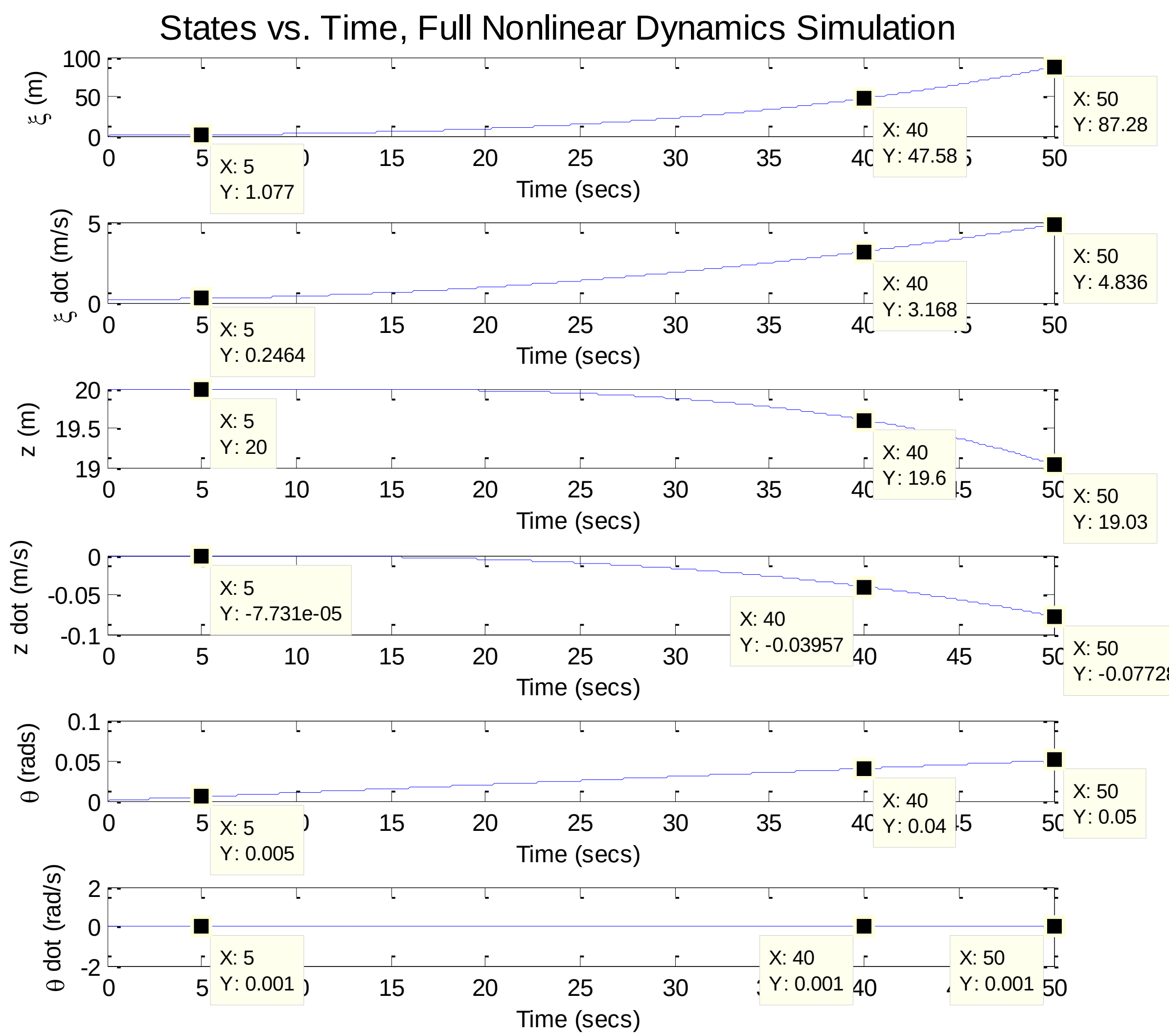
HW 8 Problem 2 Solution Sketch – Simple Skycrane problem for Project Part 1

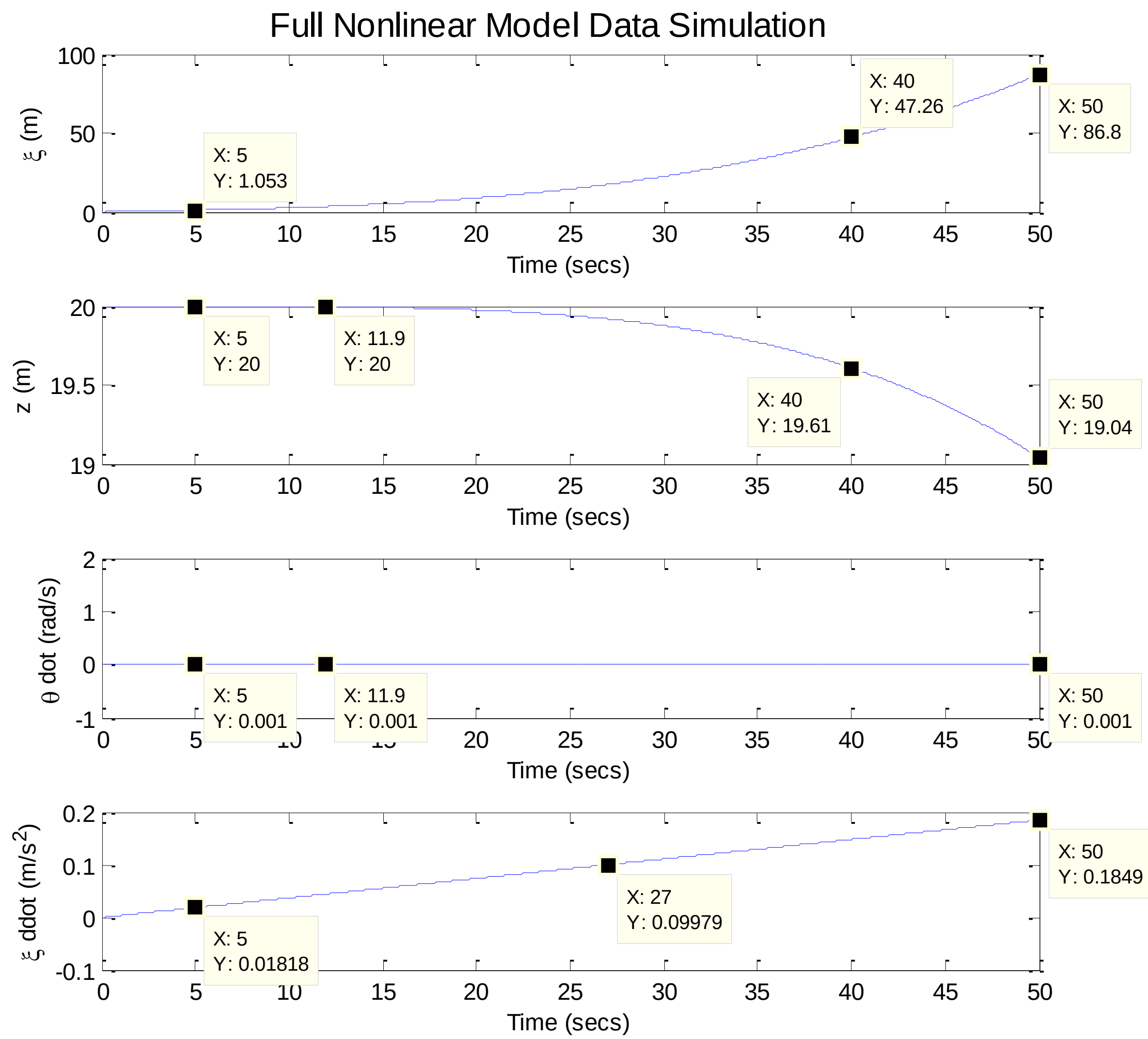
In the following plots for Project Part 1, the initial state conditions stated in the problem are perturbed by the vector

```
perturb_x0 = [0; 0.2; 0; 0; 0; 0.001]
```

with no control inputs, no process noise inputs, and with no measurement noise outputs simulated.

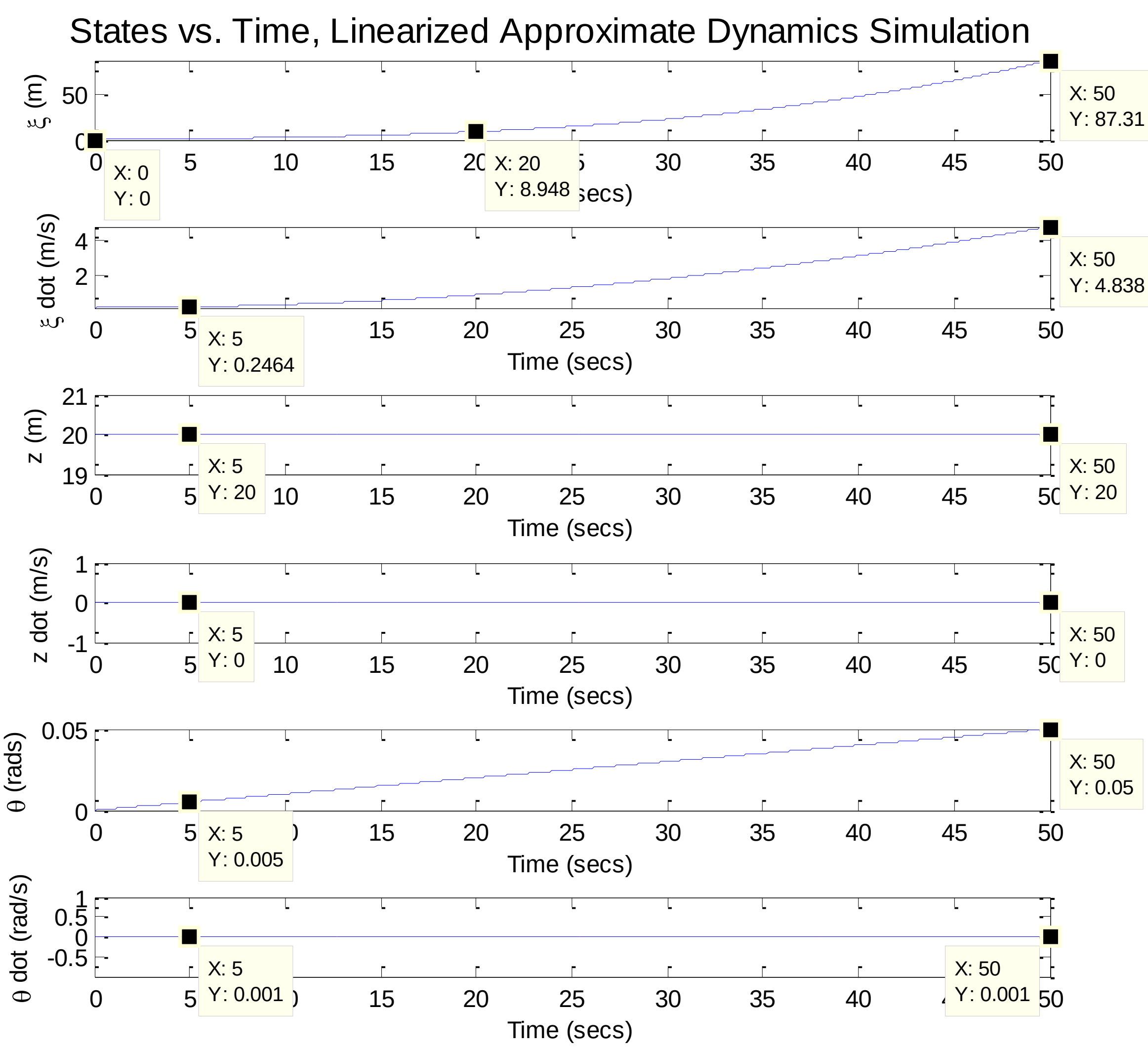
Plot of states vs. time for full nonlinear dynamics (integrated via ode45) for 500 time steps, followed by plot of resulting measurements according to full nonlinear measurement model:

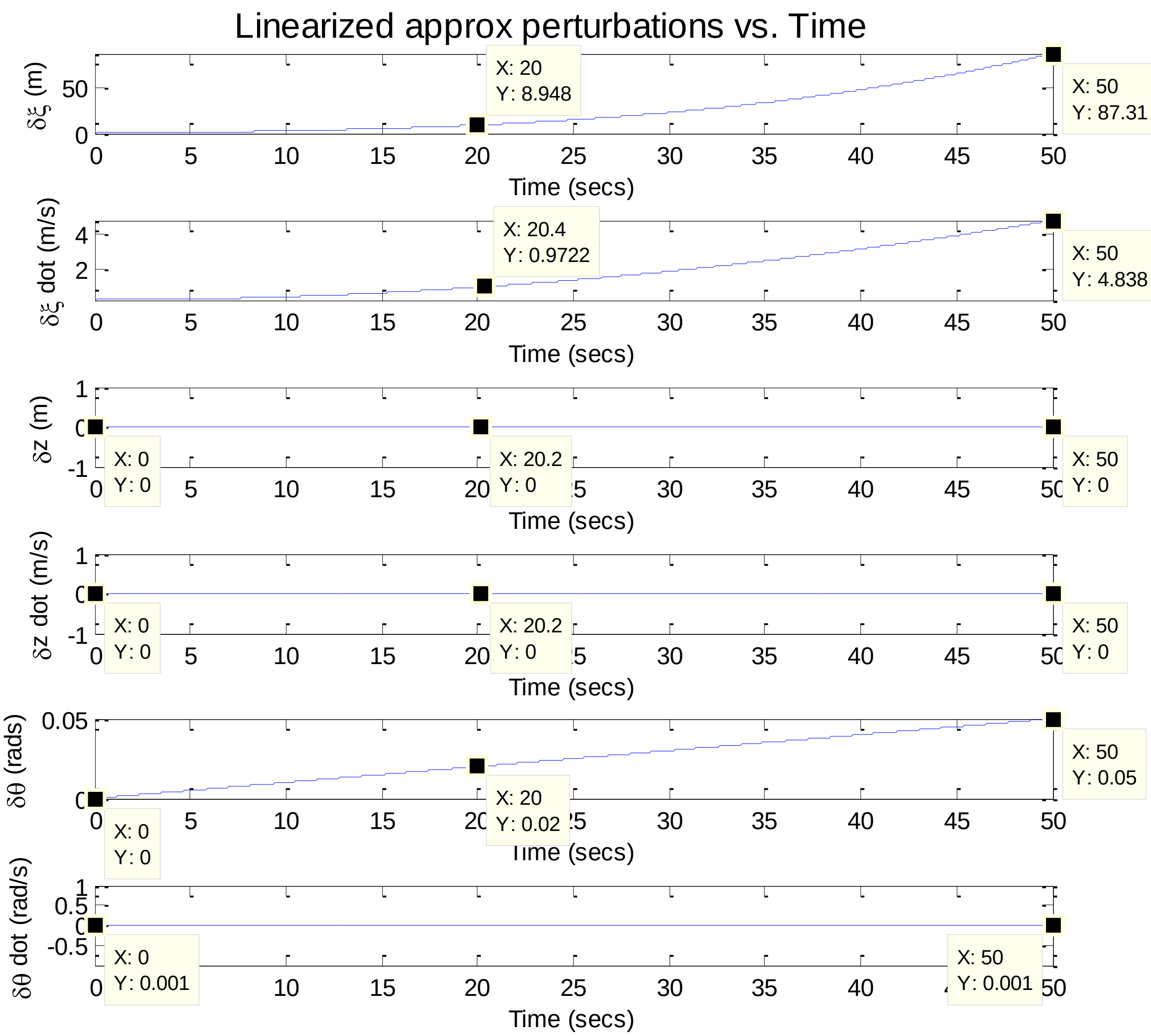




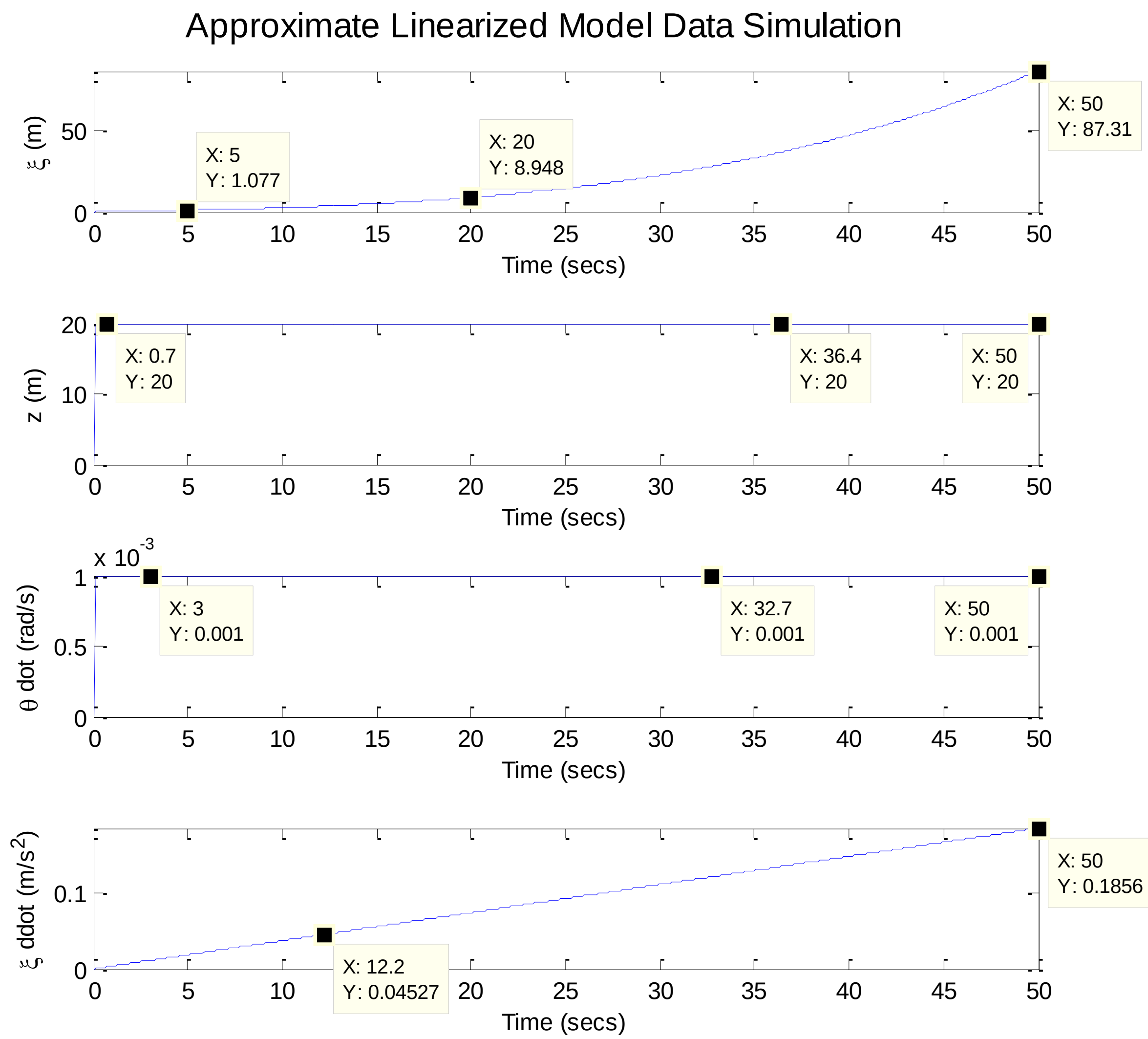
Full Nonlinear Model Data Simulation
ξ (m)
Time (secs)
X: 5
Y: 1.053
X: 40
Y: 47.26
X: 50
Y: 86.8
z (m)
X: 5
Y: 20
X: 11.9
Y: 20
X: 40
Y: 19.61
X: 50
Y: 19.04
θ dot (rad/s)
X: 5
Y: 0.001
X: 11.9
Y: 0.001
X: 50
Y: 0.001
ξ ddot (m/s2)
X: 5
Y: 0.01818
X: 27
Y: 0.09979
X: 50
Y: 0.1849

Next is a plot of total states vs. time using linearized dynamics model (linearized about nominal noise free state equilibrium at constant hover about a fixed altitude and horizontal position, with zero pitch angle and zero pitch rate), followed by plot of corresponding state perturbations (evolving according to linearized dynamics near nominal trajectory) and plot of measurements according to linearized measurement model (which accounts for nominal measurements + linearized perturbations in nominal measurements). Note: the linearized model here diverges from the full nonlinear model in the z states and z measurements, because the aircraft starts diverging from equilibrium and the dynamics linearized only about the equilibrium do not predict the emerging interaction between zdot and other states as time progresses (the only way to fix this would be to relinearize at each time about an estimated or known state, as in the EKF):





Linearized approx perturbations vs. Time
δξ (m)
X: 20
Y: 8.948
X: 50
Y: 87.31
Time (secs)
δξ dot (m/s)
X: 20.4
Y: 0.9722
X: 50
Y: 4.838
Time (secs)
δz (m)
X: 0
Y: 0
X: 20.2
Y: 0
X: 50
Y: 0
Time (secs)
δz dot (m/s)
X: 0
Y: 0
X: 20.2
Y: 0
X: 50
Y: 0
Time (secs)
δθ (rads)
X: 0
Y: 0
X: 20
Y: 0.02
X: 50
Y: 0.05
Time (secs)
δθ dot (rad/s)
X: 0
Y: 0.001
X: 50
Y: 0.001
Time (secs)



Controllability, Observability, and Stability Analysis: since the CT Jacobians about equilibrium are all LTI, we can perform these analyses.

*Eigenvalues: [0, -3.26315789473684e-06, 0, -3.26315789473684e-06, 0, 0] → system has repeated eigenvalues on Imaginary axis → system is not BIBO stable

*Controllability matrix: rank = 6 → system is controllable near equilibrium

*Observability matrix: rank = 6 → system is observable

HINT FOR PROJECT ASSIGNMENT PART 2:

In order to run the truth model tests for the linearized KF and EKF, you will need to assign a stabilizing feedback control law to generate the control signal u(k) in each instance of a truth model simulation **– the resulting signal for a particular simulation instance should be stored (along with the simulated noisy sensor data) and provided as inputs to the linearized KF/EKF that you are assessing in the NEES/NIS tests.**

The following LQR state-feedback control law should be used to generate the total control signal in the nonlinear truth model simulation for the Skycrane problem:

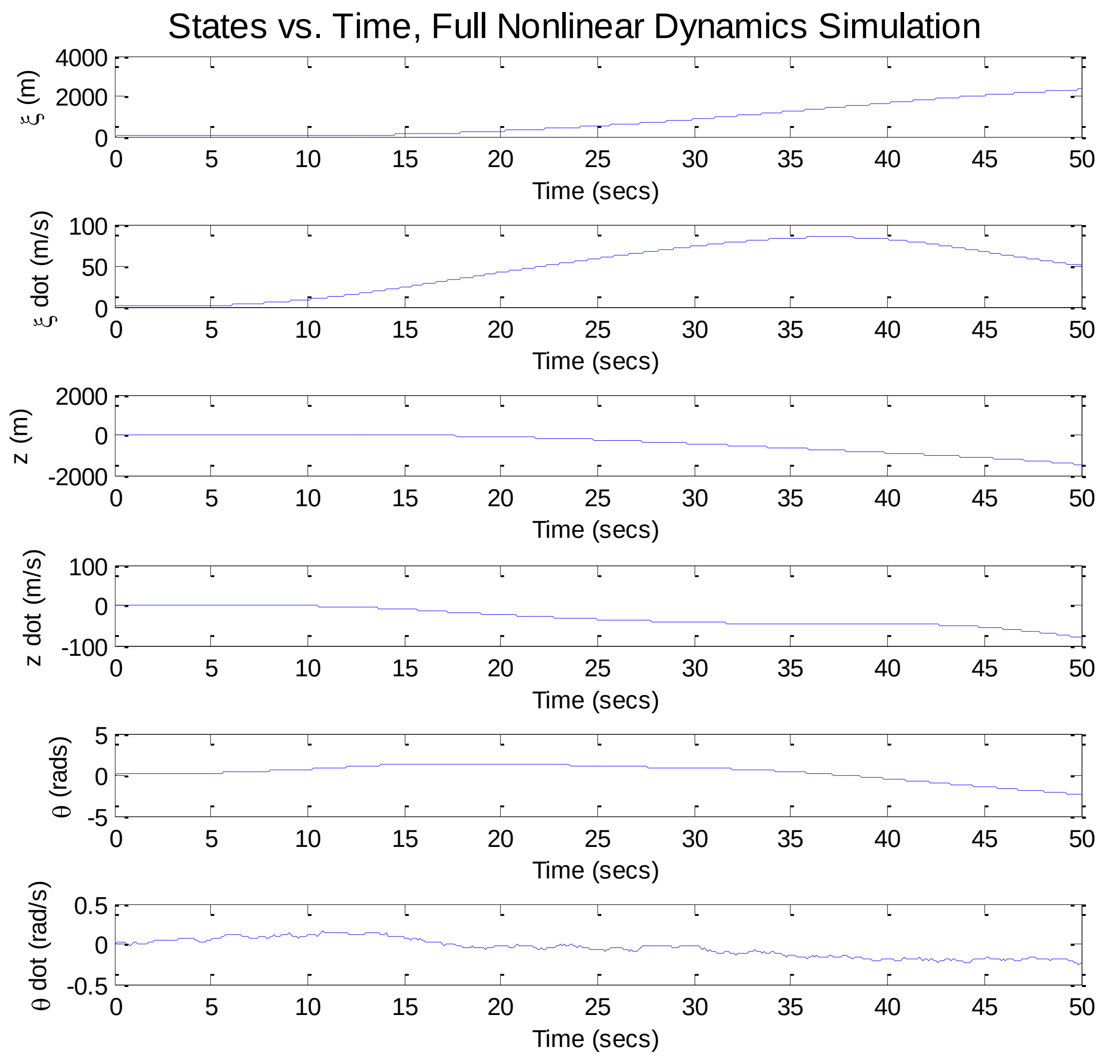
$u(k) = u_{NOMINAL}(k) – K_{LIN}*[x(k) – x_{NOMINAL}(k)]$

where K_{LIN} is a 2 x 6 linear state feedback gain matrix (specified in the final project data file), and $u_{NOMINAL}$ is the 2x1 nominal control thrust that you used for linearization about equilibrium, and $x_{NOMINAL}(k)$ is the reference nominal state at time k. Note that this control law is based on knowledge of the actual state x(k) at each time step k – this is an idealistic assumption since x(k) would be provided by a KF in a realistic application (i.e. this is exactly what the linearized KF or EKF would be used for!), but for the purposes of this project we will assume that the aircraft controller "already knows" this (since control design is out of scope for this project and we can use the separation principle from linear control theory to design the estimator and control blocks separately for this vehicle).

Therefore, in your nonlinear ground truth state simulation code for simulating x(k+1) at each time step k, you should put a line to generate the control signal u(k) at time step k and store the resulting values of u(k) before calling ode45 to update x(k+1) based on inputs x(k), w(k), and u(k) (Note that values of u(k) will be random because of process noise effects and random initial conditions). For the linearized KF, there will then be a non-zero $\delta u(k)$ term for each time step k, where $\delta u(k) = u(k) – u_{NOMINAL}(k)$. For the EKF, you only need to worry about the total control input u(k). When running either the linearized KF or EKF, you only need to use the pre-recorded values of $\delta u(k)$ or u(k) to feed into the KF (i.e. you do not need to reconstruct the control law in the linearized KF or EKF loops – just read off the values from the $\delta u(k)$ or u(k) array that you recorded in the nonlinear truth model simulation).

For the provided data log yhist, the corresponding u(k) control input time history for each time step k is provided in uhist.

As a sanity check, here is the kind of result you should see for the Skycrane if NOT using any feedback control law at all AND if there is process noise w(k) present as specified in the final project data file (i.e. only assuming $u(k) = u_{NOMINAL}(k)$ or $\delta u(k) = 0$ for all time k) – note that the aircraft wanders away from equilibrium and basically crashes into the ground:



States vs. Time, Full Nonlinear Dynamics Simulation
ξ (m)
ξ dot (m/s)
z (m)
z dot (m/s)
θ (rads)
θ dot (rad/s)
Time (secs)

And here is the kind of result you should get if using the feedback control law specified above -- the aircraft is able to keep its station at equilibrium and reject process noise disturbances that would otherwise make it go unstable:

BASE Closed loop LQR Nonlinear State Response