Laboratory N.3

MICHELE TARTARI, LUCIA BERGANTIN, OLENA HRYSHAIENKO

January 4, 2018

Abstract

In this report, referring to the third lab, we concentrate on the system aspect of the course. The aim is to analyze a concrete example as the control of a space shuttle during an aero-assisted orbit transfer.

Contents

1. Space shuttle attitude simulation

1.1. Transfer function and state space representation

In this part we have to analyze theoretically the behavior of the shuttle. We have to consider that it bounces on high atmosphere layers to follow a reference trajectory and this is performed by means of lift force tuned by means of attitude angles control. Therefore the behavior depends on the aerodynamic pressure, that we assume constant, given by the formula $\frac{1}{2} \cdot \rho \cdot v^2$, where ρ is the air density and v the velocity. The system can therefore be described by the equation

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

B8.1 $\dfrac{\ddot{\alpha}}{\omega_0^2} + \alpha - \alpha_{NOM} = Gu$

$\dfrac{\dot{\omega}}{\omega_0^2} + \tilde{\alpha} = Gu \qquad \dot{\omega} = -\tilde{\alpha}\omega_0^2 + Gu\omega_0^2$

$$\begin{cases} \omega = \dot{\alpha} \\ \dot{\omega} = -\tilde{\alpha}\omega_0^2 + Gu\omega_0^2 \end{cases}$$

$$\begin{cases} \begin{bmatrix} \omega \\ \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix}}_{A} \begin{bmatrix} \tilde{\alpha} \\ \omega \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix}}_{B} u \\ \alpha = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \begin{bmatrix} \tilde{\alpha} \\ \omega \end{bmatrix} + \alpha_{NOM} + \underbrace{\begin{bmatrix} 0 \end{bmatrix}}_{D} u \end{cases}$$

STATE SPACE REPRESENTATION

(a) Transfer function $\mathcal{L}h(s)$:

that we assume

since $\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \xrightarrow[\text{Laplace transform}]{} \begin{cases} s\mathcal{L}x(s) - \underbrace{x(0)}_{=0} = A\mathcal{L}x(s) + B\mathcal{L}u(s) \\ \mathcal{L}y(s) = C\mathcal{L}x(s) + \underbrace{D}_{=0}\mathcal{L}u(s) \end{cases}$

$$\mathcal{L}x(s) = \frac{B\mathcal{L}u(s)}{sI - A} = (sI-A)^{-1} B \mathcal{L}u(s)$$

$$\mathcal{L}y(s) = C(sI-A)^{-1}B\mathcal{L}u(s)$$

$$\mathcal{L}h(s) = \frac{\mathcal{L}y(s)}{\mathcal{L}u(s)} = C(sI-A)^{-1}B.$$

then we can use the state space representation matrices to calculate the transfer function:

$$(sI-A)^{-1} = \left(\begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix}\right)^{-1} = \begin{bmatrix} s & -1 \\ \omega_0^2 & s \end{bmatrix}^{-1} =$$

$$= \frac{1}{s^2+\omega_0^2}\begin{bmatrix} s & 1 \\ -\omega_0^2 & s \end{bmatrix}$$

$$\mathcal{L}h(s) = C(sI-A)^{-1}B = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \frac{1}{s^2+\omega_0^2}\begin{bmatrix} s & 1 \\ -\omega_0^2 & s \end{bmatrix}\begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix} =$$

$$= \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} \frac{s}{s^2+\omega_0^2} & \frac{1}{s^2+\omega_0^2} \\ -\frac{\omega_0^2}{s^2+\omega_0^2} & \frac{s}{s^2+\omega_0^2} \end{bmatrix}\begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{s}{s^2+\omega_0^2} & \frac{1}{s^2+\omega_0^2} \end{bmatrix}\begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix} = \frac{G\omega_0^2}{s^2+\omega_0^2} = \frac{4}{s^2+4}$$

substituting the given values

$$\mathcal{L}h(s) = G\frac{\omega_0^2}{s^2+\omega_0^2} = \frac{4}{s^2+4}$$

(b) Transfer function $\mathcal{Z}H(z)$ with $T_s = 0{,}1\,s$ by step variance method:

step invariance method: $\mathcal{Z}H(z) = \dfrac{\mathcal{Z}(h * step)_s}{\mathcal{Z}\, step}$

$$\mathcal{L}(h * step)(s) = \mathcal{L}h(s)\,\mathcal{L}step(s) = \frac{G\omega_0^2}{s(s^2+\omega_0^2)}$$

$$\frac{\omega_0^2}{s(s^2+\omega_0^2)} = \frac{a}{s} + \frac{bs+c}{s^2+\omega_0^2} = \frac{as^2+a\omega_0^2+bs^2+cs}{s(s^2+\omega_0^2)}$$

$$\begin{cases} a+b=0 \\ c=0 \\ a\omega_0^2 = \omega_0^2 \end{cases} \qquad \begin{cases} b=-a=-1 \\ c=0 \\ a=1 \end{cases}$$

$$\mathcal{L}(h * step)(s) = \frac{G}{s} - \frac{Gs}{s^2+\omega_0^2}$$

$$(h * step)(t) = G\,step(t) - G\cos(\omega_0 t)\,step(t) =$$
$$= G(1-\cos(\omega_0 t))\,step(t)$$
$$(h * step)(nT_s) = G(1-\cos(\omega_0 nT_s))\,step(nT_s) =$$
$$(h * step)_s[n] = G(1-\cos(\omega_0 nT_s))\,step[n]$$

z-transform:

$$\mathcal{Z}(h * step)_s(z) = G\left(\frac{1}{1-z^{-1}} - \frac{1-z^{-1}\cos(\omega_0 T_s)}{1-2z^{-1}\cos(\omega_0 T_s)+z^{-2}}\right)$$

Since $\mathcal{Z}step(z) = \frac{1}{1-z^{-1}}$:

$$\mathcal{Z}h(z) = \frac{\mathcal{Z}(h * step)_s}{\mathcal{Z}step} = G\left(\frac{1}{1-z^{-1}} - \frac{1-z^{-1}\cos(\omega_0 T_s)}{1-2z^{-1}\cos(\omega_0 T_s)+z^{-2}}\right)\cdot$$
$$\cdot(1-z^{-1}) = G\left[1 - \frac{(1-z^{-1}\cos(\omega_0 T_s))(1-z^{-1})}{1-2z^{-1}\cos(\omega_0 T_s)+z^{-2}}\right] =$$
$$= G\,\frac{(1-\cos(\omega_0 T_s))(z^{-1}+z^{-2})}{1-2z^{-1}\cos(\omega_0 T_s)+z^{-2}}$$

Substituting the given values:

$$\mathcal{Z}h(z) \simeq \frac{0{,}02\,(z^{-1}+z^{-2})}{1-1{,}96z^{-1}+z^{-2}}$$

(c) State space representation:

$$A = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

Substituting the given values:

$$A = \begin{bmatrix} 0 & 1 \\ -4 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 4 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 \end{bmatrix}$$

(d) Discretized state space representation using step invariance method:

$$\begin{cases} \tilde{A} = P e^{AT_s} P^{-1} \\ \tilde{B} = \int_0^{T_s} \tilde{A}(t) B\,dt \end{cases} \qquad \begin{cases} \tilde{x}[n+1] = \tilde{A}\tilde{x}[n] + \tilde{B}u[n] \\ y[n] = C\tilde{x}[n] + \underbrace{D\tilde{u}[n]}_{=0} \end{cases}$$

To compute e^{AT_s} we have to find first the eigenvalues of A:

$$\det(sI - A) = s^2 + \omega_0^2 = 0 \qquad s^2 = -\omega_0^2$$

therefore the eigenvalues are imaginary

$$s = \pm i\omega_0$$

$$e^{AT_s} = \begin{bmatrix} e^{\lambda_1 T_s} & \cdots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n T_s} \end{bmatrix} = \begin{bmatrix} e^{i\omega_0 T_s} & 0 \\ 0 & e^{-i\omega_0 T_s} \end{bmatrix}$$

To calculate P we have to find the eigenvectors of A:

$$A\underline{x}_1 = s_1 \underline{x}_1 \qquad \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = i\omega_0 \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix}$$

$$\begin{cases} x_{12} = i\omega_0 x_{11} \\ -\omega_0^2 x_{11} = i\omega_0 x_{12} \end{cases} \qquad \begin{cases} x_{12} = i\omega_0 x_{11} \\ x_{12} = -\frac{\omega_0}{i} x_{11} = i\omega_0 x_{11} \end{cases} \qquad \underline{x}_1 = \begin{bmatrix} 1 \\ i\omega_0 \end{bmatrix}$$

$$A\underline{x}_2 = s_2 \underline{x}_2 \qquad \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & 0 \end{bmatrix} \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} = -i\omega_0 \begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix}$$

$$\begin{cases} x_{22} = -i\omega_0 x_{21} \\ -\omega_0^2 x_{21} = -i\omega_0 x_{22} \end{cases} \qquad \underline{x}_2 = \begin{bmatrix} 1 \\ -i\omega_0 \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 1 \\ i\omega_0 & -i\omega_0 \end{bmatrix} \qquad P^{-1} = \frac{1}{2} \begin{bmatrix} 1 & \frac{1}{i\omega_0} \\ 1 & -\frac{1}{i\omega_0} \end{bmatrix}$$

$$A = Pe^{AT_s}P^{-1} = \begin{bmatrix} 1 & 1 \\ i\omega_0 & -i\omega_0 \end{bmatrix} \cdot \begin{bmatrix} e^{i\omega_0 T_s} & 0 \\ 0 & e^{-i\omega_0 T_s} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2i\omega_0} \\ \frac{1}{2} & -\frac{1}{2i\omega_0} \end{bmatrix} =$$

$$= \begin{bmatrix} e^{i\omega_0 T_s} & e^{-i\omega_0 T_s} \\ i\omega_0 e^{i\omega_0 T_s} & -i\omega_0 e^{-i\omega_0 T_s} \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2i\omega_0} \\ \frac{1}{2} & -\frac{1}{2i\omega_0} \end{bmatrix} =$$

$$= \begin{bmatrix} \dfrac{e^{i\omega_0 T_s} + e^{-i\omega_0 T_s}}{2} & \dfrac{e^{i\omega_0 T_s} - e^{-i\omega_0 T_s}}{2i\omega_0} \\ -\omega_0 \dfrac{e^{i\omega_0 T_s} - e^{-i\omega_0 T_s}}{2i} & \dfrac{e^{i\omega_0 T_s} + e^{-i\omega_0 T_s}}{2} \end{bmatrix} =$$

$$= \begin{bmatrix} \cos(\omega_0 T_s) & \dfrac{\sin(\omega_0 T_s)}{\omega_0} \\ -\omega_0 \sin(\omega_0 T_s) & \cos(\omega_0 T_s) \end{bmatrix}$$

$$\tilde{B} = \int_0^{T_s} A(\tau) B \, d\tau = \int_0^{T_s} \begin{bmatrix} \cos(\omega_0 \tau) & \dfrac{\sin(\omega_0 \tau)}{\omega_0} \\ -\omega_0 \sin(\omega_0 \tau) & \cos(\omega_0 \tau) \end{bmatrix} \begin{bmatrix} 0 \\ G\omega_0^2 \end{bmatrix} d\tau =$$

$$= \int_0^{T_s} \begin{bmatrix} G\omega_0 \sin(\omega_0 \tau) \\ G\omega_0^2 \cos(\omega_0 \tau) \end{bmatrix} d\tau = \begin{bmatrix} -G\cos(\omega_0 \tau) \\ G\omega_0 \sin(\omega_0 \tau) \end{bmatrix}_0^{T_s} =$$

$$= \begin{bmatrix} -G\cos(\omega_0 T_s) + G \\ G\omega_0 \sin(\omega_0 T_s) \end{bmatrix} = G \begin{bmatrix} 1 - \cos(\omega_0 T_s) \\ \omega_0 \sin(\omega_0 T_s) \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} \cos(\omega_0 T_s) & \dfrac{\sin(\omega_0 T_s)}{\omega_0} \\ -\omega_0 \sin(\omega_0 T_s) & \cos(\omega_0 T_s) \end{bmatrix} = \begin{bmatrix} 0{,}98 & 0{,}1 \\ -0{,}4 & 0{,}98 \end{bmatrix}$$

$$\tilde{B} = G \begin{bmatrix} 1 - \cos(\omega_0 T_s) \\ \omega_0 \sin(\omega_0 T_s) \end{bmatrix} = \begin{bmatrix} 0{,}02 \\ 0{,}4 \end{bmatrix}$$

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

1.2. Pulse width modulator (PWM)

In this second part a technical constraint will be taken into account: the shuttle's jet engines that deliver the torque E would only be switched on or off not allowing continuous modulation, being controlled by means of a PWM (pulse width modulator) whose output signal u is compared to a triangle waveform p. Therefore the torque input becomes

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

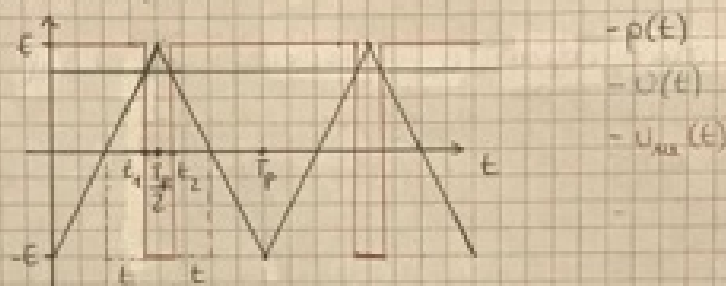
The aim is now to demonstrate that if the wanted torque u is a constant whose value is between -E and +E, then the mean value of the actual torque u_m is equal to this constant. We should then analyze the case in which its value is located outside these boundaries.

B.8.2 We can rewrite the signal u_m as:

$$u_m = \begin{cases} E & 0<t<t_1 \\ -E & t_1<t<t_2 \end{cases}$$

with t_1 and t_2 depending from the value of $v(t)$

For example:



Demonstration that the mean value of u_m is equal to $v(t)$, assuming $-E<v(t)<E$

We can write that:

$$\begin{cases} t_1 = \frac{T_p}{2} - 2t \\ t_2 = \frac{T_p}{2} + 2t \end{cases}$$

We can then deduce that

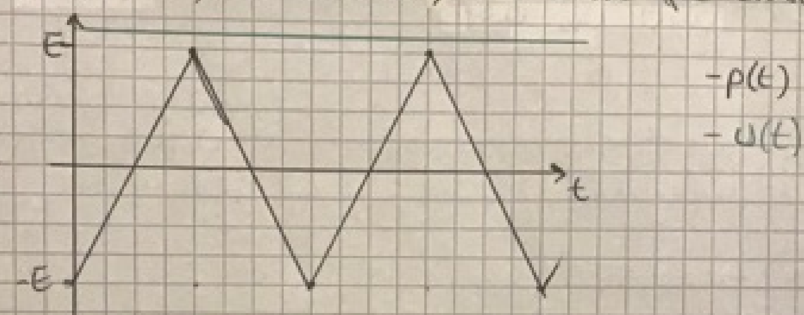
$$\bar{u}_m = \frac{-Et_1 + Et_2}{t_1+t_2} = \frac{-E\left(\frac{T_p}{2}-2t\right) + E\left(\frac{T_p}{2}+2t\right)}{\left(\frac{T_p}{2}+2t\right)+\left(\frac{T_p}{2}-2t\right)} = \frac{E4t}{T_p}$$

We have to calculate t, that be deduce by Thales Theorem:

$$\frac{E}{\frac{T_p}{4}} = \frac{v}{t} \qquad t = \frac{v}{\frac{4E}{T_p}} = \frac{T_p}{4E}v$$

Therefore: $\bar{u}_m = \frac{E4}{T_p}t = \frac{E4}{T_p}\cdot\frac{T_p}{4E}v = v$

If $|u(t)| > E$, however, we have for instance:



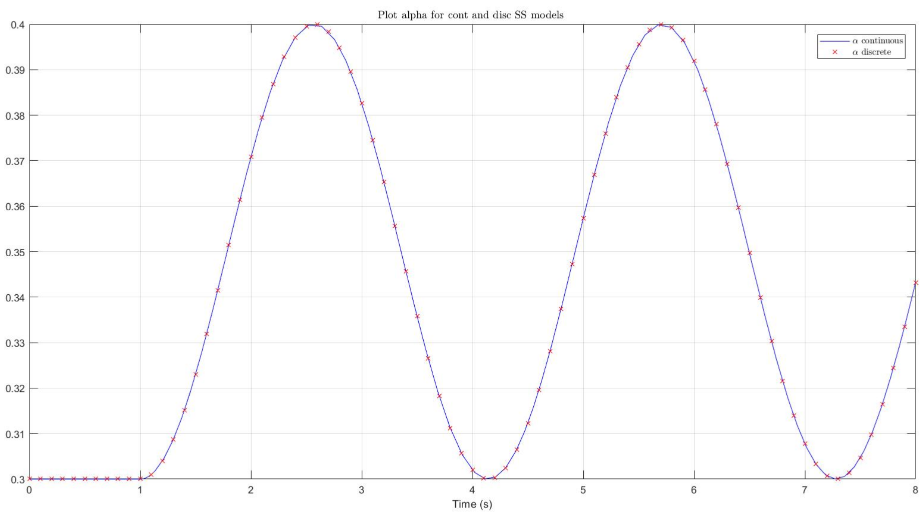
In this case we have saturation:

if $u(t) > E$ $\quad u_m = E\,\text{sign}(u-p) = E$

if $u(t) < -E$ $\quad u_m = E\,\text{sign}(u-p) = -E$

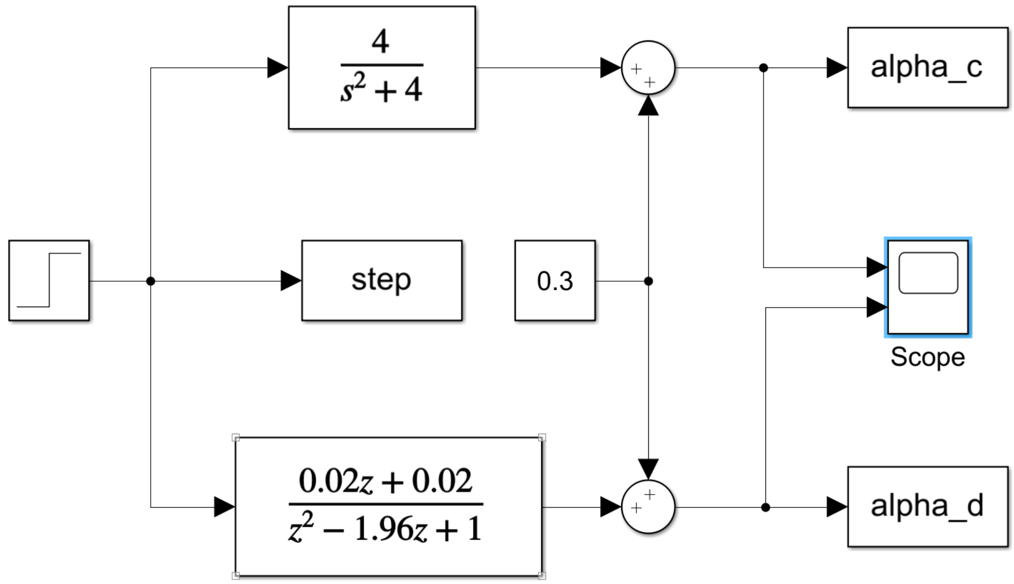
1.3. Matlab Simulation

As part of this task we simulated the response of the angle α, initially at its nominal value 0.3rad, from time $t_0 = 0s$ to $t_1 = 8s$ when the torque input u is a step from 0 to 0.05Nm at time $t_0 = 1s$. In the continuous case, the differential equation $\dot{x} = Ax + Bu$ can therefore be solved by the means of the function ode45. However, in the discrete case, the recursion $x_s[n+1] = \tilde{A}x_s[n] + \tilde{B}u_s[n]$ has to be solved by the means of the costume function. At first we wrote a Matlab script main.m, inserting the numerical computation of the discrete time transfer function and state space representation obtained with the theoretical calculation shown in the section 1.1. We then coded two functions navettecontinue and navettediscrete to compute the derivative $\dot{x}$ at each iteration in continuous and discrete time respectively and then used the functions ode45 and ore to solve the differential equations as mentioned above. The plot below shows the value of α for the different cases:

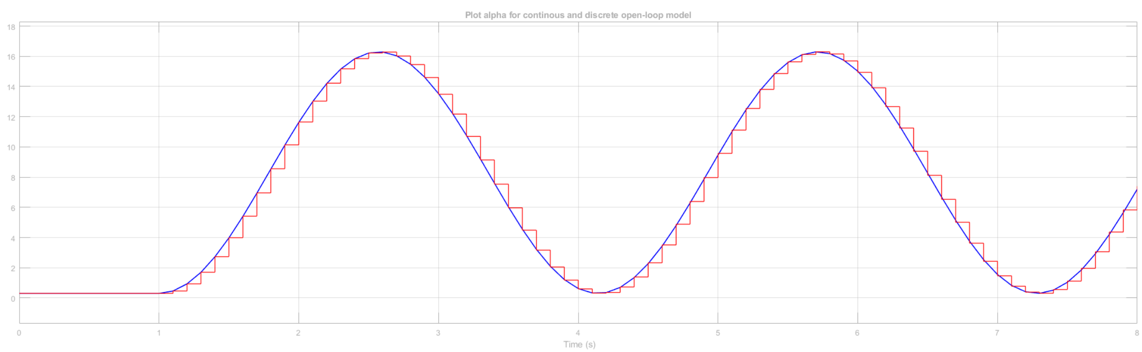


1.4. Simulink Simulation

We then implemented the previous task in the Simulink environment, generating the following scheme:



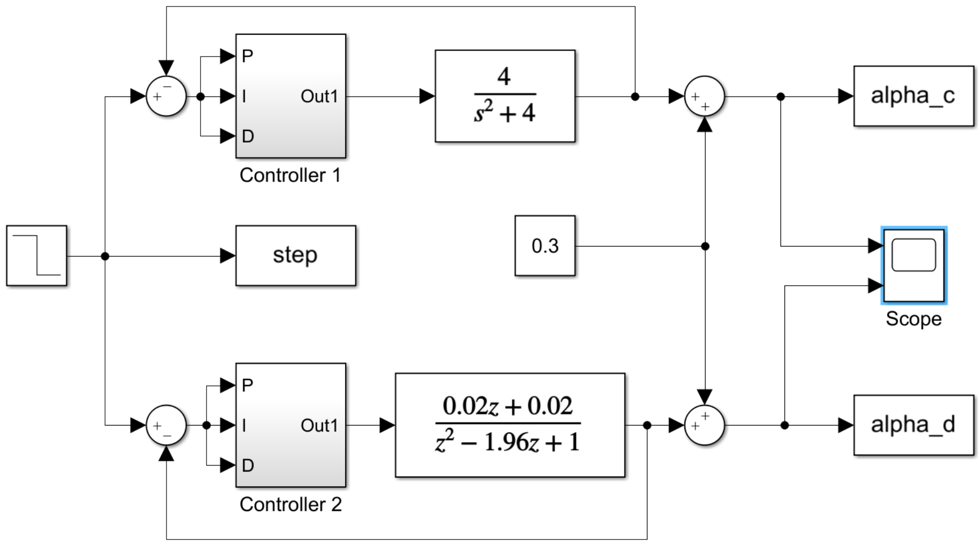
Which plots:



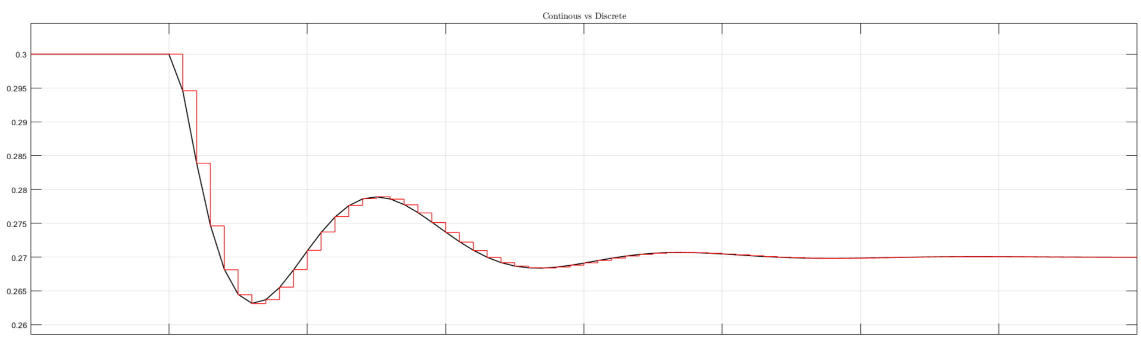
In this case to the plot the $\alpha_{discrete}$ we hold the value, obtaining a step-wave signal otherwise identical to the one obtained using Matlab. Thus the output of the Matlab script and of the Simulink simulation correspond, as expected.

1.5. PID controller simulation

We now supposed that the pilot, to modify the lift force, wants a jump from 0.3rad to 0.27rad at $t = 1s$. In this case to implement the system we have to add a PID controller and close the loop as shown in the scheme below:

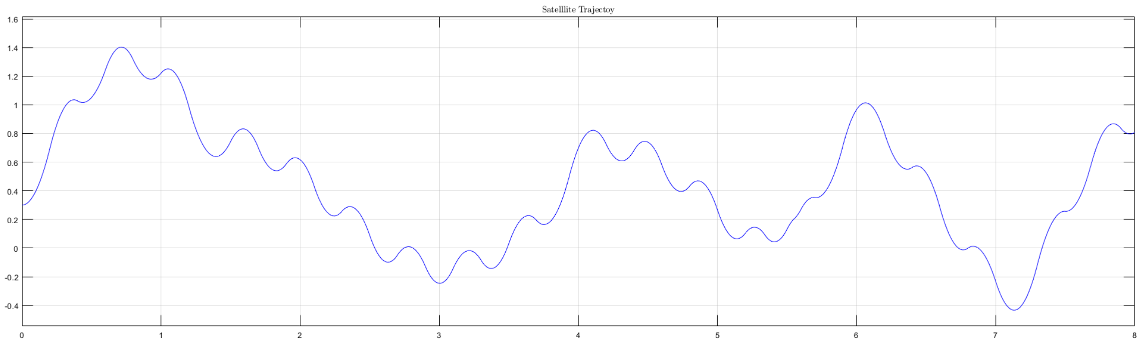


This is done to consider the system's feedback output in order to compensate the error by the means of the controller. As shown above, we still consider both the continuous and the discrete cases. The scheme plots:

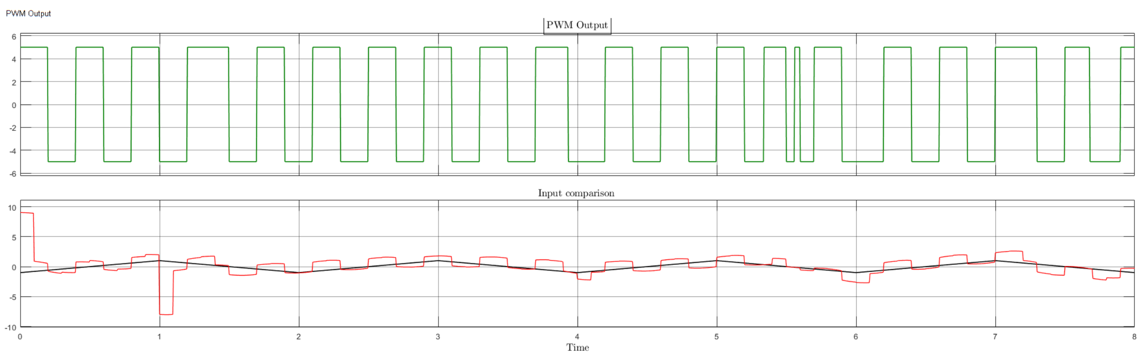


1.6. PWM simulation

In this task we implemented in the Simulink environment what calculated theoretically in the section 1.2.



Since now we consider to have two engines that can't work in continuous and that only one of the two can be turn on at a specific time, we use a PWM that compares a triangular wave with the PID output and gets as an output what visible below:



In the first plot we can see the PWM output which is given directly as an input to the engines, while the second shows the two signals compared.

Conclusions

In this paper we successfully worked on a concrete example that is the control of a space shuttle during an aero-assisted orbit transfer. In section 1.1 and 1.2 we analyzed the theoretical background needed to understand and implement all the various tasks. Section 1.3 and 1.4 we represented the system and plotted the attack angle α in both Matlab and Simulink environments for both the discrete and the continuous case. Section 1.5 introduced a PID to control the system and in section 1.6 we consider to have two engines that can't work in continuous and that only one of the two can be turn on at a specific time, and control them using PWM.