Project Phase 1

AER 1202 – Prof. Liu

Sam Weinberg (1005347634)

Navneet Gopinath (1005077319)

1) Altitude Control System

An altitude control system was designed for a Boeing 747 flying at an altitude of 40,000 ft and a speed of Mach 0.8. The longitudinal state space model was formulated using the derivation outlined in Etkins (pg. 112).

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

For the longitudinal system, we use the normal state vector with the addition of the z_E to use as feed back for the altitude command.

$$x = \begin{bmatrix} \Delta u \\ w \\ q \\ \Delta\theta \\ \Delta z_E \end{bmatrix}$$

The control inputs for the longitudinal system are the elevator and throttle.

$$u = \begin{bmatrix} \delta_e \\ \delta_p \end{bmatrix}$$

The system matrix was calculated using the parameters (derivative coefficients, weight, etc.) provided for the Boeing 747 aircraft. The system matrix was calculated to be

$$A = \begin{bmatrix} -0.0069 & 0.0139 & 0 & -9.81 & 0 \\ -0.0905 & -0.3149 & 235.89 & 0 & 0 \\ 0.0004 & -0.0034 & -0.4281 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -235.89 & 0 \end{bmatrix}$$

Similarly, the control matrix was calculated using the control derivatives from Etkins (pg.)

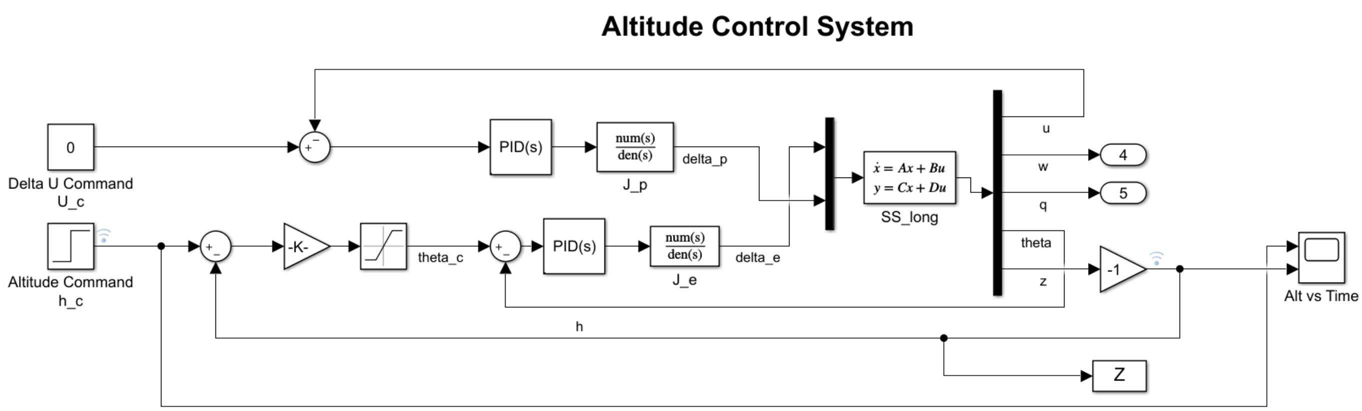
$$B = \begin{bmatrix} -0.000057 & 2.943 \\ -5.4714 & 0 \\ -1.159 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The state space model is completed with the C and D matrices.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \; and \; D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The state space model was created in Matlab using the "ss" function. The transfer functions for each input/output combination can be extracted from this model to develop the control law system.

The block diagram Simulink model for the altitude control is shown below. The diagram can be enlarged in the attached Simulink file.

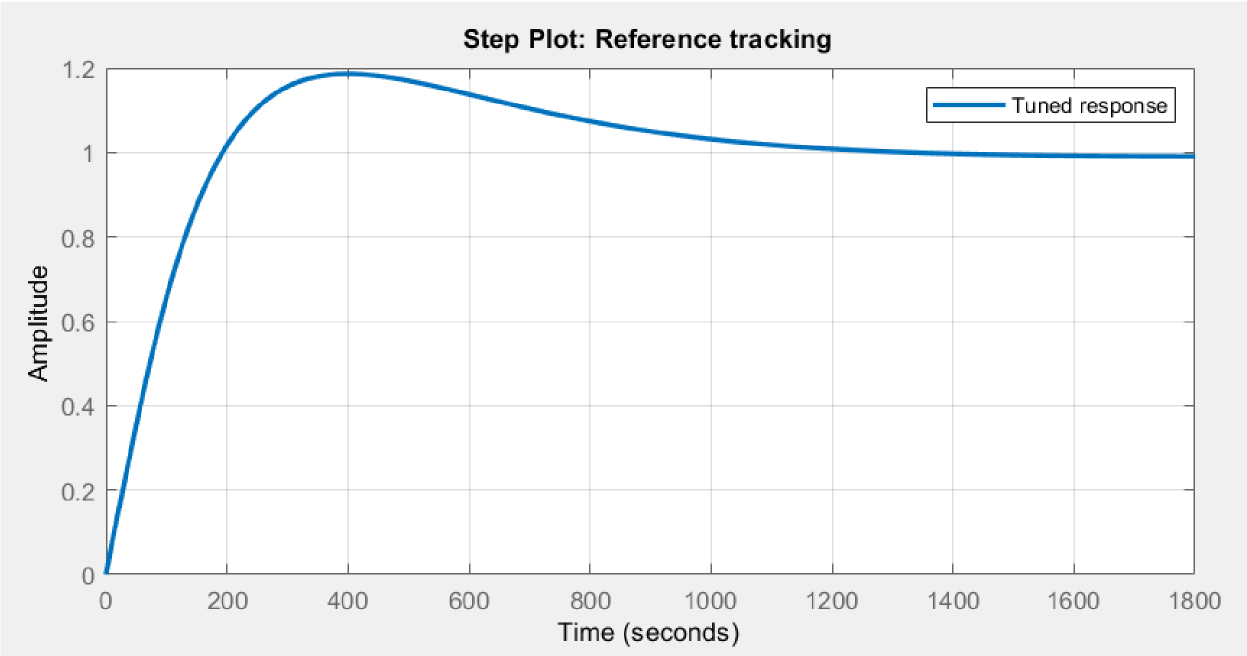


The altitude command is a step input of Δh in metres, where the reference altitude is 40,000 ft as described in the question. As a result, the step input final value is 1524 m (5000 ft), which represents a climb from 40000 ft to 45000 ft as required. The throttle is used to maintain a velocity u of 235.9 m/s (Mach 0.8) during the ascent. Therefore, the Δu command is held at a constant zero to maintain the velocity, however this can be altered if another velocity is desired.

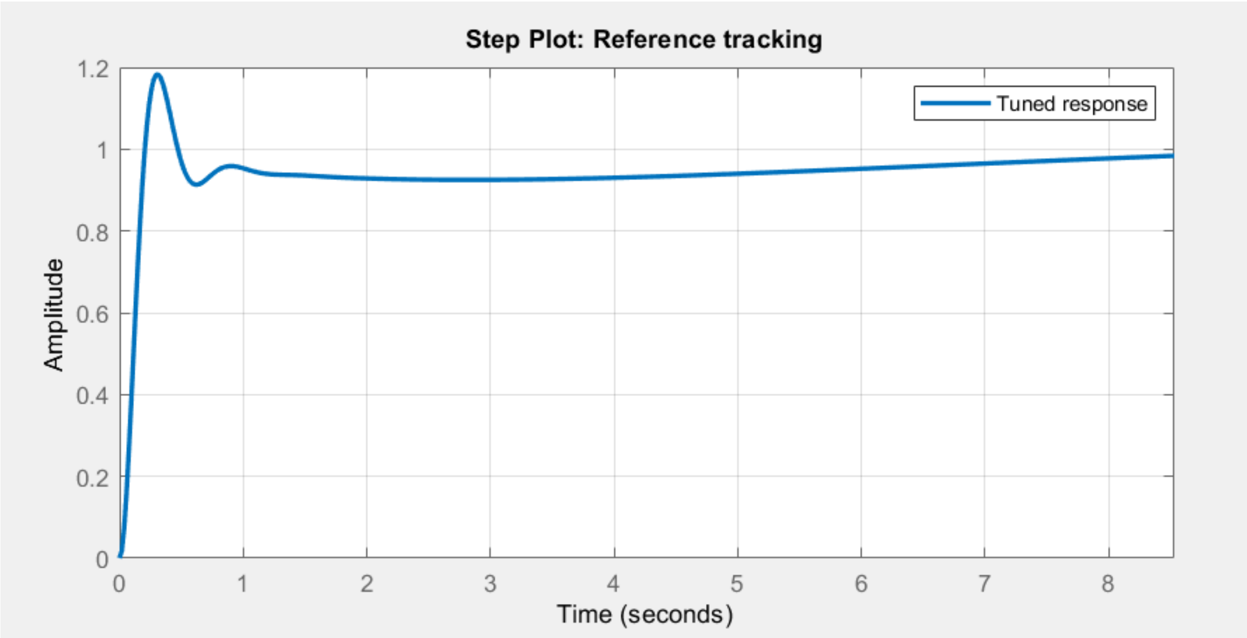
The altitude error is converted to a desired pitch angle through a gain and a limiter. The gain that produced reasonable results was reference from Etkins (pg. 280) to be 0.0002 (larger values were too responsive whereas smaller values produced long settling times). The limiter is used to maintain a desired pitch angle that is realistic and produces a smooth ascent. The maximum pitch angle was chosen to be 5 degrees, however this value can be altered as desired. The pitch angle error is propagated through a PID control law system. Additionally, the elevator servo actuator block is used to simulate a realistic system with time constant 0.1. The overall control law is summarized as

$$C(s) = \left(K_p + K_i \frac{1}{s} + K_d s\right)\left(\frac{1}{\tau_e s + 1}\right)$$

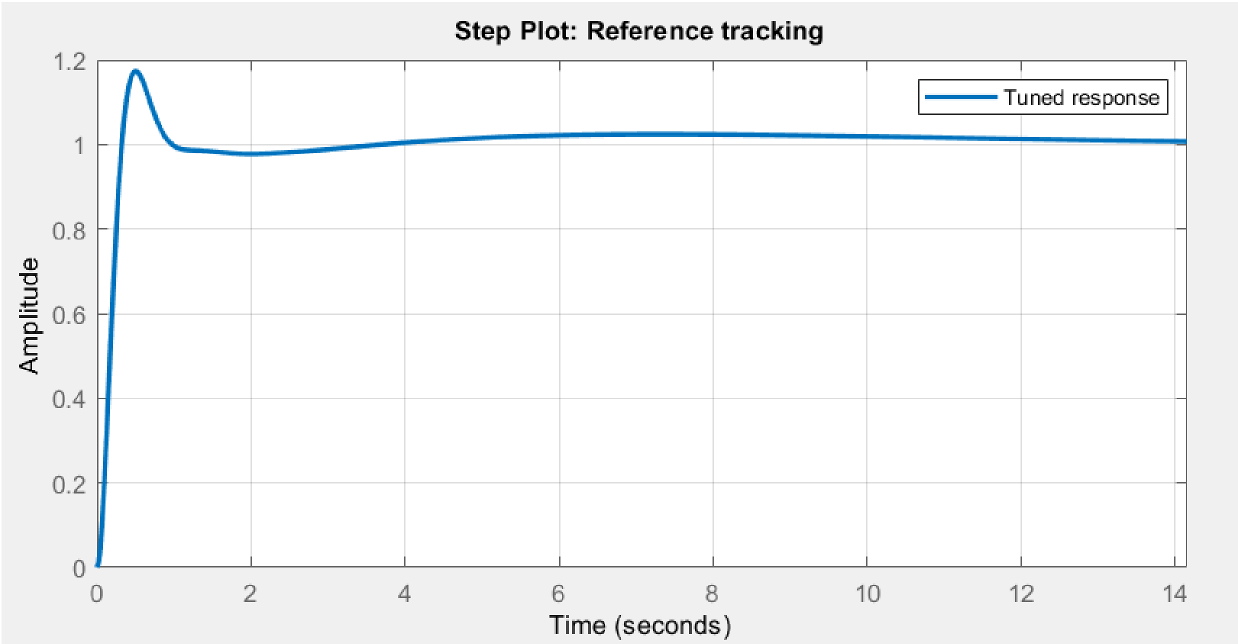
The PID controller constants were found using the PID tuner in Simulink. An example of this tuning process is demonstrated below. The initial PID tuning is show below.



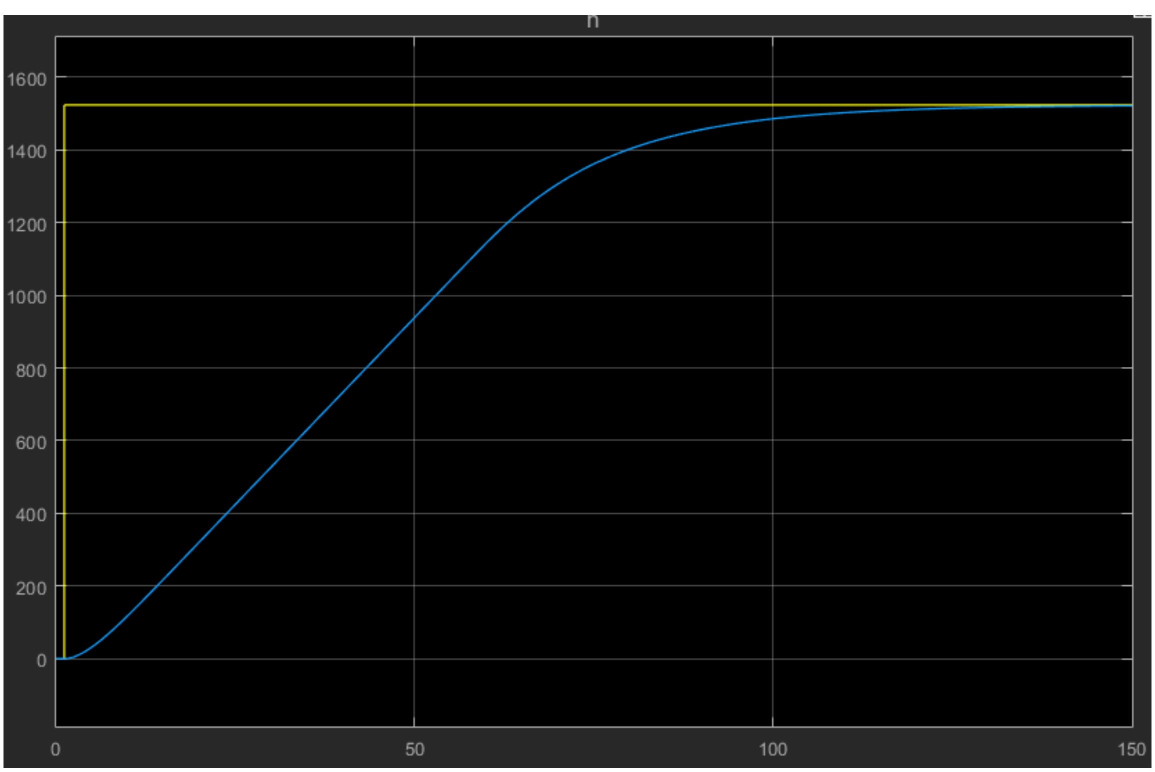
While the plot produces reasonable results, the desired response needs to be much quicker and ideally have less overshoot. We tune the response time to be faster and obtain the following plot.



Now we've reduced the response time and maximum overshoot remains the same, but steay state error takes a while to die out. The signal takes over 8 seconds to reach our desired value. Further tweaking of the response time and transient behaviour coupled with observations of the altitude response yields the final PID tuning.



The resultant elevator input along with the throttle input, which was modelled in a similar manner (PID control and time delay), were input into the longitudinal state space model to obtain the state vector. Both the pitch angle and height difference were fed back to the desired commands to complete the closed loop system. The step response to an altitude command of 1524 m (5000 ft) is shown below. The percent overshoot and steady state error are well within their specifications. The ascent takes approximately 130 seconds.



2) Heading Control System

The heading control system was designed to perform a steady turn of 90 degrees. A lateral state space model was created using Etkins again. The state vector includes the yaw to provide feedback to the heading command.

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

$$x = \begin{bmatrix} \mathrm{v} \\ p \\ r \\ \phi \\ \psi \end{bmatrix}$$

The control inputs for the lateral system include the ailerons and rudder.

$$u = \begin{bmatrix} \delta_{\mathrm{a}} \\ \delta_r \end{bmatrix}$$

The system matrix was calculated using the lateral dimensional derivatives and found to be

$$A = \begin{bmatrix} -0.0558 & 0 & -235.9 & 9.81 & 0 \\ -0.0127 & -0.4349 & 0.4142 & 0 & 0 \\ 0.0036 & -0.0061 & -0.1458 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

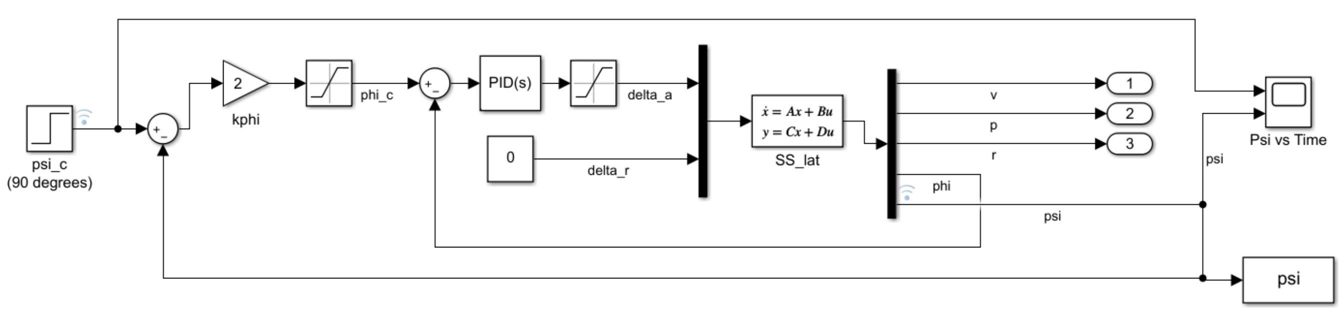
The control matrix was calculated using the control derivatives from Etkins (pg. 207).

$$B = \begin{bmatrix} 0 & 1.7188 \\ -0.1433 & 0.1146 \\ 0.0038 & -0.4859 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

We use the same C and D matrices as before to complete the state space model. The “ss” function in Matlab was again used to extract the transfer functions.

The block diagram for the heading control system is displayed below. Again, an enlarged version is available in the Simulink file.

Heading Control System

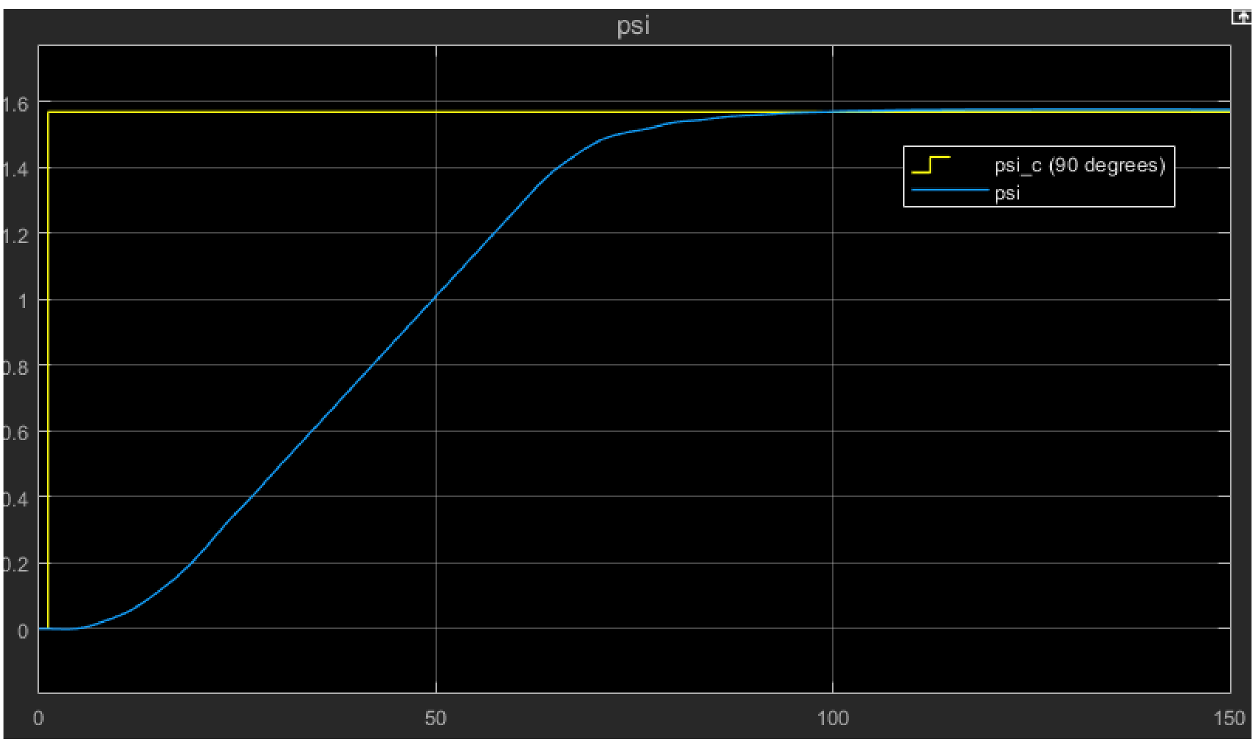


The proposed turning mechanism was to perform a steady banked turn using the ailerons to produce a desired roll angle. The yaw error was converted to a roll angle command using a somewhat arbitrarily selected gain of 2 (produced high response to minimize steady state error) and a limiter to maintain a maximum roll angle (25 degrees) that produced a smooth turn. The roll angle error was used to determine the required aileron deflection, which was controlled using a PID control law.

The PID coefficients were calculated using the PID tuner in Simulink. This was found to be more effective than Zeigler-Nichols tuning, and was easier to carry out with ongoing changes to the model. The PID tuning process is similar to that shown in the longitudinal model. The control law is summarized as

$$C(s) = K_p + K_i \frac{1}{s} + K_d s$$

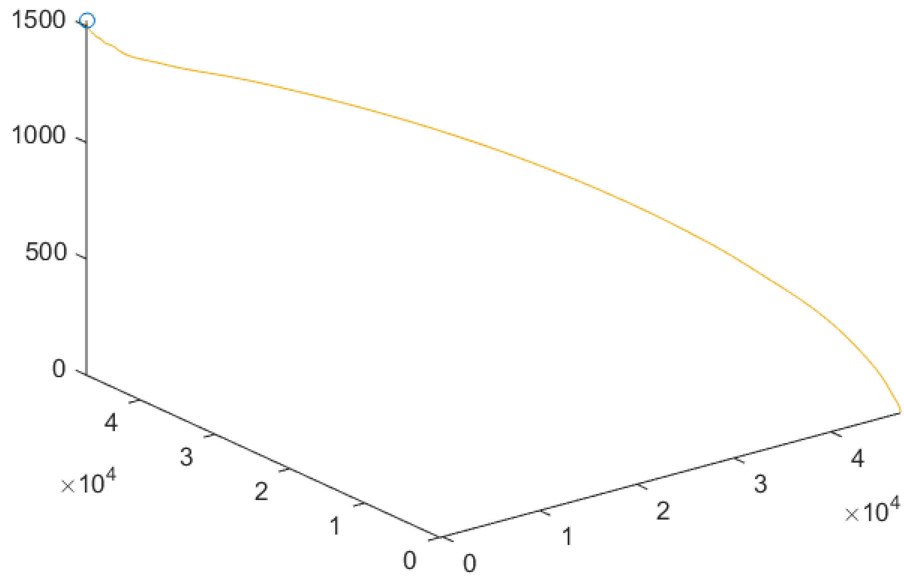
The aileron and rudder deflections are input into the lateral state space model to calculate the state vector. The yaw and roll angles were used as feedback to the desired values. The yaw angle step response is shown below, where the maximum overshoot was approximately 4.5% and the steady state error was well under 5%. The system takes approximately 100 seconds to complete the turn.

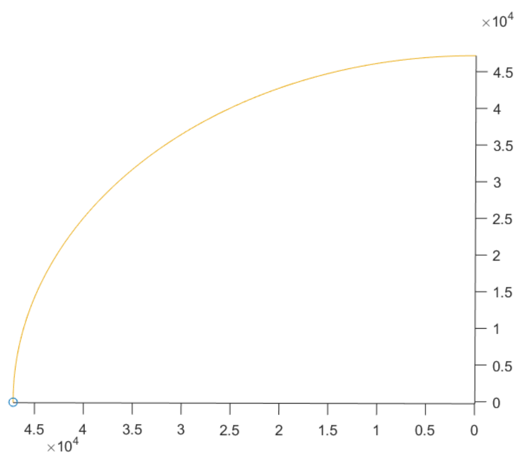


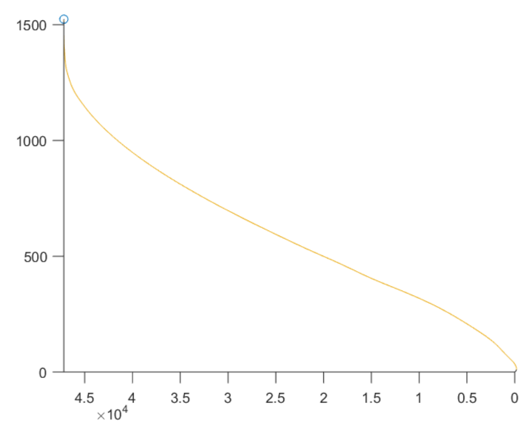
3) Simultaneous Turn/Ascend

The altitude and heading control systems were combined into the attached Simulink file as required.

A 3D model was developed to demonstrate the flight path of the airplane during the simultaneous turn and ascend command. The images below show the 3D flight path, as well as xy-plane and xz-plane views of the flight path.







These plots were made using the comet3 Matlab function. Simply run the Matlab file attached and observe the flight path simulation in real time.