Problem 1

```
function [A,B] = linearizedmodelaircraft01(xeq,ueq,m,S,CLalpha,...
                                            CD0,oneoverpiARe)
%
%  Copyright (c) 2019 Mark L. Psiaki.  All rights reserved.
%
%  This function computes the linearized version of the
%  nonlinear dynamics point-mass model of an airplane
%  flying over a flat Earth in an atmosphere whose air
%  density decays exponentially with altitude.  This
%  is a linearization of the nonlinear model
%  that is contained in ffunctaircraft04.m.  Its
%  equilibrium state and control inputs, xeq and ueq,
%  should have been determined using the function
%  solvesteadystateaircraft01.m or a similar function.
%
%
%  Inputs:
%
%    xeq                  = [X;Y;Zeq;Veq;gammaeq;psieq],
%                         the 6-by-1 state vector of this system
%                         whose last four elements are steady-
%                         motion values.  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 -Zeq being the
%                         altitude displacement from a reference
%                         position.  The fourth element of x
%                         is 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).
%
%    ueq                  = [Teq;alphaeq;phieq], the 3-by-1
%                         equilibrium control input vector.
%                         Teq is the thrust in Newtons, alphaeq is
%                         the angle of attack in radians, and
%                         phieq is the roll angle in
%                         radians -- positive to the right.
%
%                         Note: the entries in xeq(3:6,1) and
%                         in ueq must be equilibrium values
%                         so that xdoteq(3:6,1) equals 0.
%                         Otherwise, a warning will be displayed
%                         by this function, and its outputs will
%                         be empty arrays.
%
%    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.
%
%  Outputs:
%
%    A                     The 6-by-6 state coefficient matrix
%                          in the linearized model about xsm(t) and
%                          ueq.
%
%    B                     The 6-by-3 control coefficient matrix
%                          in the linearized model about xsm(t) and
%                          ueq.
%
%                          The linearized dynamics model takes
%                          the form
%
%                            Deltaxdot(t) = A*Deltax(t) + B*Deltau(t)
%
%                          where Deltax(t) = x(t) - [XSM(t);YSM(t);...
%                          xeq(3:6,1)] = x(t) - xSM(t) and
%                          Deltau(t) = u(t) - ueq, with
%
%                             XSM(t) = X(t0) + XdotSM*(t - t0)
%                             YSM(t) = Y(t0) + YdotSM*(t - t0)
%
%                          and with XdotSM and YdotSM as calculated by
%                          solvesteadystateaircraft01.m or a
%                          similar function.
%
%


%
%  Test that xeq and ueq really contain equilibrium values.
%
   feq = ffunctaircraft04(xeq,ueq,m,S,CLalpha,CD0,oneoverpiARe);
   if norm(feq(3:6,1)) > 1.e-09
      disp(' ')
      disp('Failure in linearizedmodelaircraft01.m because the')
      disp(' inputs xeq and ueq do not correspond to an')
      disp(' equilibrium.')
      A = [];
      B = [];
      return
   end
%
%  Extract the thrust, angle-of-attack, and roll/bank-angle
%  inputs from u.
```

```
%
   Teq = ueq(1,1);
   alphaeq = ueq(2,1);
%  phieq = ueq(3,1);   % Not needed.  It is known to be zero at all
                       %   straight-and-level equilibria.
%
%  Extract the equilibrium altitude, airspeed, flight-path
%  angle, and heading angle.
%
   Zeq = xeq(3,1);
   Veq = xeq(4,1);
%  gammaeq = xeq(5,1); % Not needed.  It is known to be zero at all
                       %   straight-and-level equilibria.
   psieq = xeq(6,1);
%
%  Compute the lift and drag coefficients and their first
%  derivatives with respect to alpha.
%
   CL = CLalpha*alphaeq;
   CD = CD0+CL^2*oneoverpiARe;
   CLprime = CLalpha;
   CDprime = 2*oneoverpiARe*CL*CLalpha;
%
%  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 -Zeq + 649.7
%  is the aircraft altitude above sea level in meters.
%  649.7 m is the altitude of the coordinate system
%  origin above sea level.  The origin is at the
%  center of the runway of the airport in Blacksburg, VA.
%  Also compute the density's derivative with respect to Zeq.
%
   rho_sealevel = 1.225; % kg/m^3
   hscale = 10230.;      % meters
   rho = rho_sealevel*exp((Zeq - 649.7)/hscale); % kg/m^3
   rhoprime = rho/hscale;
%
%  Determine the dynamic pressure.
%
   Veqsq = Veq^2;
   qbar = 0.5*rho*Veqsq;
%
%  Set the flat-Earth gravitational acceleration at the
%  Blacksburg airport minus the effects of centrifugal
%  acceleration at the Blacksburg airport due to the
%  Earth's rotation vector.
%
   g = 9.79721; % meters/second^2
%
%  Initialize the A and B outputs.
%
   A = zeros(6,6);
   B = zeros(6,3);
%
%  Compute the non-zero elements of A.
%
   cos_psieq = cos(psieq);
   sin_psieq = sin(psieq);
   A(1,4) = cos_psieq;
```

```
   A(1,6) = -Veq*sin_psieq;
   A(2,4) = sin_psieq;
   A(2,6) = Veq*cos_psieq;
   A(3,5) = -Veq;
   oneoverm = 1/m;
   rho_S_over_m = rho*S*oneoverm;
   rhoprime_S_over_twom = rhoprime*S/(2*m);
   A(4,3) = -rhoprime_S_over_twom*CD*Veqsq;
   A(4,4) = -rho_S_over_m*Veq*CD;
   A(4,5) = -g;
   A(5,3) = rhoprime_S_over_twom*CL*Veq;
   A(5,4) = rho_S_over_m*CL;
%
%  Compute the non-zero elements of B.
%
   cos_alphaeq = cos(alphaeq);
   sin_alphaeq = sin(alphaeq);
   oneovermVeq = 1/(m*Veq);
   qbar_S = qbar*S;
   B(4,1) = cos_alphaeq/m;
   B(4,2) = -(Teq*sin_alphaeq+qbar_S*CDprime)/m;
   B(5,1) = sin_alphaeq*oneovermVeq;
   B(5,2) = (Teq*cos_alphaeq+qbar_S*CLprime)*oneovermVeq;
   B(6,3) = g/Veq;
```

Published with MATLAB® R2017a