Assignment 2

Konrad Dittrich, Simon Spång

Task 1:

The answer can be found in the class astrobee under the method cartesian_ground_dynamics. It is implemented between the rows 68 to 74. See figure 1.

```
64        Ac = ca.DM.zeros(self.n, self.n)
65        Bc = ca.DM.zeros(self.n, self.m)
66
67        # TODO: Fill the matrices Ac and Bc according to the model in (1)
68        Ac[0,2] = 1
69        Ac[1,3] = 1
70        Ac[4,5] = 1
71
72        Bc[2,0] = 1/self.mass
73        Bc[3,1] = 1/self.mass
74        Bc[5,2] = 1/self.inertia
75
76        self.Ac = np.asarray(Ac)
77        self.Bc = np.asarray(Bc)
78
79        return self.Ac, self.Bc
```

Figure 1 - A and B matrices

Task 2:

See the code in the file task2.py for the use of the methods casadi_c2d and set_discrete_dynamics.

Task 3:

The rendezvous constraints are declared as in figure 2.

```
con_ineq.append(x_ref[0:2] - x_t[0:2])
con_ineq_ub.append(self.pos_tol)
con_ineq_lb.append(-self.pos_tol)

con_ineq.append(x_ref[2] - x_t[4])
con_ineq_ub.append(self.att_tol)
con_ineq_lb.append(-self.att_tol)
```

Figure 2 - Rendezvous constraints

Where $\overline{x_{ref}} = \begin{bmatrix} P_x \\ P_y \\ \theta \end{bmatrix}$ and $\overline{x_t} = \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \\ \theta \\ \omega \end{bmatrix}$

Task 4:

The simulation results in the trajectory plots, see figure 3. The goal state x_{ref} is reached after ca. $t = 25\ sec$. The system behaves as expected and specified.

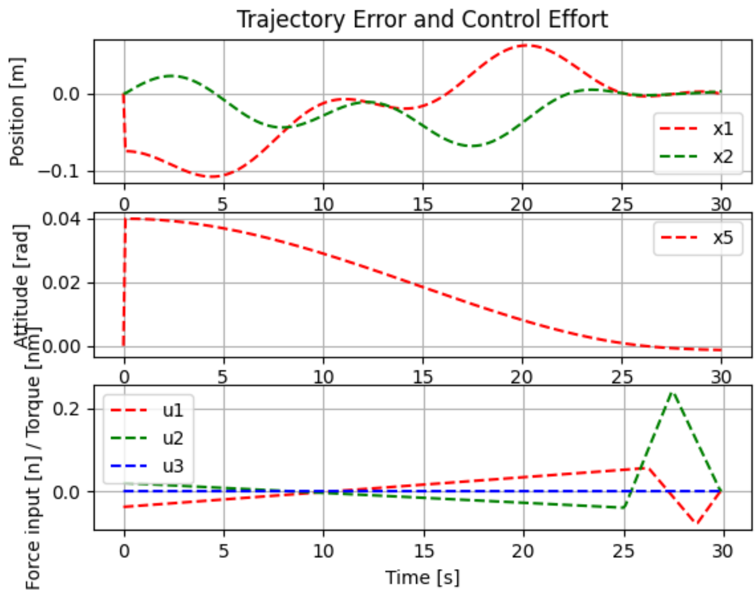

Figure 3: Trajectory in x and y-direction

Task 5:

Since the thruster u_2 is broken the Force input is heavily disturbed. This leads to deviations in the control of the position. As a result, the reference state is not reached during simulation time. The root cause for this behavior is that the control is not able to compensate the deviations of the input signal of u_2 since the design of the control is not robust enough.

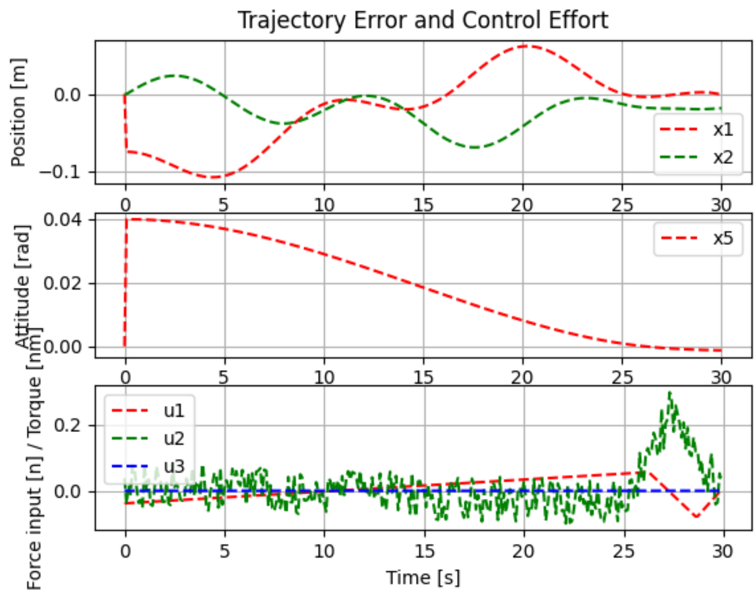

Figure 4: Trajectories with broken thruster u2

Task 6:

To fix the deviation in y caused by the noisy u_2-signal we could add a feedback-loop, since this additional policy can reduce the effects of unforeseen disturbances and make the control more robust, see Lecture Chapter 3.1.

Task 7:

The model is to be extended by the third dimension by expanding the *cartesiana_3d_dynamics*-function. The trajectories are plotted in figure 5. A new controller *ctrl_wz* is used which needed 3.73 sec to obtain a solution. The 2d-simulation took 0.52 secs.

By comparing the simulation in 2d- and 3d-setup a reasoning for the increase of computation time is found, see table 1. By adding a third dimension, which means two more state variables and one more input variable, the number of variables through the whole calculation is increased 1.78 times, the constraints raise by factor 1.33 and all other matrices and calculation entries are more than doubled. As a result, the computation time increases by factor 7.17.

Table 1: comparison of 2d- and 3d-simulation setup

	2d-simulation	3d-simulation	Factor of increase
Number of variables	2706	4808	1.78
Number of constraints	2853	3804	1.33
Number of nonzeros in H	2700	6400	2.37
Number of nonzeros in A	19053	42204	2.22
Number of nonzeros in KKT	45465	97820	2.15
Number of nonzeros in QR(V)	43467	97932	2.25
Number of nonzeros in QR(R)	84111	180124	2.14

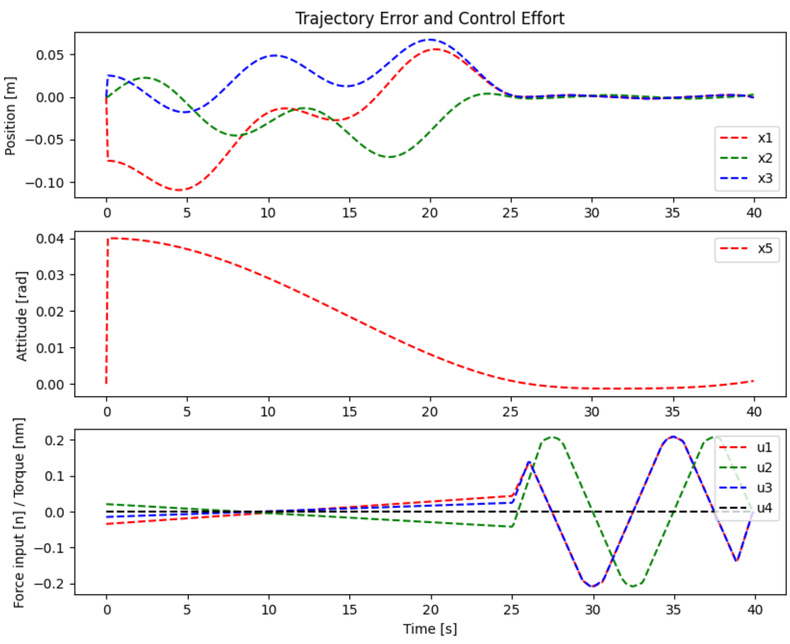

Figure 5: Trajectory and control for the 3D-system