Model Predictive Control - EL2700

Assignment 1: State Feedback Control Design

Group 7: Christopher Biel, Davide Torrini

1 Q1

To answer this question, the matrices A_c and B_c given by

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad B_c = \begin{bmatrix} 0 \\ \frac{1}{m_G} \end{bmatrix}$$

were implemented in the function `one_axis_ground_dynamics`.

2 Q2

The mathematical steps to analitycally discretize the model, are described in the following. The matrices for the discretized system can be retrieved according to

$$A_d = e^{A_c h} \ , \ B_d = \int_{s=0}^{h} e^{A_c s} B_c ds \tag{1}$$

Moreover, the matrix exponential of $M \in \mathbb{R}^{n \times n}$ is defined by the power series

$$e^M = I + M + \frac{1}{2!}M^2 + \frac{1}{3!}M^3 + \cdots \tag{2}$$

so, it is possible to write

$$A_d = e^{A_c h} = I + hA_c + \frac{h^2}{2!}{A_c}^2 + \frac{h^3}{3!}{A_c}^3 + \cdots \tag{3}$$

Since ${A_c}^k = 0 \ \forall \ k \geq 2$, than eq.3 reduces to

$$A_d = e^{A_c h} = I + hA_c = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \tag{4}$$

The same results can be achieved by means of the *Laplace Transform*, indeed:

$$(sI - A_c)^{-1} = \begin{bmatrix} s & -1 \\ 0 & s \end{bmatrix}^{-1} = \frac{1}{s^2} \begin{bmatrix} s & 1 \\ 0 & s \end{bmatrix} = \begin{bmatrix} 1/s & 1/s^2 \\ 0 & 1/s \end{bmatrix} \tag{5}$$

$$A_d = e^{A_c h} = \mathcal{L}^{-1}\left((sI - A_c)^{-1}\right) = \begin{bmatrix} \mathcal{L}^{-1}(1/s) & \mathcal{L}^{-1}\left(1/s^2\right) \\ 0 & \mathcal{L}^{-1}(1/s) \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix} \tag{6}$$

For what concern B_d, according to eq.1

$$B_d = \int_{s=0}^{h} \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1/m_G \end{bmatrix} ds = \int_{s=0}^{h} \begin{bmatrix} s/m_G \\ 1/m_G \end{bmatrix} ds = \begin{bmatrix} h^2/2\, m_G \\ h/m_G \end{bmatrix} = \begin{bmatrix} 0.00023923445 \\ 0.004784689 \end{bmatrix} \tag{7}$$

From `casadi_c2d` the following matrices were obtained:

$$A_d = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \qquad B_d = \begin{bmatrix} 0.00024417 \\ 0.00478469 \end{bmatrix}$$

As can be witnessed, matrix A_d is the same, while matrix B_d is slightly different. This could be due to computational approximation.

3 Q3

Since

$$\ddot{p_x}(t)\, m_G = u(t) \tag{8}$$

the *Laplace Transform* can be exploited to get

$$s^2 P_x(s) m_G = U(s) \tag{9}$$

$$P(s) = \frac{U(s)}{s^2\, m_G} \tag{10}$$

$$P(s) = \frac{F_x}{s^2\, m_G} \tag{11}$$

This is the *TF* of the continuous-time system from the control input $u(t)$ to the output $y(t)$. The system does not have any zeros, but two poles in the origin. This can also be retrieved by the plot of the poles and zeros of the continuous system presented in fig.1

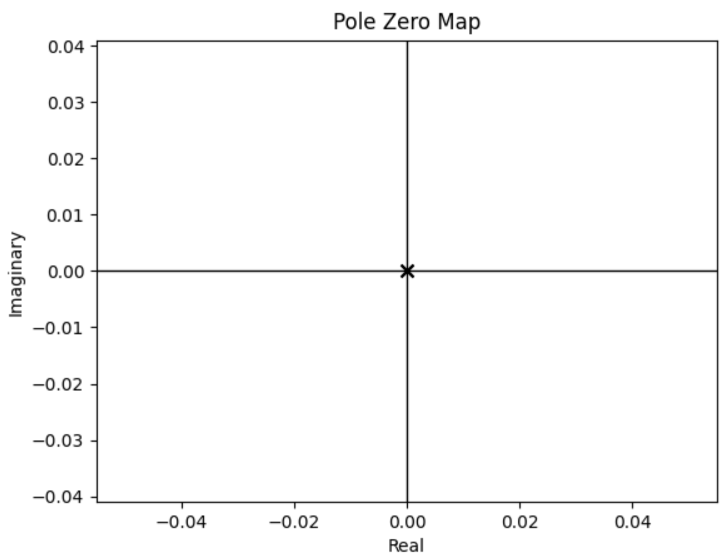

Figure 1: Pole and Zero of the continuous time system

A good pole location for the continuous-time system is showed in 2.

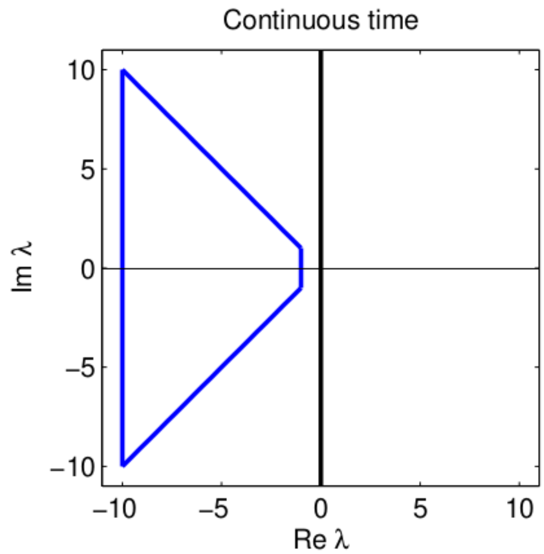

Figure 2: Good pole location for continuous-time systems

It is possible to conclude that the continuous-time system does not have convenient pole locations.

The poles and zeros of the discrete-time model can be witnessed in fig.3. As expected the discrete-time model also has two poles, their location can be computed from $exp(0) = 1$. Furthermore the discrete system has a zero, which is a result from the sampling in the continuous to discrete transfer. We did not expect that in the beginning, but after some consideration is does make sense.

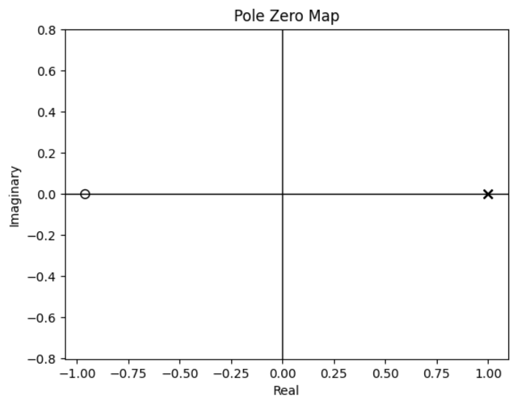

Figure 3: Pole and Zero of the discrete time system

A good pole location for the discrete-time system is showed in fig.4. In this case, the poles are

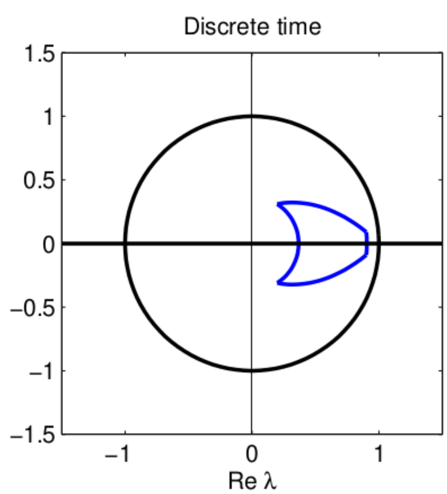

Figure 4: Good pole location for continuous-time systems

located on the boundary of the unit circle, resulting in a non-optimal position for a discrete-time. Indeed, they should be located inside the unit circle.

4 Q4

By taking into account what fig.5 displays i.e.:

- increasing the imaginary part of the chosen pole lowers the damping
- increasing the absolute value of the pole increases the frequency

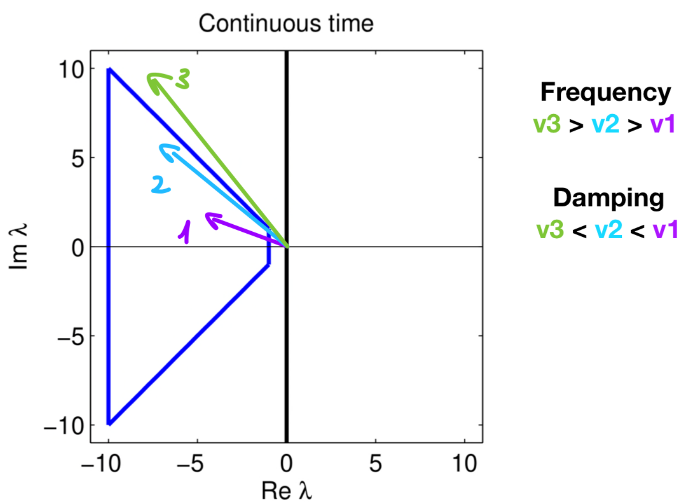

Figure 5: Pole effect on the system

we designed our poles in the continous time system and trasnferred them to discrete with $p_d = exp(p_c)$. Since overshoot is to be avoided at any means (so as not to crash into the wall), the damping should be $D > 1$ which results in poles without an imaginary value. To still keep the damping low the poles are placed as close to each other as possible. We experimented with the absolute value to achieve the fastest possible result without exceeding the maximum control input allowed. The two poles in continuous time that satisfy the requirements are -0.02 and -0.02005. In this way the system is able to:

- achieve 90% of the reference within 25 s
- keep the maximum control input below 0.85 N.

This is shown in the resulting plots displayed in fig.6.

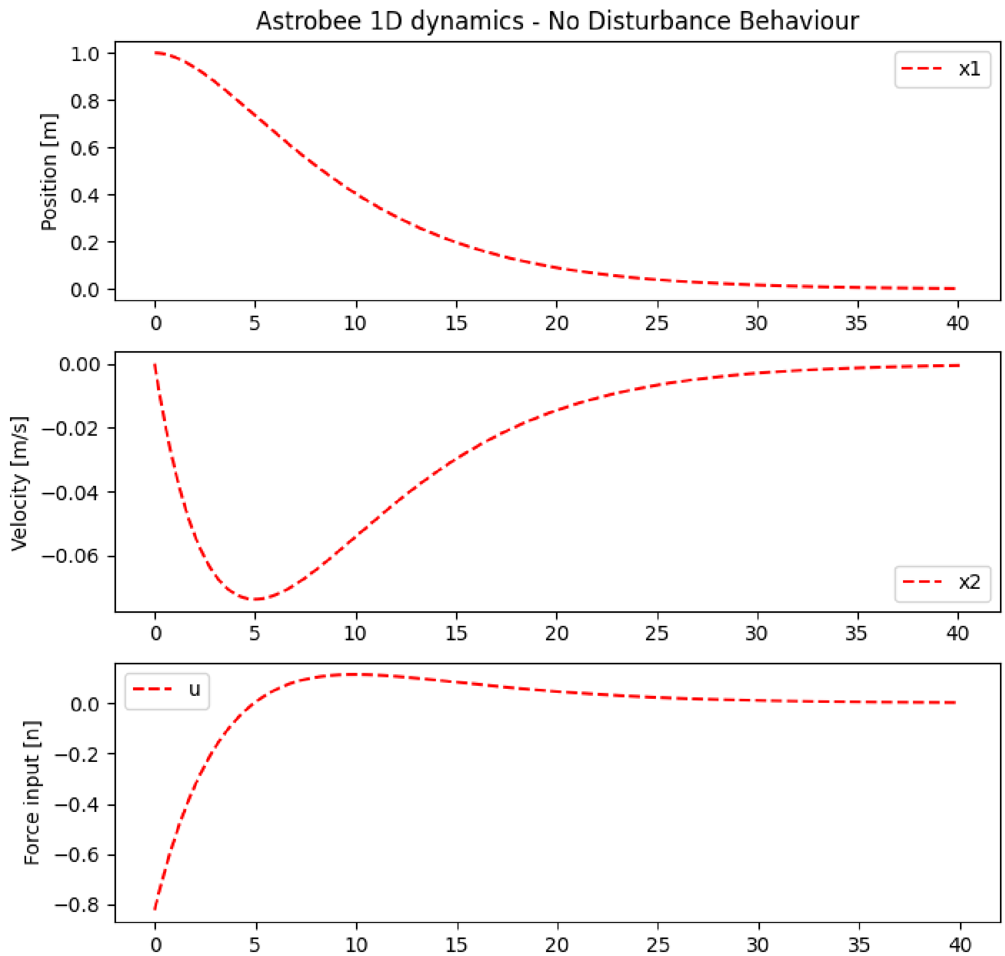

Figure 6: Dynamics of the astrobee with a simple P-Controller and no disturbance. More than 90% of the target state are reached within 25 s and a maximum control input below 0.85 N.

5 Q5

The implementation of the integral part allowed the system to respond to the additional disturbances. We experimented with different values for the integrator gain K_i to minimize the possible overshoot. For long simulation periods there will always be overshoot with an integrator, since it requires an overshoot to decrease the integrator error. We achieved a fast response within 30 s and no overshoot (in our 40 s simulation period) with a controller gain of $K_i = 0.03$. The result of that implementation is shown in fig. 7.

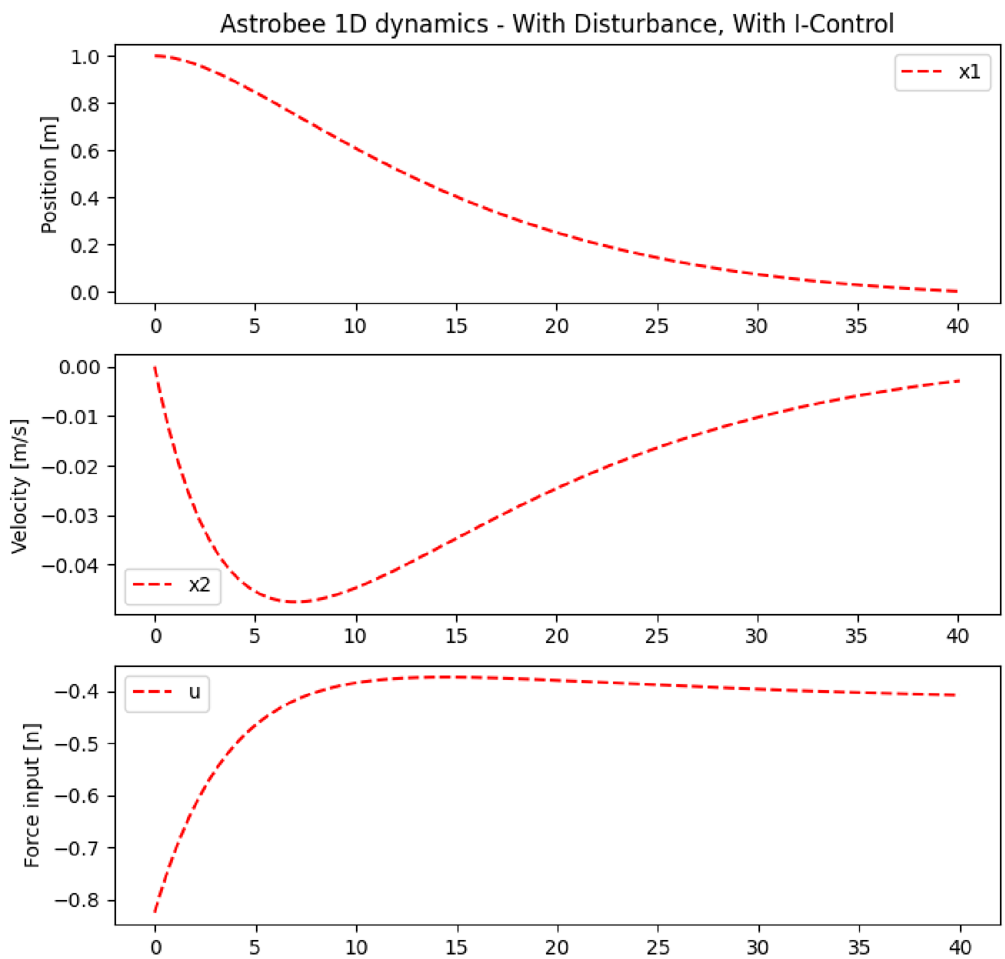

Figure 7: Dynamics of the astrobee with a PI-Controller and actuator disturbance. The position shows no overshoot (yet - it would for longer simulation times). More than 90% of the target state are reached within 30 s and a maximum control input below 0.85 N.