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;
```

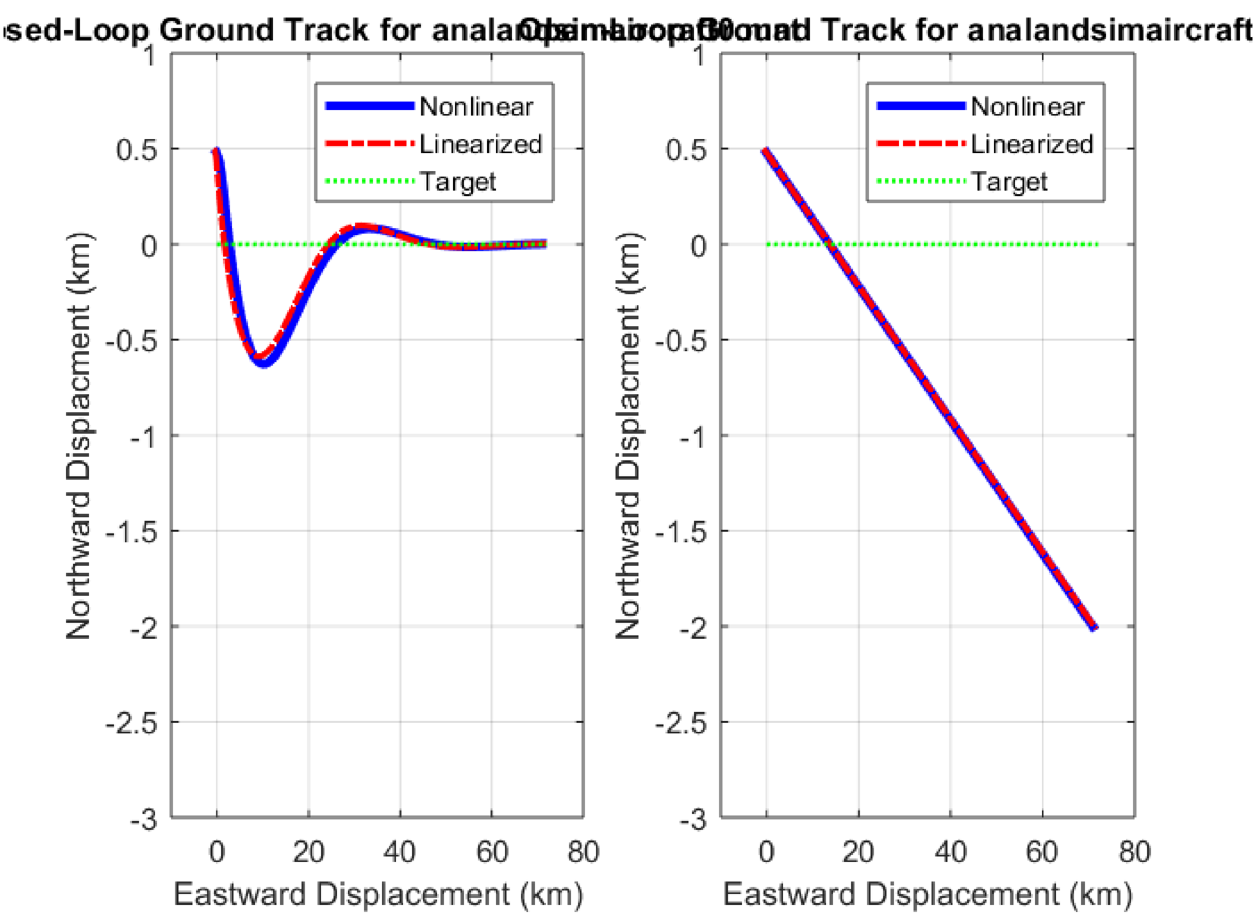
Problem 2

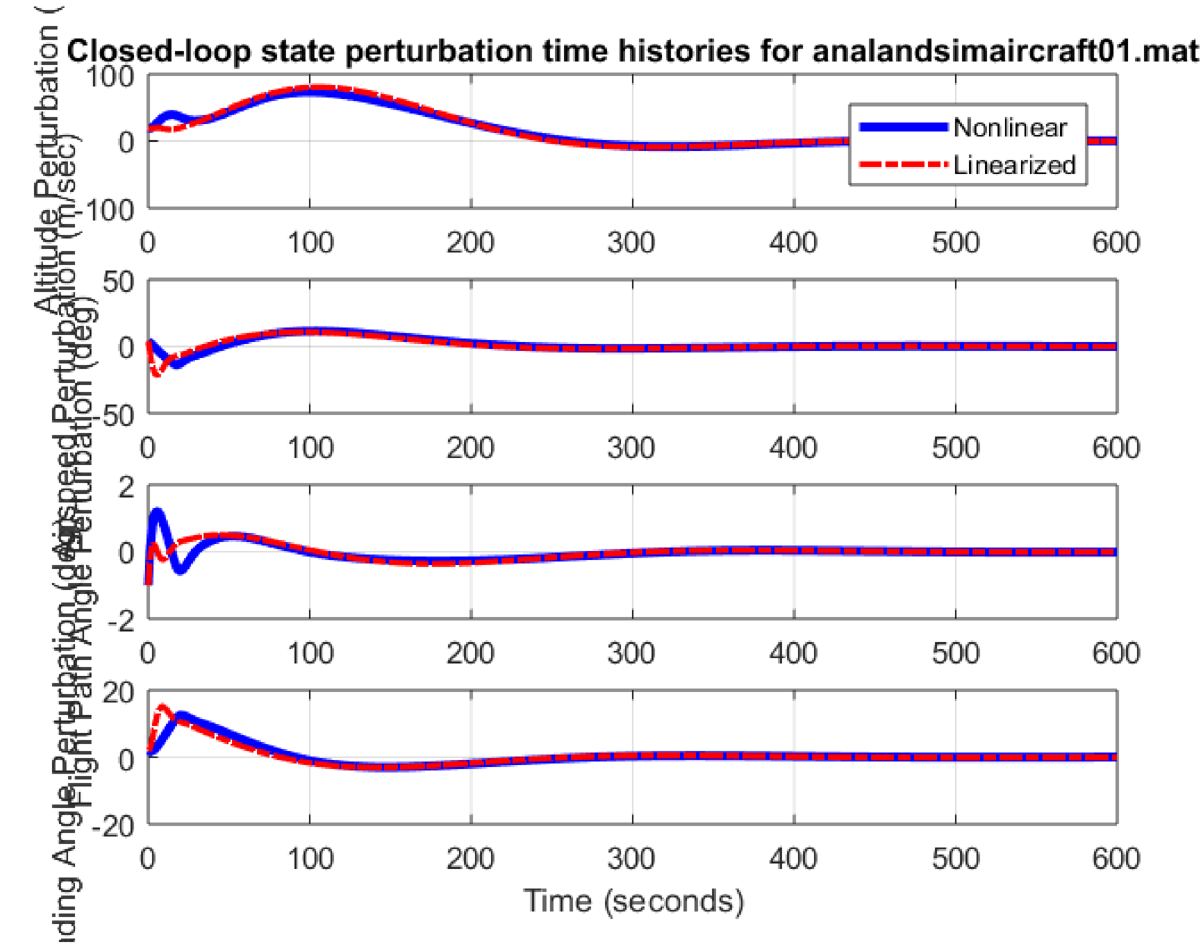
```
%script_analandsimaircraft01.m

   Controllabilitymat = ctrb(A,B); % C = [B, A*B, A^2 * B, ... , A^5 * B ]

   K = place(A,B,closedloopeigenvalues);

   ulinearfeedbackfunct = @(targdum,xargdum) ...
         ueq - K*(xargdum - xeq - [XdotSM;YdotSM;zeros(4,1)]*targdum); % U = Ueq+deltaU = Ueq-K*DeltaX

   Acl = A-B*K;
```

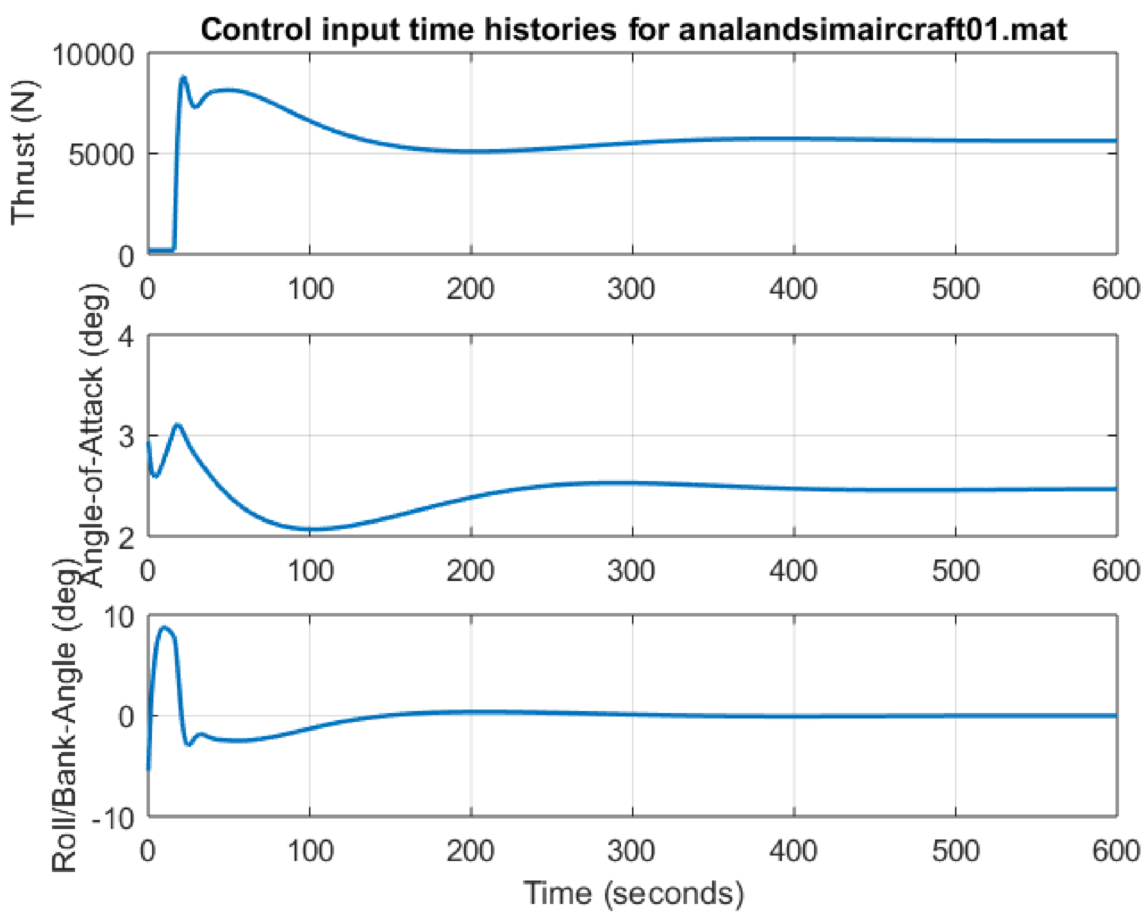
Output

```
A =

   1.0e+02 *

  Columns 1 through 3

                  0                   0                   0
                  0                   0                   0
                  0                   0                   0
                  0                   0  -0.000001048938339
                  0                   0   0.000000079431148
                  0                   0                   0

  Columns 4 through 6

   0.000000000000000                   0  -1.200000000000000
   0.010000000000000                   0   0.000000000000000
                  0  -1.200000000000000                   0
  -0.000178843986844  -0.097972100000000                   0
   0.000013543010718                   0                   0
                  0                   0                   0


B =

                  0                   0                   0
                  0                   0                   0
                  0                   0                   0
   0.000190227105528  -4.521936465223459                   0
   0.000000068313232   1.895722868536017                   0
                  0                   0   0.081643416666667


lambdavec =

 -0.008942199342177 + 0.118918387346877i
 -0.008942199342177 - 0.118918387346877i
 -0.000000000000000 + 0.000000000000000i
  0.000000000000000 + 0.000000000000000i
  0.000000000000000 + 0.000000000000000i
  0.000000000000000 + 0.000000000000000i
```

```
maxreallambda =

     0

This system may be neutrally stable because the maximum
 eigenvalue real part maximized over all of its eigenvalues
 appears to be zero to within machine precision.

Warning: For eigenvalue lambda = -1.8629e-18, the rank of (lambda*eye(n) - A) is
 4 but it should be smaller, it should be 2 in order for neutral
 stability to hold true, because this eigenvalue is repeated 4 times.
 Therefore, this system is unstable.

svsControllabilitymat =

   1.0e+02 *

   2.283055293661830
   0.185122207963509
   0.097972100000000
   0.048885111439677
   0.000816434166667
   0.000001396474218


The system is controllable.
```



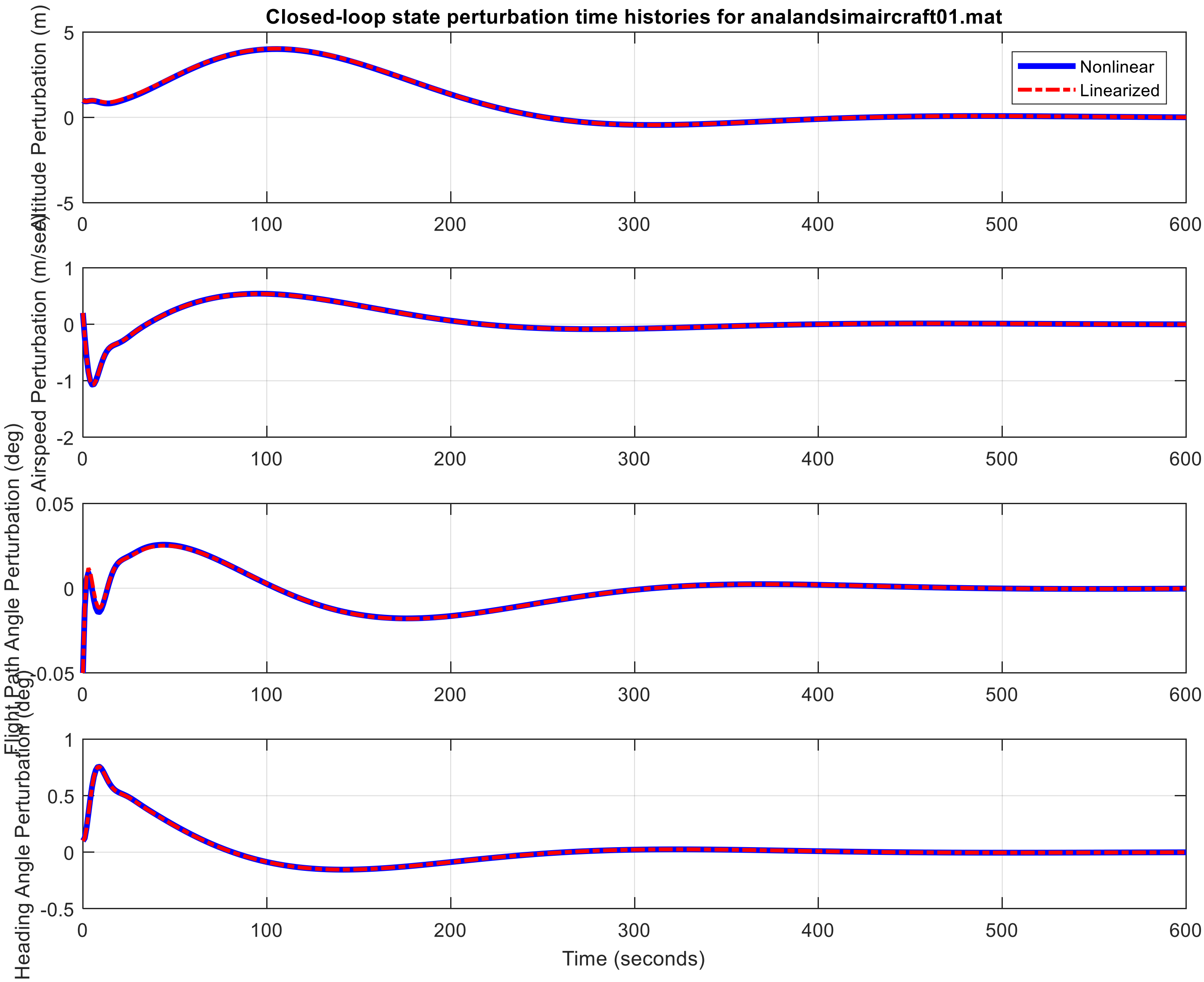

Closed-loop state perturbation time histories for analandsimaircraft01.mat
Nonlinear
Linearized
Time (seconds)


Control input time histories for analandsimaircraft01.mat
Thrust (N)
Angle-of-Attack (deg)
Roll/Bank-Angle (deg)
Time (seconds)

Q) How well does the closed-loop system track the target steady-motion trajectory after initial transients have died out?

A) The error in trajectory is within 2mts and 4 mts for X and Y respectively at the final simulation time. The oscillation are not completely settled at this time.

Case with 1/20 Factor of initial petrubation:



Q) In which case does the nonlinear response more closely match the linear response? Is this what you would expect?

A) The smaller X0 matches the nonlinear response better. This behavior is expected because; the new X0 is closer to the equilibrium and this improvement is to be anticipated. The A,B are linearized around the Xeq, and closer the X0 is to the Xeq, the A,B matrices represent the original nonlinear system better. This would mean that the controller works better, at is derived with A,B at equilibrium.

Problem 3

```
function [A,B] = linearizedmodelgravgradsc01(norbit,Ib11,Ib22,Ib33)
%
%  Copyright (c) 2019 Mark L. Psiaki.  All rights reserved.
%
%  This function computes the linearized version of the
%  nonlinear dynamics model of the rigid-body attitude
%  dynamics of a spacecraft that is subject to the gravity-
%  gradient torques caused by a spherical central attracting
%  body.  The spacecraft orbits this body in a circular
%  orbit with mean motion norbit radians/sec and orbital period
%  Torbit = 2*pi/norbit.  The principal moments of inertia
%  are Ib11, Ib22, and Ib33, with Ib11 being the moment-of-
%  inertia about the principal axis that is nominally
%  aligned with the velocity vector (i.e., nominally the
%  roll axis), Ib22 being the moment-of-inertia
%  about the principal axis that nominally points out
%  the "right wing" (i.e., nominally the pitch axis),
%  and Ib33 being the moment-of-inertia about the
%  principal axis that nominally points towards nadir/
%  the center of the Earth (i.e., nominally the yaw axis).
%  This is a linearization of the nonlinear model that is
%  contained in ffunctgravgradsc03.m.
%
%  The state vector of the linearized dynamic model has only 6
%  elements despite the corresponding nonlinear model having a
%  7-element state vector.  This model's 6-element state vector is:
%
%     Deltaxtil = [Deltaq1;Deltaq2;Deltaq3;Deltaomegab1;...
%
%                  Deltaomegab2;Deltaomegab3]
%
%  where Deltaq1, Deltaq2, and Deltaq3 are the perturbations
%  of the first three elements of the actual quaternion from
%  the nonlinear system's equilibrium quaternion value
%  qeq = [0;0;0;1] and where Deltaomegab1, Deltaomegab2, and
%  Deltaomegab3 are the perturbations of the components of the
%  actual inertial angular rate along body axes (which are
%  principal axes) from the equilibrium value omegabeq = ...
%  [0;-norbit;0].  Thus, xeq = [0;0;0;1;0;-norbit;0] is the
%  equilibrium state from which perturbations are measured.
%
%  Recall that q = x(1:4,1) in the original
%  nonlinear system state vector is the unit-normalized
%  attitude quaternion for the rotation from local-level
%  orbit-following coordinates to spacecraft body-axes
%  coordinates and that omegab = x(5:7,1) in the original
%  nonlinear system state vector is the spin-rate vector
%  of the body-axis coordinate system relative to inertial
%  coordinates and resolved into components that are defined
%  along the body-fixed axes.
```

```
%
%  Note that the control input is the net external torque
%  in addition to the gravity-gradient torque.  It is
%  defined along spacecraft body-fixed axes.  Thus, u = Tb.  Note
%  that the equilibrium value is ueq = Tbeq = [0;0;0].
%
%
%  Inputs:
%
%    norbit                 The mean orbital motion in radians/sec.
%                           Note that the orbital period is Torbit
 = ...
%                           2*pi/norbit.
%
%    Ib11                   The moment of inertia about the principal
%                           axis that is nominally aligned with
%                           the roll axis (the velocity axis),
%                           in kg-m^2.
%
%    Ib22                   The moment of inertia about the principal
%                           axis that is nominally aligned with
%                           the pitch axis (out the "right wing"),
%                           in kg-m^2.
%
%    Ib33                   The moment of inertia about the principal
%                           axis that is nominally aligned with
%                           the yaw axis (the nadir-pointing axis),
%                           in kg-m^2.
%
%  Outputs:
%
%    A                      The 6-by-6 state coefficient matrix
%                           of the linearized model about xeq and
%                           ueq.
%
%    B                      The 6-by-3 control coefficient matrix
%                           of the linearized model about xeq and
%                           ueq.
%
%                           The linearized dynamics model takes
%                           the form
%
%                             Deltaxtildot(t) = A*Deltaxtil(t) +
 B*Deltau(t)
%
%                           where Deltaxtil = x([1:3,5:7],1) - ...
%                           xeq([1:3,5:7],1) and Deltau = u - ueq,
%                           with xeq and ueq defined above.
%                           Thus, Deltaxtil has had the 4th element
%                           of Deltax = x - xeq deleted from it
%                           because this fourth element, Deltaq4
%                           is known to equal 0 to first-order
%                           in the linearized perturbations due to
%                           the quaternion unit normalization
```

```
%                            constraint.
%

%
%  Initialize the output arrays.
%
   A = zeros(6,6);
   B = zeros(6,3);
%
%  Assign the individual non-zero elements of these two arrays.
%
   A(1,3) = norbit;
   A(1,4) = 0.5;
   A(2,5) = 0.5;
   A(3,1) = -norbit;
   A(3,6) = 0.5;
   norbitsq = norbit^2;
   sixnorbitsq = 6*norbitsq;
   Iratio_row4 = (Ib33 - Ib22)/Ib11;
   A(4,1) = sixnorbitsq*Iratio_row4;
   A(4,6) = norbit*Iratio_row4;
   Iratio_row5 = (Ib33 - Ib11)/Ib22;
   A(5,2) = sixnorbitsq*Iratio_row5;
   Iratio_row6 = (Ib22 - Ib11)/Ib33;
   A(6,4) = norbit*Iratio_row6;
   B(4,1) = 1/Ib11;
   B(5,2) = 1/Ib22;
   B(6,3) = 1/Ib33;
```

Published with MATLAB® R2017a

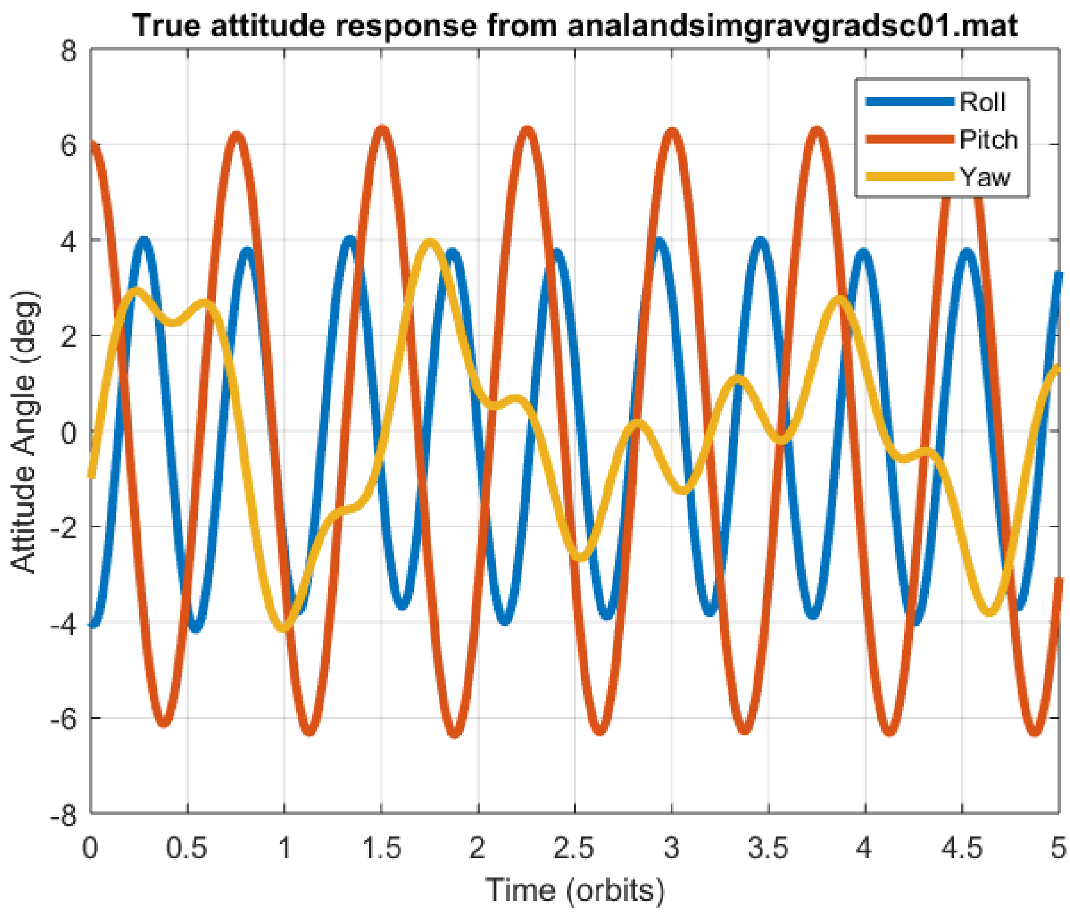
Problem 4

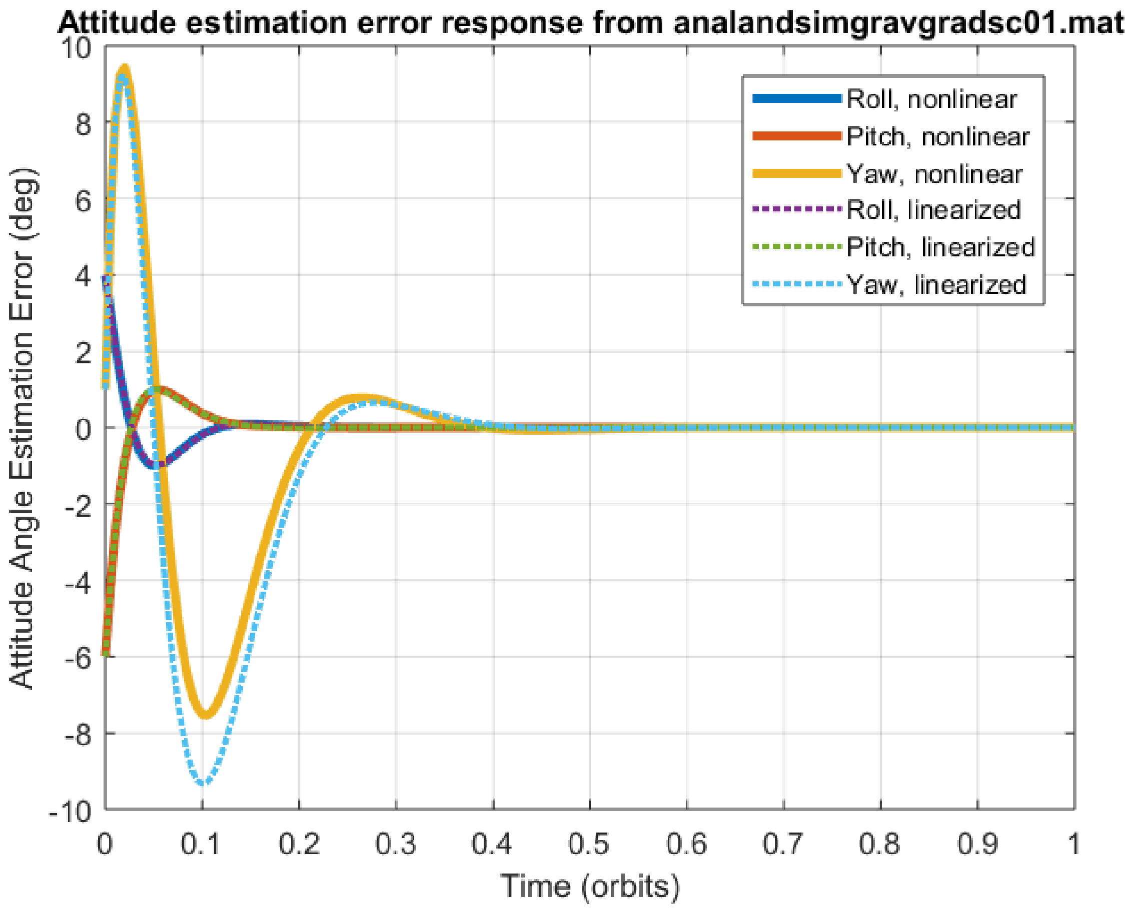
```
%script_analandsimgravgradsc01.m
%
   Observabilitymat = obsv(A,C); % O = [C; C*A; C * A^2; ... ; C * A^5]
%
   L = place(A',C',observereigenvalues)';
%
   Aclobs = A-L*C;
```

Output

```
A =

  Columns 1 through 3

                  0                   0   0.001047197551197
                  0                   0                   0
  -0.001047197551197                  0                   0
  -0.000005805649648                  0                   0
                  0  -0.000003947841760                   0
                  0                   0                   0

  Columns 4 through 6

   0.500000000000000                  0                   0
                  0   0.500000000000000                   0
                  0                   0   0.500000000000000
                  0                   0  -0.000923997839291
                  0                   0                   0
   0.000628318530718                  0                   0


B =

                  0                   0                   0
                  0                   0                   0
                  0                   0                   0
   0.011764705882353                  0                   0
                  0   0.010000000000000                   0
                  0                   0   0.040000000000000


lambdavec =

 -0.000000000000000 + 0.000804738358774i
 -0.000000000000000 - 0.000804738358774i
  0.000000000000000 + 0.001404962946208i
  0.000000000000000 - 0.001404962946208i
  0.000000000000000 + 0.001983030174700i
  0.000000000000000 - 0.001983030174700i


maxreallambda =

     2.710505431213761e-19
```

```
This system may be neutrally stable because the maximum
 eigenvalue real part maximized over all of its eigenvalues
 appears to be zero to within machine precision.

svsObservabilitymat =

   2.000000000015996
   2.000000000003896
   1.000002390641478
   1.000000007589253
   1.000000000001948
   0.000001315944118


The system is observable.
```

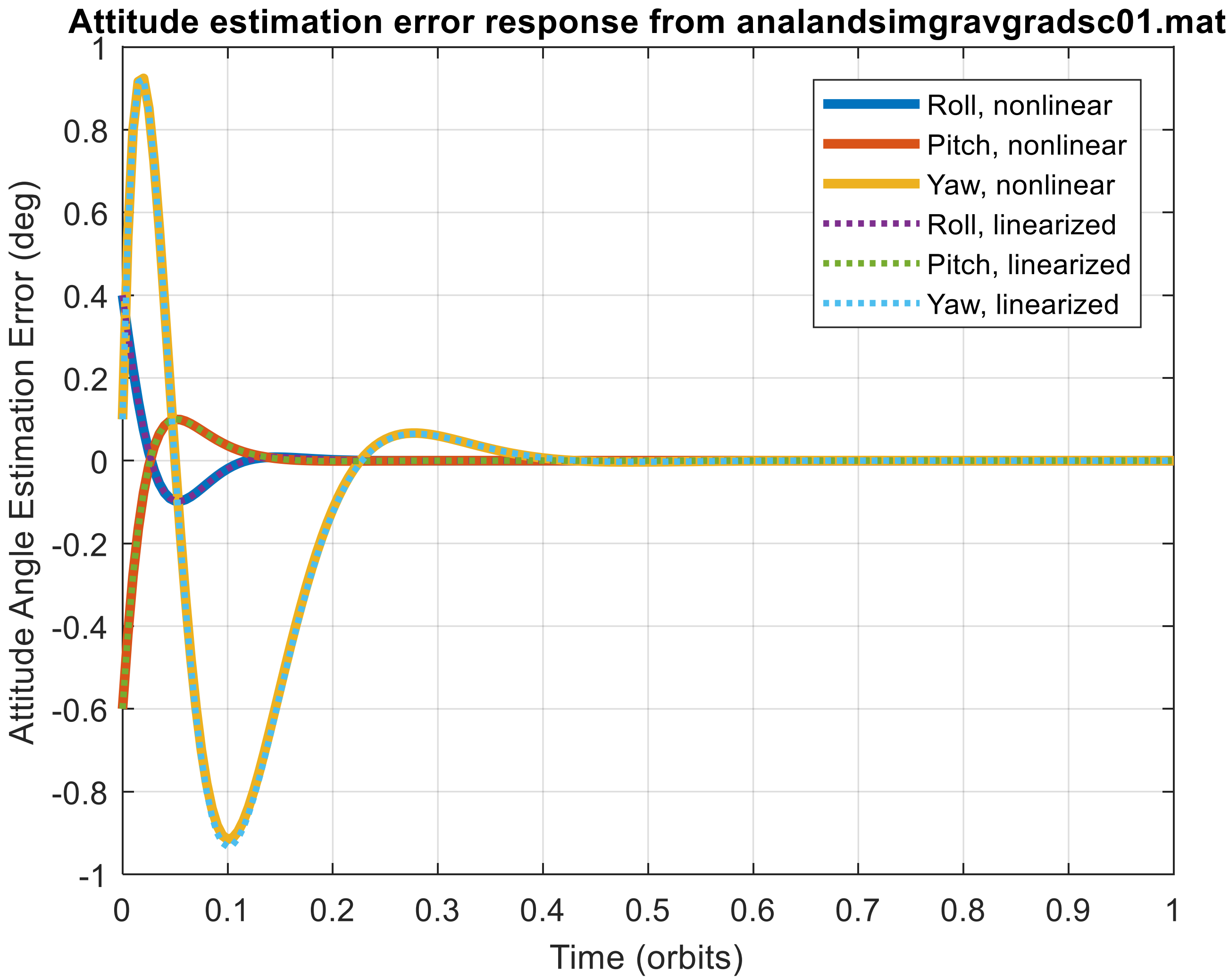




Q) How well do the observer errors converge to zero?

A) The final error is of the order of 1e-6. The convergence is good and practically zero.

Case with 1/10 Factor of initial petrubation:



Q) In which case does the nonlinear observer error response more closely match the linear response? Is this what you would expect?

A) The smaller X0 matches the nonlinear response better. This behavior is expected because; the new X0 is closer to the equilibrium and this improvement is to be anticipated. The A is linearized around the Xeq, and closer the X0 is to the Xeq, the A matrix represents the original nonlinear system better. This would mean that the state estimator works better, as it is derived with A at equilibrium.