Space Shuttle Altitude Simulation: Lab Report

Authors: Sampreet Sarkar[1], Varad Pradeep Diwakar[2]

Introduction

We have to control the attack angle of the space shuttle, which is depicted in the Figure 1 below:

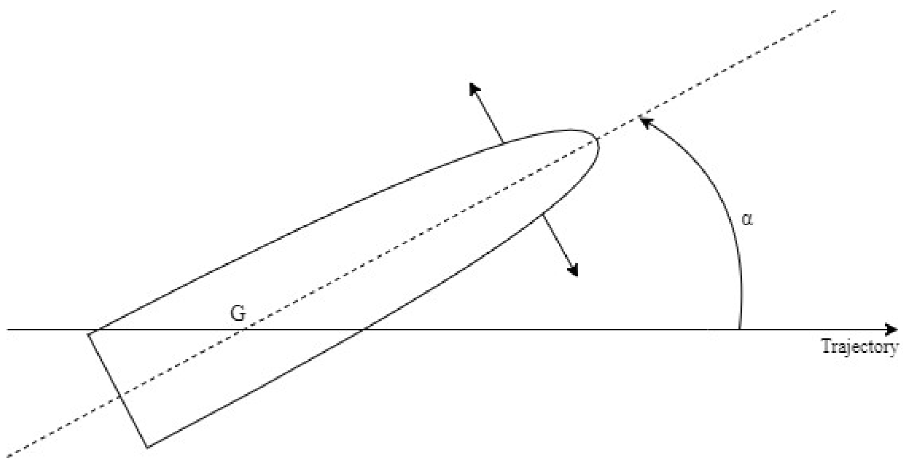

Figure 1: Space Shuttle

For simplicity, it is assumed that the aerodynamic pressure is constant, and ignore the effects of pitch and twist on the shuttle. Therefore, the lift will only depend on the attack angle α. Thus, using Newton's second law, we can find out the relation between the toque u and the attack angle α as shown below:

$$\frac{\ddot{\alpha}}{\omega_0^2}+\alpha-\alpha_{nom}=G\cdot u \qquad \text{. . . ..}(i)$$

where ω_0 is the angular frequency which depends upon the geometry of the shuttle, and α_{nom} is the nominal attack angle. The values to be used for the exercise are provided below:

$$\alpha_{nom} = 0.3 \text{ rad}$$

$$\omega_0 = 2 \text{ rad s}^{-1}$$

$$G = 1 \text{ rad. N}^{-1}\text{m}^{-1}$$

1 *M1 Control and Robotics: Embedded Real-Time Systems, Ecole Centrale de Nantes*
2 *M1 Control and Robotics: Embedded Real-Time Systems, Ecole Centrale de Nantes*

3.1 Transfer function and state-space representation

It is given that $\tilde{\alpha}=\alpha-\alpha_{nom}$ and $\omega=\dot{\alpha}$

a). In order to find the transfer function of the system with input u and output $\tilde{\alpha}$, we can proceed as follows:

From Eqn.(i), we have:

$$\frac{\ddot{\alpha}}{\omega_0^{2}}+\alpha-\alpha_{nom}=G\cdot u$$

Taking the Laplace Transform of both sides and considering all initial conditions to be zero, we have: $\frac{s^2\tilde{\alpha}(s)}{\omega_0^{2}}+\tilde{\alpha}(s)=G\cdot u(s)$ . . . . . $\{\tilde{\alpha}(s)=\alpha-\alpha_{nom}\}$

Therefore, $\tilde{\alpha}(s)\left(\frac{s^2}{\omega_0^{2}}+1\right)=G\cdot u(s)$

Since a transfer function is the representation of the Laplace transform of the output versus that of the input, we have:

$\frac{\tilde{\alpha}(s)}{u(s)}=\left(\frac{G}{\frac{s^2}{\omega_0^{2}}+1}\right)$, which can be simplified as : $\frac{\tilde{\alpha}(s)}{u(s)}=\frac{G\cdot\omega_0^{2}}{s^2+\omega_0^{2}}$ . . . . *(ii)*

Substituting the value of G = 1 rad. $N^{-1}m^{-1}$ and ω_0 = 2 rad s^{-1} in Eqn *(ii)*, we have –

$$\frac{\tilde{\alpha}(s)}{u(s)}=\frac{4}{s^2+4}$$

b). Now we will have to calculate the transfer function $Z\tilde{h}(z)$ discretized with sampling period T_s = *0.1s*, by step invariance method.

$$Z\tilde{h}(z)=\frac{Z(h*step)_s(z)}{Z\,step(z)} = (1-z^{-1})\,z(h*step)_s(z)$$

$$= (1-z^{-1})\,z\left[L^{-1}\left(\frac{Lh(s)}{s}\right)\right] \rightarrow \left[L^{-1}\left(\frac{Lh(s)}{s}\right)\right]=\frac{G\omega_0^{2}}{s(s^2+\omega_0^{2})} .$$

By using partial fraction expansion, we can write:

$$\frac{G\omega_0^{2}}{s(s^2+\omega_0^{2})}=\frac{A}{s}+\frac{Bs+C}{s^2+\omega_0^{2}} = G\omega_0^{2}=As^2+A\omega_0^{2}+Bs^2+Cs$$

Solving the equation, we can get

A = G , B = -G, and C = 0

$$\frac{G}{s}+\frac{(-Gs)+0}{s^2+\omega_0^{2}} = \frac{G}{s}-\frac{Gs}{s^2+\omega_0^{2}} = G\left(\frac{1}{s}-\frac{s}{s^2+\omega_0^{2}}\right) \quad . . . .(iii)$$

By taking inverse Laplace transform of equation *(iii)*, we get

$$((h{*}\,step)_s(t)) \quad = \quad G(1-\cos\omega_0 t)\,step(t) \qquad \text{. . . . . (iv)}$$

$$((h{*}\,step)_s(n)) = G(1-\cos\omega_0 nT_s)\,step(nT_s) \;= \qquad \text{. . . . . (v)}$$

$$z(h{*}\,step)_s(z) \quad = \quad z\big((h*step)_s[n]\big)$$

$$z(h*step)_s(z) = G\left(\frac{z}{z-1} - \frac{z(z-\cos\omega_0 Ts)}{z^2-2\,z\cos\omega_0 Ts+1}\right) \qquad \text{. . . . . (vi)}$$

$$Z\tilde{h}(z) = (1-z^{-1})\,z(h*step)_s(z) = G(1-z^{-1})\left(\frac{z}{z-1} - \frac{z(z-\cos\omega_0 Ts)}{z^2-2\,z\cos\omega_0 Ts+1}\right) \;.$$

Substituting value of G=1, Ts=0.1s, and ω_0 =2 rad/s, we get—

$$Z\tilde{h}(z) = \frac{0.01993\,z+0.01993}{z^2-1.96\,z+1}$$

c) State space representation of system.

$$\tilde{x}(t) = A\cdot x(t) + B\cdot u(t)$$
$$y(t) = C\cdot x(t) + D\cdot u(t)$$

A is state matrix or system matrix

B is input matrix

C is Output matrix or system matrix
D is direct feed through matrix.

$$\frac{\ddot{\alpha}}{\omega_0^2} + \alpha - \alpha_{nom} = G\cdot u \quad \text{....(system equation)}$$

Dimension of state matrix X can be decided by order of differential equation of system. Here the order of differential equation is 2, hence the dimension of the state vector *X* is 2. Let x_1 and x_2 be the two elements of state vector *x*.

$x_1 = \tilde{\alpha}$ . . . . *(vii)* and $x_2 = \dot{x}_1 = \dot{\tilde{\alpha}}$ . . . . *(viii)*

From equation *(vii)* we get— $\dot{x}_1 = x_2 = \dot{\tilde{\alpha}}$,
and from equation *(viii)* we get— $\dot{x}_2 = \ddot{x}_1 = \dot{\omega} = -\omega_0^2\tilde{\alpha} + G\omega_0^2 u$

from equation *(vii)* and *(viii)* we can describe them in matrix form as shown below—

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ G\,\omega_0^2 \end{bmatrix} u$$

$$y = \dot{\tilde{\alpha}} = \begin{bmatrix} 0 & 1 \end{bmatrix} * \begin{bmatrix} \dot{\tilde{\alpha}} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} u$$

From the above equations, we can write A, B, C, D matrix as following:

$$x = \begin{bmatrix} \dot{\tilde{\alpha}} \\ \dot{\omega} \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \qquad D = \begin{bmatrix} 0 \end{bmatrix}$$

d) Now finding discrete state space representation we will use following formula:

$$\tilde{x}[n+1] = \tilde{A}\tilde{x}[n] + \tilde{B}u,\ \tilde{y}[n] = \tilde{C}\tilde{x}[n] + \tilde{D}u \qquad \text{. . . . . (ix)}$$

now we can compute $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$ by using following formulae.

Where, $$\tilde{A} = e^{AT_s} \qquad \tilde{B} = \int_0^{T_S} e^{AT} B\, dT \qquad \tilde{C} = C \qquad \tilde{D} = D \qquad \text{. . . . . (x)}$$

We can find e^A by using formula $e^A = P e^s \bar{P}$ where e^s is diagonal matrix which can be found by using eigenvalues.

Let λ be the eigenvalue, now by finding determinant of (A-) we will get eigenvalues λ_1, and λ_2

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \rightarrow \lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \text{ ,therefore } A - \lambda I = \begin{bmatrix} -\lambda & 1 \\ -\omega_0^2 & -\lambda \end{bmatrix}.$$

by finding the determinants we will get $\lambda_1 = j\,\omega_0 \quad \lambda_2 = -j\,\omega_0$

Thus, $$A = \lambda I = \begin{bmatrix} -j\omega_0 & 0 \\ 0 & -j\omega_0 \end{bmatrix} \text{ and } e^{AT_s} = \begin{bmatrix} e^{j\omega_0 T_s} & 0 \\ 0 & e^{-j\omega_0 T_s} \end{bmatrix}$$

so corresponding to two eigenvalues we will get the eigenvector P[3]

for $\lambda 1$, we obtain $x = \begin{bmatrix} 1 \\ j\omega_0 \end{bmatrix}$ and for $\lambda 2$ $x = \begin{bmatrix} 1 \\ -j\omega_0 \end{bmatrix}$

Now we can write $P = \begin{bmatrix} 1 & 1 \\ j\omega_0 & -j\omega_0 \end{bmatrix}$ and $\bar{P} = \begin{bmatrix} 1/2 & 1/2\,j\omega_0 \\ 1/2 & -1/2\,j\omega_0 \end{bmatrix}$

3 By using the formula $(A - \lambda I)x = 0$

The matrix $\widetilde{A}$ can be calculated as:

$$\widetilde{A} = P e^{s} \bar{P} = \begin{bmatrix} 1 & 1 \\ j\omega_0 & -j\omega_0 \end{bmatrix} \begin{bmatrix} e^{j\omega_0 T_s} & 0 \\ 0 & e^{-j\omega_0 T_s} \end{bmatrix} \begin{bmatrix} 1/2 & 1/2\, j\omega_0 \\ 1/2 & -1/2\, j\omega_0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\omega_0 T_s & \sin\omega_0 T_s / \omega_0 \\ -\omega_0 \sin\omega_0 T_s & -\cos\omega_0 T_s \end{bmatrix}$$

$$\widetilde{A} = \begin{bmatrix} \cos\omega_0 T_s & \sin\omega_0 T_s / \omega_0 \\ -\omega_0 \sin\omega_0 T_s & -\cos\omega_0 T_s \end{bmatrix} = \widetilde{A} = \begin{bmatrix} 0.98 & 0.1 \\ -0.40 & 0.98 \end{bmatrix}$$

$$\widetilde{B} = \int_0^{T_S} e^{AT} B\, dT = \int_0^{T_S} \begin{bmatrix} \cos\omega_0 T & \sin\omega_0 T / \omega_0 \\ -\omega_0 \sin\omega_0 T & -\cos\omega_0 T \end{bmatrix} \begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix} dT$$

$$\widetilde{B} = \int_0^{T} \begin{bmatrix} G\omega_0 \sin\omega_0 T_s \\ G\omega_0^2 \cos\omega_0 T \end{bmatrix} dT$$

$$\widetilde{B} = \begin{bmatrix} G(1-\omega_0 \cos\omega_0 T_s) \\ G\omega_0^2 \sin\omega_0 T_s \end{bmatrix} = \widetilde{B} = \begin{bmatrix} 0.02 \\ 0.4 \end{bmatrix}$$

$\widetilde{C} = C$ and $\widetilde{D} = D$

3.2 Pulse Width Modulator(PWM)

The jet engines can only be switched on and off, they deliver normal torque E. At each time there is one jet switched on. The actual torque input of the shuttle is then um such that :

$$u_m = E \cdot sign(u - p)$$

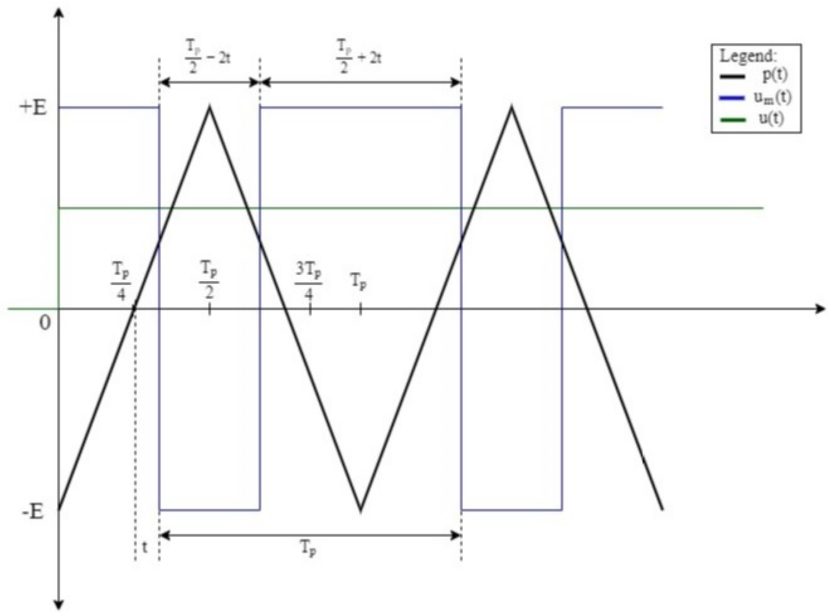

Figure 2: *The PWM signal*

From the above figure it is shown that signal um is periodic with the period equal to that of triangular wave. Period of triangular waveform is T_p. It is given that wanted torque u is constant between -E to +E.

1. value of u_m is -E for duration T_p-$2t$. (refer above figure). We will this duration as T1

2. value of u_m is -E for duration T_p+$2t$. (refer above figure). We will this duration as T2

We can obtain the value of t from relation,

$$\frac{4E}{Tp}=\frac{u}{t} \quad \text{and} \quad t=\frac{u\,Tp}{4E} \qquad \text{. . . . . (xi)}$$

The mean value of $\vec{u}_m$ can be calculated as following:

$$\vec{u}_m=\frac{T1+T2}{Tp} = \frac{(-E*(\frac{Tp}{2}-2t))+(E*(\frac{Tp}{2}+2t))}{Tp} = \frac{4tE}{Tp}=u \qquad \text{. . . . . (xii)}$$

Hence the mean value of actual torque input um is equal to constant u between -E to +E.

Now if |u(t)| > E then controller output signal u and triangular waveform signal p will the intersect. As we can not compare the signal[4] and we will get actual torque um is equal to constant value E.*(straight horizontal line)*

4 The effect of *sign(u – p)* is reduced to zero

3.3 Open-loop Simulation(`MATLAB`)

In this section, we compared and plotted the continuous time and discrete time system. The MATLAB script is provided below for your convenience. In the first part of the exercise, we had to perform the numerical computation of the system. When comparing with our own results, we see that the answers hold up to their value. We then implemented a function called ***navettecontinue***, in order to describe the differential equation of the state vector of the continuous-time system. We did the same for the discrete-time system, by writing a function called ***navettediscrete***. We had substituted the call to the function ode45, by the function ***ore***, which would be our solver for this exercise. We also calculated the run time of the continuous and discrete time systems which are 0.5456s and 0.0859s respectively. This is because the continuous time system has higher sampling frequency than that of discrete time system. Therefore it has more sampling points than discrete time system during the same sampling time. Finally, we plot the step response of both the continuous-time and discrete-time systems on the same figure, shown below:

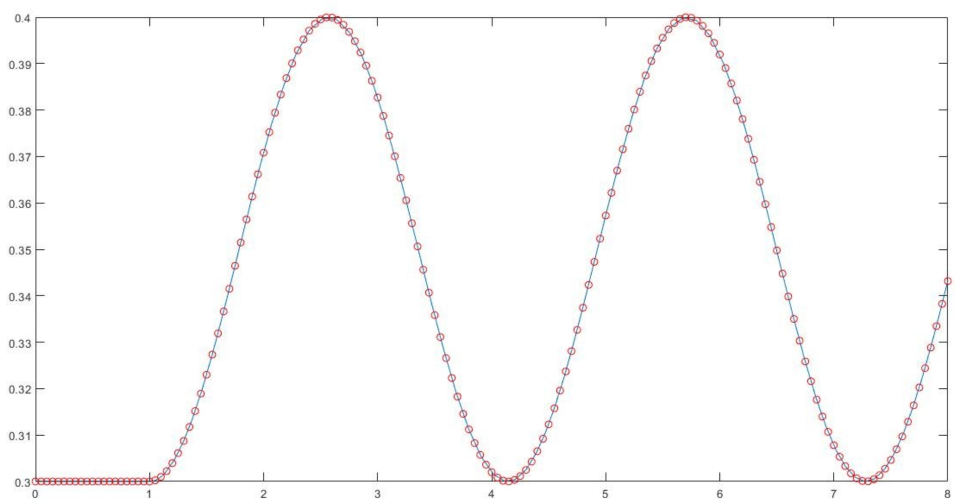

Figure 3:

3.4 Open-loop Simulation (`Simulink`)

In this exercise, we utilize the system modeling software Simulink to create and analyze a model of our system. We will perform analyses on both the continuous-time system and the discrete time system. The model is shown in the figure below:

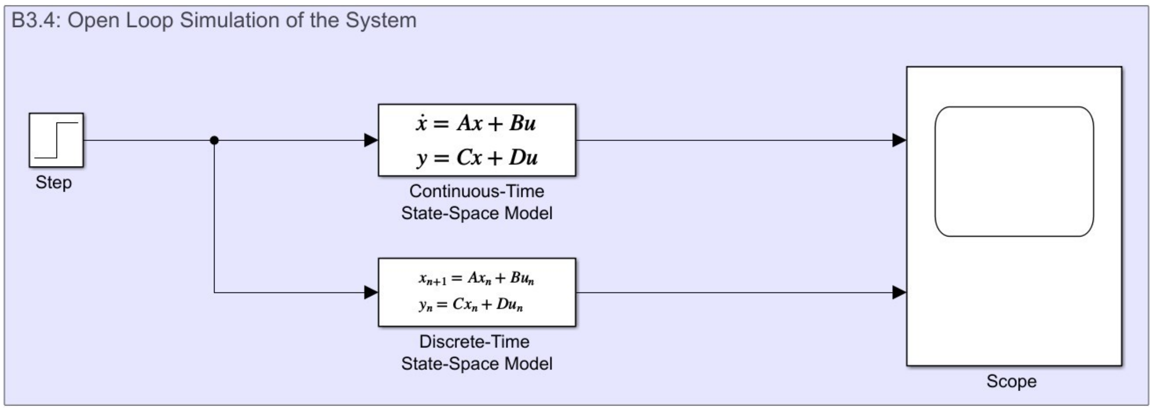

Figure 4:

The output from the scope should be similar to the output we have observed in the previous exercise, as is given in the figure below:

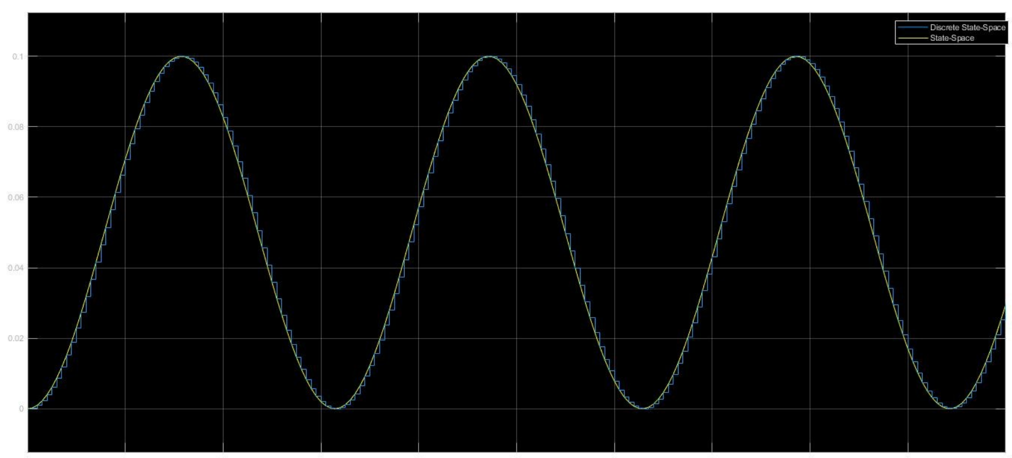

Figure 5:

3.5 Closed-loop Simulation

In this exercise, we introduce a discrete-time controller which will enable the system to track a reference signal a_c. It is provided to us that the sampling time T_s of the controller is *0.1s*. The controller is implemented according to the following equation:

$$Z\,u_s[n] = K_I \frac{T_s}{1-z^{-1}}\left(Z\,\tilde{\alpha}_{cs}(z) - Z\,\tilde{\alpha}_s(z)\right) - K_p\,Z\,\tilde{\alpha}_s(z) - K_D\frac{1-z^{1}}{T_s}Z\,\tilde{\alpha}_s(z) \quad \ldots\ldots\ldots\ldots \textit{(xiii)}$$

where, $K_p = 2.70\ N\ m\ rad^{-1}s^{-1}$, $K_p = 1.25\ N\ m\ rad^{-1}$, and $K_D = 0.75\ N\ m\ rad^{-1}$.
The model with the controller implemented can be seen in the figure below:

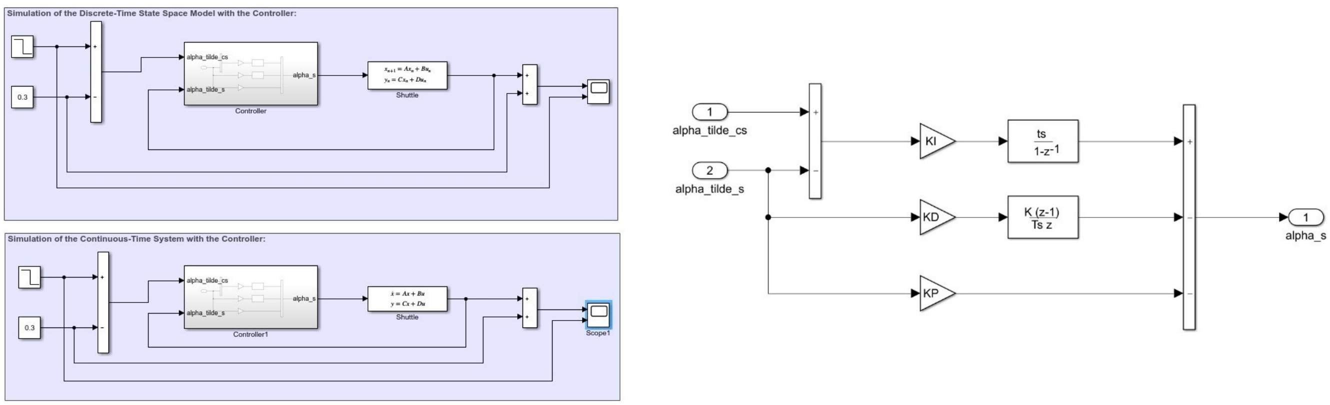

Figure 6:

We are supposed to simulate both the continuous-time system and the discrete-time system with the discrete-time controller, and plot the reference signal α_c and the output signal α on the same plot. It is said that the input signal should be a step which goes from 0.3 rad to 0.27 rad at time *t=1s*. The scope output can be observed in the figure below:

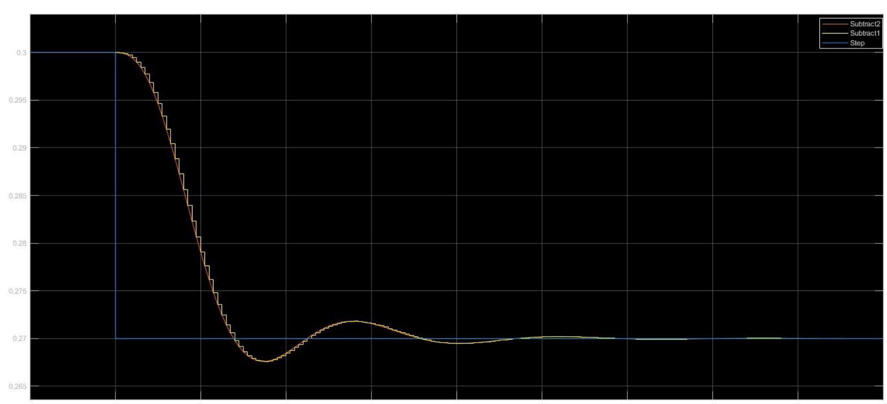

Figure 7:

We can observe the initial step input as the blue line, the continuous-time system output as the orange line, and the discrete-time system output with the yellow line.

The variable step and fixed step solvers seem to give the same response, indicating that the variable step size chosen automatically by the solver is optimal as well for this closed loop system.

3.6 Closed-loop Simulation with PWM

In this exercise, we are told that the signal from the controller now travels through a PWM before reaching the system. We have to now model this PWM section, given $T_{pwm} = 0.02s$, and the torque produced by the actuators is $E = 0.05Nm$. The model for the PWM based controller can be found below:

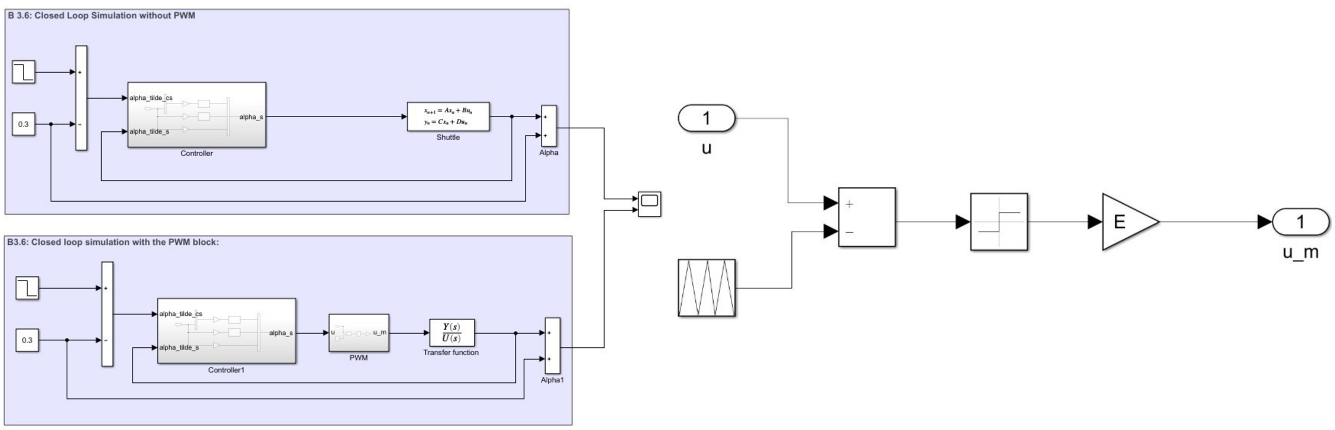

Figure 8:

We have used the Dormand-Prince solver, and we will observe the scope outputs in the following cases:

1. Solver: Variable step solver and step size Auto.

- Max step size is the largest time step that the solver can take ('auto' means 50 steps). Here we observe that computation time in case of continuous time systems with Pulse Width Modulated signal is greater than in case of discrete time system.

- We will see that the continuous time response does not track the discrete time response .This is because the automatically chosen step size is too large and hence, frequency is too low to capture the high frequency PWM signal. Max step size of solver must be less than period of PWM signal used in the model. If this condition is not satisfied then PWM will not be appropriately sampled by the Solver.

2. Solver: Variable step solver and step size $T_{pwm}/10 = 0.002$ s

- When we set max step size to T$pwm/10$, we sample at a higher frequency than the PWM signal, so we are able to capture all its frequencies (as per Shannon's sampling theorem ($f_s > 2f_{max}$) only if sampling frequency is high enough)

- We will see that the output does track the input as closely as when we set max step size to 0.002.

We will see that the variable step response does not track the input as closely as the fixed step response. This is because the automatically chosen step size is too large and hence, frequency is too low to capture the high frequency PWM signal.

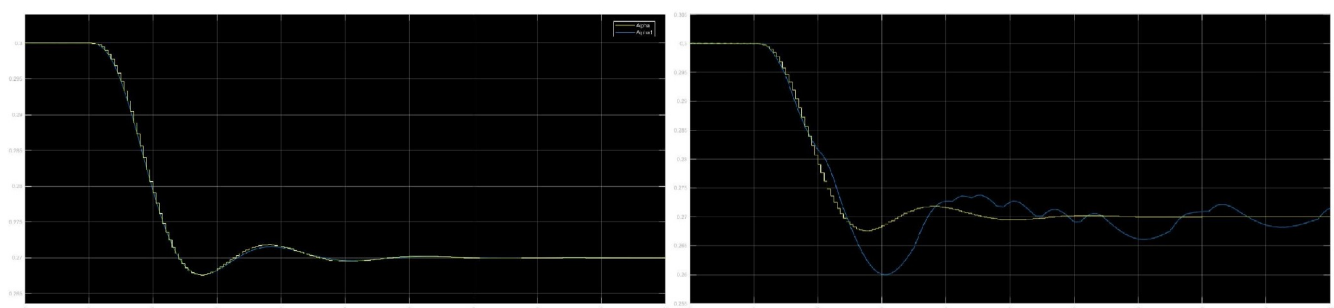

Figure 9:

Appendix: Code Reference

All the MATLAB scripts and Simulink models are provided for your convenience. In order to access the file, you can just click on the link[5].

`main.m`: Contains the main program used in the Exercise B3.3 onward.

`navettecontinue.m`: Contains the implementation of the MATLAB function which describes the differential equation of the state vector in the continuous-time domain.

`navettediscrete.m`: Contains the implementation of the MATLAB function which describes the recursion of the state vector of the discrete system.

`control_sipro_b35.slx`: Contains the Simulink model used in the Exercise B3.4

`control_sipro_wController.slx`: Contains the Simulink model used in Exercise B3.5

`sipro_B36_withPWM.slx`: Contains the Simulink model used in the Exercise B3.6.

5 Please make sure you have at least MATLAB 2018a while accessing the Simulink models