^2}{V^2} \Rightarrow \cos^2\gamma = 1-\frac{\dot{Z}^2}{V^2} = \frac{V^2-\dot{Z}^2}{V^2}$$

$$\cos\gamma = \frac{\sqrt{\dot{X}^2+\dot{Y}^2}}{V} \quad \text{—— ③}$$

$$V\cos\gamma = \sqrt{\dot{X}^2+\dot{Y}^2}$$

$$\dot{X} = \sqrt{\dot{X}^2+\dot{Y}^2}\cos\psi \quad \& \quad \dot{Y} = \sqrt{\dot{X}^2+\dot{Y}^2}\sin\psi$$

$$\cos\psi = \frac{\dot{X}}{\sqrt{\dot{X}^2+\dot{Y}^2}} \; ; \quad \sin\psi = \frac{\dot{Y}}{\sqrt{\dot{X}^2+\dot{Y}^2}} \quad \text{—— ④ \& ⑤}$$

Rewriting the dynamics without $\cos\gamma$, $\sin\gamma$, $\cos\psi$ & $\sin\psi$

$$\boxed{\begin{aligned}
\sum F_x &= TD\frac{\dot{X}}{V} + TL\left(\cos\phi\frac{\dot{Z}\dot{X}}{V\sqrt{\dot{X}^2+\dot{Y}^2}} - \sin\phi\frac{\dot{Y}}{\sqrt{\dot{X}^2+\dot{Y}^2}}\right)\\
\sum F_Y &= TD\frac{\dot{Y}}{V} + TL\left(\cos\phi\frac{\dot{Z}\dot{Y}}{V\sqrt{\dot{X}^2+\dot{Y}^2}} + \sin\phi\frac{\dot{X}}{\sqrt{\dot{X}^2+\dot{Y}^2}}\right)\\
\sum F_Z &= TD\frac{\dot{Z}}{V} - TL\left(\cos\phi\frac{\sqrt{\dot{X}^2+\dot{Y}^2}}{V}\right) + Mg
\end{aligned}}$$

$$M\ddot{X} = \sum F_X$$
$$M\ddot{Y} = \sum F_Y$$
$$M\ddot{Z} = \sum F_Z$$

where M is the A/C Mass.

~~TR~~ Aerodynamic forces

$$TL = T\sin\alpha + L$$

$$L = \tfrac{1}{2}\rho V^2 S C_l$$

$$C_l = C_{L_\alpha}\alpha$$

Similarly

$$D = \tfrac{1}{2}\rho V^2 S C_D$$

$$C_D = C_{D_0} + \frac{C_L^2}{\pi e AR}$$

$$C_D = C_{D_0} + \frac{C_{L_\alpha}^2}{\pi e AR}\alpha^2$$

$\rho \rightarrow$ density of free stream

$S \rightarrow$ Wing reference Area.

$C_{L_\alpha} \rightarrow$ Lift curve slope

$C_{D_0} \rightarrow$ Zero-lift Drag coefficient

$e \rightarrow$ Oswald Efficiency

$AR \rightarrow$ Aspect-ratio

$\therefore$

$$TL = T\sin\alpha + \tfrac{1}{2}\rho V^2 S C_{L_\alpha}\alpha$$

$$TD = T\cos\alpha - \tfrac{1}{2}\rho V^2 S\left(C_{D_0} + \frac{C_{L_\alpha}^2}{\pi e AR}\alpha^2\right)$$

The final dynamics is as follows

$$M\ddot{X} = TD\frac{\dot{X}}{V} + TL\left(\cos\phi\frac{\dot{Z}\dot{X}}{V\sqrt{\dot{X}^2+\dot{Y}^2}} - \sin\phi\frac{\dot{Y}}{\sqrt{\dot{X}^2+\dot{Y}^2}}\right)$$

$$M\ddot{Y} = TD\frac{\dot{Y}}{V} + TL\left(\cos\phi\frac{\dot{Z}\dot{Y}}{V\sqrt{\dot{X}^2+\dot{Y}^2}} + \sin\phi\frac{\dot{X}}{\sqrt{\dot{X}^2+\dot{Y}^2}}\right)$$

$$M\ddot{Z} = TD\frac{\dot{Z}}{V} - TL\left(\cos\phi\frac{\sqrt{\dot{X}^2+\dot{Y}^2}}{V}\right) + Mg$$

where, the 3 of the 4 forces T, L, D are captured in Expressions for TD & TL

$$TL = T\sin\alpha + \tfrac{1}{2}\rho V^2 S C_{L_\alpha}\alpha$$

$$TD = T\cos\alpha - \tfrac{1}{2}\rho V^2 S\left(C_{D_0} + \frac{C_{L_\alpha}^2}{\pi e AR}\alpha^2\right)$$

where, $V=\sqrt{\dot{X}^2+\dot{Y}^2+\dot{Z}^2}$

The weight, 'Mg' is always acting in the downward direction so, it is in-line with our definition of +Z in the NED Co-ord System. So, it just shows up as a term in $\ddot{Z}$ Equation.

```
function f = ffunctaircraft01(t,x,m,S,CLalpha,CD0,oneoverpiARe,...
                              tinhist,Thist,alphahist,phihist)
%
%  Copyright (c) 2019 Mark L. Psiaki.  All rights reserved.
%
%  This function implements a nonlinear dynamic model
%  of a point-mass airplane flying over a flat Earth
%  in an atmosphere whose air density decays exponentially
%  with altitude.  This function models the effects of
%  time-varying thrust, angle-of-attack, and roll/bank-angle
%  inputs.
%
%  The dynamic model takes the form:
%
%    xdot = f(t,x)
%
%  where xdot is the time rate of change of the 6-by-1
%  state vector x and where the 6-by-1 vector function
%  f(t,x) is the output of this Matlab function.
%
%  Note:  The aerodynamic model does not include stall.
%
%
%  Inputs:
%
%    t                      The time, in seconds, at which f is
%                           to be computed.
%
%    x                      = [X;Y;Z;V;gamma;psi], the 6-by-1 state
%                           vector of this system.  The first three
%                           elements give the Cartesian position
%                           vector of the aircraft's center of
%                           mass in local coordinates, in meters
%                           units, with X being the northward
%                           displacement from a reference position,
%                           Y being the eastward displacement from
%                           a reference position, and -Z being the
%                           altitude above sea level (so that
%                           a positive value of x(3,1) indicates
%                           flight below sea level, perhaps over
%                           Dead Sea).  The fourth element of x
%                           the the airspeed (and the inertial
%                           speed assuming no wind) in meters/second.
%                           The fifth element is the flight path
%                           angle in radians.  The sixth element is
%                           the heading angle in radians (0 is due
%                           north, +pi/2 radians is due east).
%
%    m                      The aircraft mass in kg.
%
%    S                      The wing area, in meters^2, which is
%                           the aerodynamic model's reference area.
%
%    CLalpha                The lift curve slope, dCL/dalpha, which
%                           is non-dimensional.
%
%    CD0                    The drag at zero lift, which is non-
```

```
%                           dimensional.
%
%   oneoverpiARe            = 1/(pi*AR*e), where AR is the non-
%                           dimensional aspect ratio of the wing
%                           and e is the Oswald efficiency factor.
%                           This composite input quantity is non-
%                           dimensional.  It is the coefficient
%                           of CL^2 in the drag coefficient model.
%
%   tinhist                 = [tin0;tin1;tin2;...;tinM], the
%                           (M+1)-by-1 vector of times, in seconds,
%                           at which the airplane control inputs in
%                           Thist, alphahist, and phihist are
%                           defined.  This must be a monotonically
%                           increasing vector.  Also, it is required
%                           that tinhist(1,1) = tin0 <= t <= tinM = ...
%                           tinhist(M+1,1).  Otherwise, an error
%                           condition will occur.
%
%   Thist                   = [T0;T1;T2;...;TM], the (M+1)-by-1 vector
%                           of thrust inputs that apply at the times
%                           in tinhist, in Newtons.
%
%   alphahist               = [alpha0;alpha1;alpha2;...;alphaM], the
%                           (M+1)-by-1 vector of angle-of-attack
%                           inputs that apply at the times in tinhist,
%                           in radians.
%
%   phihist                 = [phi0;phi1;phi2;...;phiM], the (M+1)-by-1
%                           vector of roll/bank-angle inputs that apply
%                           at the times in tinhist, in radians.
%
%                           Note: a piecewise cubic hermite
%                           interpolating polynomial is used
%                           to interpolate between times in tinhist
%                           in order to compute the thrust, angle-of-
%                           attack, and roll/bank angle that apply at
%                           time t.  These interpolations are computed
%                           using interp1.m.
%
%  Outputs:
%
%   f                       = [Xdot;Ydot;Zdot;Vdot;gammadot;psidot],
%                           the 6-by-1 vector that contains the
%                           computed time derivatives of the state
%                           from the kinematics and dynamics models
%                           of the aircraft.  f(1:3,1) is given
%                           in meters/second.  f(4,1) is given in
%                           meters/second^2, and f(5:6,1) is given
%                           in radians/second.
%


%
%  Compute the thrust, angle-of-attack, and roll/bank-angle
%  inputs that apply at time t.  It is more
%  efficient to do all three piecewise cubic hermite
%  interpolations simultaneously, as is done here.
```

```
%
   Talphaphi = interp1(tinhist,[alphahist,Thist,phihist],t,'pchip');
   alpha = Talphaphi(1,1);
   T = Talphaphi(1,2);
   phi = Talphaphi(1,3);
%
%  Compute the air density using a decaying exponential
%  model.  This model is good to about 1500 m altitude
%  (about 5000 ft).  This model recognizes that - x(3,1) is
%  the aircraft altitude above sea level in meters.
%
   rho_sealevel = 1.225; % kg/m^3
   hscale = 10230.;      % meters
   haltitude = -x(3);
   rho = rho_sealevel*exp(-haltitude/hscale); % kg/m^3
%
%  Determine the airspeed.
%
   V = x(4);
%
%  Determine the dynamic pressure.
%
   qbar = 0.5 .* rho .* V.*V;
%
%  Compute the lift and drag coefficients.
%
   CL = CLalpha.*alpha;
   CD = CD0+(CL.*CL.*oneoverpiARe);
%
%  Determine the lift and drag forces.
%
   L = qbar.*S.*CL;
   D = qbar.*S.*CD;
%
%  Set the flat-Earth gravitational acceleration.
%
   g = 9.81; % meters/second^2
%
%  Compute the kinematics part of the model.
%
   cosgamma = cos(x(5));
   singamma = sin(x(5));
   cospsi = cos(x(6));
   sinpsi = sin(x(6));
   V_cosgamma = V*cosgamma;
   Xdot =  V_cosgamma.*cospsi;
   Ydot =  V_cosgamma.*sinpsi;
   Zdot = -V.*singamma;
%
%  Compute the dynamics part of the model.
%
   cosalpha = cos(alpha);
   sinalpha = sin(alpha);
   cosphi = cos(phi);
   sinphi = sin(phi);
   oneoverm = 1/m;
   Vdot = (oneoverm .* (T*cosalpha-D)) - (g*singamma) ;
```

```
   T_sinalpha_plus_L = T*sinalpha + L;
   gammadot = (1/V) .* ( ( oneoverm .* cosphi .* T_sinalpha_plus_L ) -
(g*cosgamma) ) ;
   psidot   = (1/V_cosgamma) .* ( oneoverm .* sinphi .* T_sinalpha_plus_L ) ;
%
%  Assemble the computed state time derivative elements
%  into the output vector.
%
   f = [ Xdot ; Ydot ; Zdot ; Vdot ; gammadot ; psidot ];
```