
Table of Contents

.....	1
-------	---

```
function [Alon, Blon, Alat, Blat] = AircraftLinearModel(trim_definition,
aircraft_parameters)
% Takes as input the trim definition for straight wings level flight and an
% aircraft parameters structure. Calculates the linear matrices A and B for
% the lateral and longitudinal modes
% Inputs:
%   trim_definition -> full 12x1 aircraft state
%   aircraft_parameters -> inertial wind velocity in inertial coordinates
% Output:
%   Alon -> longitudinal linearized A matrix from trim
%   Blon -> longitudinal linearized B matrix from trim
%   Alat -> lateral linearized B matrix from trim
%   Blat -> lateral linearized B matrix from trim
%
% Author: Thomas Dunnington
% Date Modified: 9/23/2024

% Aircraft parameters
ap = aircraft_parameters;

% Get full state
[trim_state, trim_control] = TrimCalculator(trim_definition,
aircraft_parameters);

% Variables from trim definition
Va0 = trim_definition(1);
gamma0 = trim_definition(2);
h0 = trim_definition(3);

% Controls
de0 = trim_control(1);
da0 = trim_control(2);
dr0 = trim_control(3);
dt0 = trim_control(4);

% State
phi0 = trim_state(4);
theta0 = trim_state(5);
u0 = trim_state(7);
v0 = trim_state(8);
w0 = trim_state(9);

p0 = trim_state(10);
q0 = trim_state(11);
r0 = trim_state(12);
beta0 = asin(v0/Va0);
```

```

% Air density
rho = stdatmo(h0);

% Angle of attack
alpha0 = atan2(w0, u0);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Longitudinal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Lift and drag
qhat = (q0*ap.c)/(2*Va0);
CL = ap.CL0 + ap.CLalpha*alpha0 + ap.CLq*qhat + ap.CLde*de0;
CD = ap.CDmin + ap.K*(CL - ap.CLmin)^2;

%%% Trim values
CLtrim = CL;
CDtrim = CD;

% Body aerodynamic forces
CX = -CD*cos(alpha0) + CL*sin(alpha0);
CZ = -CD*sin(alpha0) - CL*cos(alpha0);

% Propulsion
CT = 2*ap.Sprop/ap.S * ap.Cprop * dt0/(Va0^2) * (Va0 + dt0*(ap.kmotor - Va0))*(ap.kmotor - Va0);

% Derivatives
CDalpha = 2*ap.CLalpha*ap.K*(CLtrim - ap.CLmin);
CDq = 2*ap.K*(CLtrim - ap.CLmin) * ap.CLq;
CDde = 2*ap.K*(CLtrim - ap.CLmin) * ap.CLde;

%%% Nondimensional stability derivatives in body coordinates
CDu = 0;%CDM*Ma;
CLu = 0;%CLM*Ma;
Cmu = 0;%CmM*Ma;

% Body force derivatives
CXalpha = CDtrim*sin(alpha0) + CLtrim*cos(alpha0) + ap.CLalpha*sin(alpha0) - CDalpha*cos(alpha0);
CXq = -CDq*cos(alpha0) + ap.CLq*sin(alpha0);
CXde = -CDde*cos(alpha0) + ap.CLde*sin(alpha0);

CZalpha = -CDalpha*sin(alpha0) - CDtrim*cos(alpha0) - ap.CLalpha*cos(alpha0) + CLtrim*sin(alpha0);
CZq = -ap.CLq;
CZde = -CDde*sin(alpha0) - ap.CLde*cos(alpha0);

% Inertia term
Jy = ap.Iy;

% Longitudinal dimensional stability derivatives
Xu = u0*rho*ap.S/ap.m * CX - rho*ap.S*w0*CXalpha/(2*ap.m) +

```

```

rho*ap.S*ap.c*CXq*u0*q0/(4*ap.m*Va0) + ...
    rho*ap.Sprop*ap.Cprop*dt0/ap.m * (ap.kmotor*u0/Va0 * (1-2*dt0) +
2*u0*(dt0 - 1));
Xw = -q0 + w0*rho*ap.S/ap.m * CX + rho*ap.S*ap.c*CXq*w0*q0/(4*ap.m*Va0) +
rho*ap.S*CXalpha*u0/(2*ap.m) + ...
    rho*ap.Sprop*ap.Cprop*dt0/ap.m * (ap.kmotor*w0/Va0 * (1-2*dt0) +
2*w0*(dt0 - 1));
Xq = -w0 + rho*Va0*ap.S*CXq*ap.c/(4*ap.m);
Xde = rho*Va0^2*ap.S*CXde/(2*ap.m);
Xdt = rho*ap.Sprop*ap.Cprop/ap.m * (Va0*(ap.kmotor - Va0) + 2*dt0*(ap.kmotor
- Va0)^2);

Zu = q0 + u0*rho*ap.S/ap.m * CZ - rho*ap.S*CZalpha*w0/(2*ap.m) +
u0*rho*ap.S*CZq*ap.c*q0/(4*ap.m*Va0);
Zw = w0*rho*ap.S/ap.m * CZ + rho*ap.S*CZalpha*u0/(2*ap.m) +
rho*w0*ap.S*ap.c*CZq*q0/(4*ap.m*Va0);
Zq = u0 + rho*Va0*ap.S*CZq*ap.c/(4*ap.m);
Zde = rho*Va0^2*ap.S*CZde/(2*ap.m);

Mu = u0*rho*ap.S*ap.c/Jy * (ap.Cm0 + ap.Cmalpha*alpha0 + ap.Cmde*de0) -...
    rho*ap.S*ap.c*ap.Cmalpha*w0/(2*Jy) + rho*ap.S*ap.c^2*ap.Cmq*q0*u0/
(4*Jy*Va0);
Mw = w0*rho*ap.S*ap.c/Jy * (ap.Cm0 + ap.Cmalpha*alpha0 + ap.Cmde*de0) +...
    rho*ap.S*ap.c*ap.Cmalpha*u0/(2*Jy) + rho*ap.S*ap.c^2*ap.Cmq*q0*w0/
(4*Jy*Va0);
Mq = rho*Va0*ap.S*ap.c^2*ap.Cmq/(4*Jy);
Mde = rho*Va0^2*ap.S*ap.c*ap.Cmde/(2*Jy);

% Longitudinal Matrices
Alon = [Xu, Xw*Va0*cos(alpha0), Xq, -ap.g*cos(theta0), 0;
    Zu/(Va0*cos(alpha0)), Zw, Zq/(Va0*cos(alpha0)), -ap.g*sin(theta0)/
(Va0*cos(alpha0)), 0;
    Mu, Mw*Va0*cos(alpha0), Mq, 0, 0;
    0, 0, 1, 0, 0;
    sin(theta0), -Va0*cos(theta0)*cos(alpha0), 0, u0*cos(theta0) +
w0*sin(theta0), 0];

Blon = [Xde, Xdt;
    Zde/(Va0*cos(alpha0)), 0;
    Mde, 0;
    0, 0;
    0, 0];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Lateral
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Inertia terms
gamma = InertiaTerms(ap);

% Stability Derivatives
Yv = rho*ap.S*ap.b*v0/(4*ap.m*Va0)*(ap.CYp*p0 + ap.CYr*r0) + ...

```

```

    rho*ap.S*v0/ap.m * (ap.CY0 + ap.CYbeta*beta0 + ap.CYda*da0 +
ap.CYdr*dr0) + ...
    rho*ap.S*ap.CYbeta/(2*ap.m)*sqrt(u0^2 + w0^2);
Yp = w0 + rho*Va0*ap.S*ap.b*ap.CYp / (4*ap.m);
Yr = -u0 + rho*Va0*ap.S*ap.b*ap.CYr / (4*ap.m);
Yda = rho * Va0^2 * ap.S * ap.CYda / (2*ap.m);
Ydr = rho * Va0^2 * ap.S * ap.CYdr / (2*ap.m);

Lv = rho*ap.S*ap.b^2*v0/(4*Va0) * (ap.Cpp * p0 + ap.Cpr * r0) + ...
    rho*ap.S*ap.b*v0*(ap.Cp0 + ap.Cpbeta*beta0 + ap.Cpda*da0 + ap.Cpdr*dr0)
+ ...
    rho*ap.S*ap.b*ap.Cpbeta/2 * sqrt(u0^2 + w0^2);
Lp = gamma(1)*q0 + rho*Va0*ap.S*ap.b^2/4 * ap.Cpp;
Lr = -gamma(2)*q0 + rho*Va0*ap.S*ap.b^2/4 * ap.Cpr;
Lda = rho*Va0^2*ap.S*ap.b/2 *ap.Cpda;
Ldr = rho*Va0^2*ap.S*ap.b/2 *ap.Cpdr;

Nv = rho*ap.S*ap.b^2*v0/(4*Va0) * (ap.Crp*p0 + ap.Crr*r0) + ...
    rho*ap.S*ap.b*v0*(ap.Cr0 + ap.Crbeta*beta0 + ap.Crda*da0 + ap.Crdr*dr0)
+ ...
    rho*ap.S*ap.b*ap.Crbeta/2 * sqrt(u0^2 + w0^2);
Np = gamma(7)*q0 + rho*Va0*ap.S*ap.b^2 * ap.Crp / 4;
Nr = -gamma(1)*q0 + rho*Va0*ap.S*ap.b^2 * ap.Crr / 4;
Nda = rho*Va0^2*ap.S*ap.b*ap.Crda / 2;
Ndr = rho*Va0^2*ap.S*ap.b*ap.Crdr / 2;

Alat = [Yv, Yp/(Va0*cos(beta0)), Yr/(Va0*cos(beta0)), ap.g*cos(theta0)/
(Va0*cos(beta0)), 0;
    Lv*Va0*cos(beta0), Lp, Lr, 0, 0;
    Nv*(Va0*cos(beta0)), Np, Nr, 0, 0;
    0, 1, cos(phi0)*tan(theta0), q0*cos(phi0)*tan(theta0) -
r0*sin(phi0)*tan(theta0), 0;
    0, 0, cos(phi0)*sec(theta0), p0*cos(phi0)*sec(theta0) -
r0*sin(phi0)*sec(theta0), 0];

Blat = [Yda/(Va0*cos(beta0)), Ydr/(Va0*cos(beta0));
    Lda, Ldr;
    Nda, Ndr;
    0, 0;
    0, 0];

end

Not enough input arguments.

Error in AircraftLinearModel (line 18)
ap = aircraft_parameters;

```

Published with MATLAB® R2024a