MPC: Serie 4

Groupe 20
Camilla Carta
Etienne Clement

April 19, 2018

Exercise 1

For this exercise, the following system was considered:

$$x^+ = \begin{bmatrix} 0.9752 & 1.4544 \\ -0.0327 & 0.9315 \end{bmatrix} x + \begin{bmatrix} 0.0248 \\ 0.0327 \end{bmatrix} u \tag{1}$$

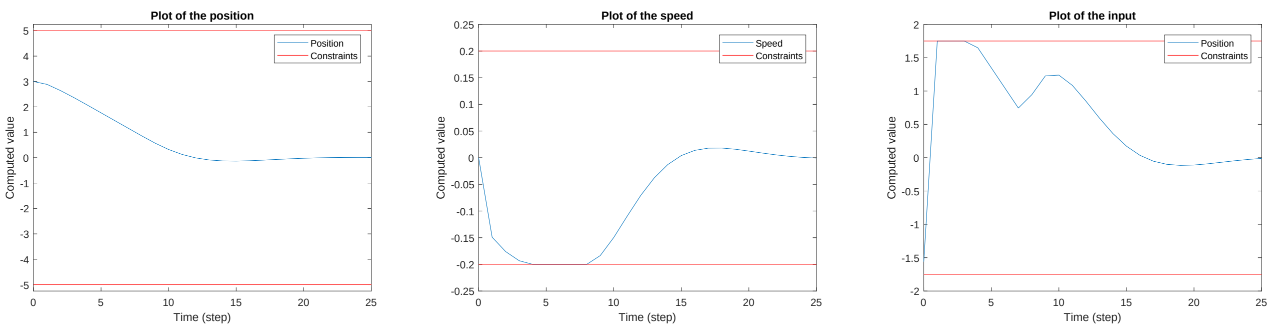
with constraints

$$\mathbb{X} = \{x \mid |x_1| \leq 5, |x_2| \leq 0.2\} \qquad \mathbb{U} = \{u \mid |u| \leq 1.75\} \tag{2}$$

The goal was to implement an MPC controller with an horizon of $N = 10$ and a stage cost given by $I(x, u) = 10x^T x + u^T u$.

As such, the implementation found in the Matlab code `ex4_1.m` gives the results that can be observed in Figure 1.

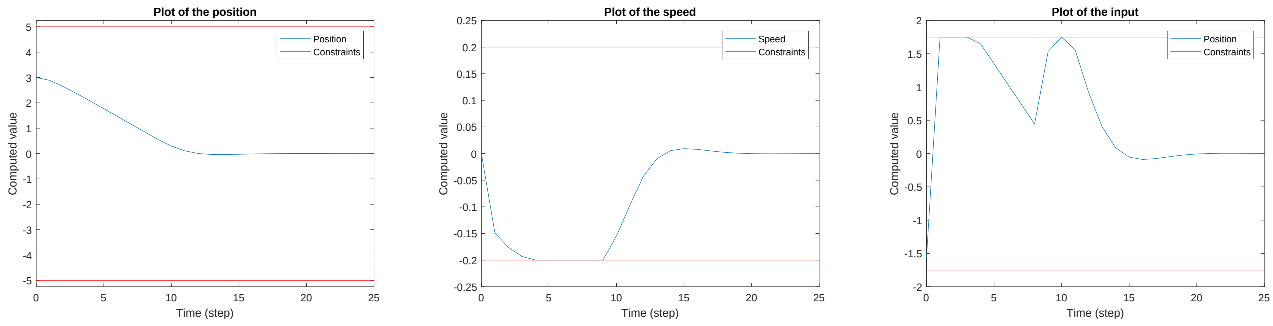
In all results beneath it can be noticed that the constraints, plotted in red, are never overstepped. This confirms that they were taken into account by the system.



(a) Position (b) Velocity (c) Input

Figure 1: Plot of the states for $Q = 10$ and $R = 1$

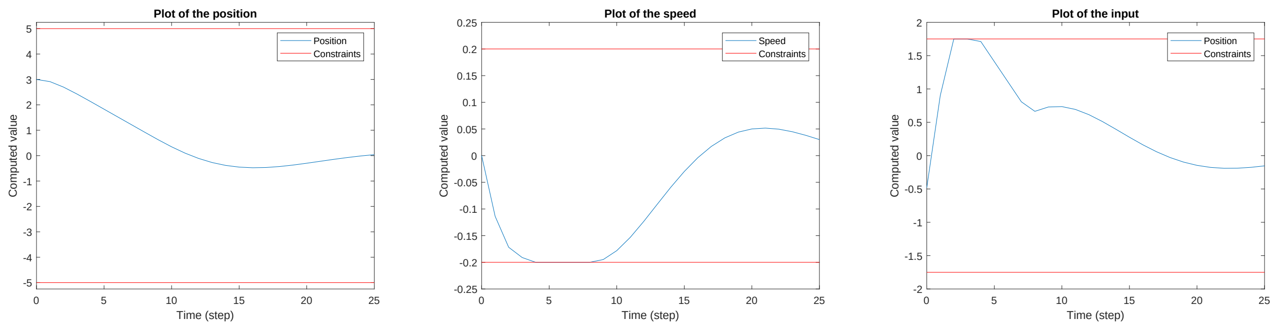
As explained below, the implementation using the YALMIP toolbox did not work for the cost function given above. For this reason, the results for the same cost function used in Exercise 2 are shown in Figure 2. It can be observed that, while the plots are quite similar to the previous, the different cost function gives more aggressive results.



(a) Position (b) Velocity (c) Input

Figure 2: Plot of the states for $Q = 10$ and $R = 1$

Finally, the R parameter was changed to $R = 10$ to study its effect on the results. It can be noticed in Figure 3 that the curves are smoother, although the system itself is less stable and does not achieve the steady-state as quickly. This is expected as a large R tries to minimize variation of the input.



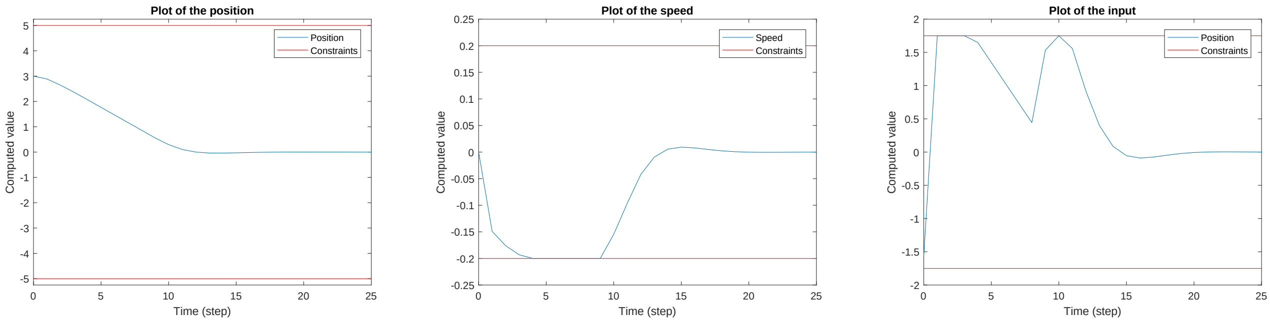
(a) Position (b) Velocity (c) Input

Figure 3: Plot of the states for $Q = 10$ and $R = 10$

Exercise 2

In this exercise the YALMIP toolbox was used to optimize the input. The flag *isfeasible* was set to 0 meaning that the problem could not be solved through this method. However the first steps were still solved and are shown in the figures below.

Some problems were encountered while trying to obtain these results: at at first the number of possible state steps was 5 at most, after which the computation gave an input of value Nan, which invalidated the following calculations. After discussing with the assistants it was found that YALMIP was not working properly with the default solver, as such it would not find any possible solution for $Q = 10$. Changing the solver to the `quadprog` option did not solve the problem. A solution was then found by changing the parameter to $Q = 100$. In Figure 4 the results can be observed. In particular, it can be seen that the results are very similar to the ones obtained in the exercise above.



(a) Position (b) Velocity (c) Input

Figure 4: Plot of the results found using the YALMIP toolbox