Model Predictive Control - Assignment 1

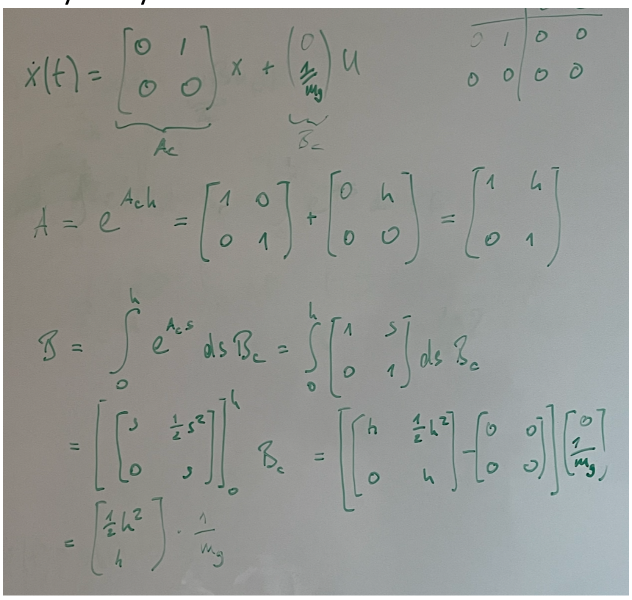
Question 1: See code astrobee.py line 56/57

Question 2:

Computed with casadi_c2d:

```
Ad = [[1.   0.1]
 [0.   1. ]]
Bd = [[0.00024417]
 [0.00478469]]
Cd =  [1, 0]
Poles: [1.+0.j 1.+0.j]
Zeros: [-0.9595789+0.j]
```

Analytically discretized:



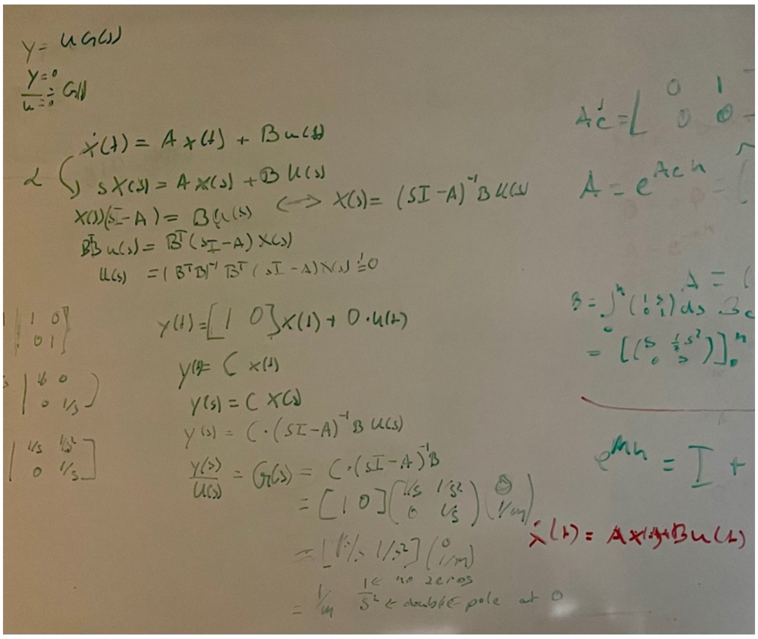
$$=> A = \begin{pmatrix} 1 & 0.1 \\ 0 & 1 \end{pmatrix} \; and \; B = \frac{1}{m_g} \begin{pmatrix} h^2/2 \\ h \end{pmatrix} = \frac{1}{20.9} \begin{pmatrix} 0.005 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0{,}000239 \\ 0{,}004784 \end{pmatrix},$$

$$with \; h = dt = 0.1 \; and \; m_g = 20.9$$

As we can see the values for A and B are approximately the same.

Written by Dalim Wahby and Philipp Ahrendt - 07/09/2023

Question 3:



The system has two poles and no zeros. The poles are both located at s=0.

We expected no zeros and two poles. We expect the poles to be at 1, due to the following relationship: $\lambda = e^{\lambda_c} = e^0 = 1,\ \ with\ \lambda_c = 0$.
Our intuition was not entirely correct. The poles were at 1, as expected. However, there is one zero at about -0.96. This is due to the sample-and-hold operation.

Question 4: See code in file controller.py in function control_law().
We bound the control input signal, so that $|u_{\max}| = 0.85$. We chose one pole to be closer to the origin, to achieve a faster response. We chose to put the second pole close to the unit circle, to get rid of the overshoot.
The poles $p_1 = 0.6$ and $p_2 = 0.96$, to achieve convergence to the reference position after about 10 seconds.

Question 5:

With the pole placement from Q4, we get a steady-state error of approximately $\epsilon_{ss} = 0.01$, due to the disturbance. After implementing the integral action, we drive the steady-state error down to $\epsilon_{ss} = 0.0002$.