Assignment 4

Konrad Dittrich and Simon Spång

**Q1:**

En bild som visar skärmbild, text, skärm, Rektangel

Automatiskt genererad beskrivning

Figure - N = 5 steps

En bild som visar skärmbild, text, skärm, lila

Automatiskt genererad beskrivning

Figure - N = 10 steps

En bild som visar skärmbild, text, Färggrann, skärm

Automatiskt genererad beskrivning

Figure 3 - N = 20 steps

As seen in figure 1-3, the distance from the origin in x-direction increases with the increasing control horizon. The y-direction is constrained between -0.1 and 0.1. In general, an increase of horizon steps *N* increases the initial set of feasible solutions.

**Q2:** With an increased boundary of the controller the set of feasible solutions increases. Once again, only in x-direction. When increasing the boundaries of u, the invariance set increases proportionally in x-direction until the limit of ±1.2 is reached.

En bild som visar text, skärmbild, skärm, Rektangel

Automatiskt genererad beskrivning

Figure 4 - Varying of control constraints

**Q3:**

1. The state constraints are set by appending upper and lower bound as inequality constraints separately for every timestep t. Regarding the upper bound, xub, it is set as:

And the lower bound is set as:

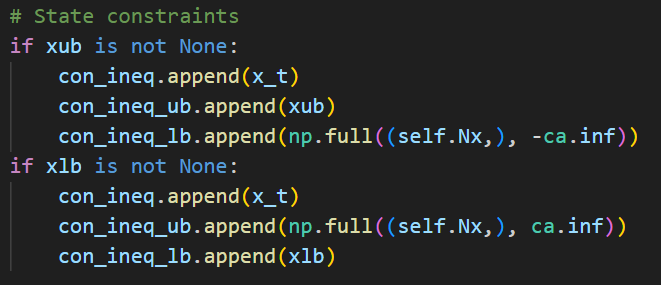


Figure 5 - State constraints

1. The object function is formulated by summing the running cost over every timestep and adding the terminal cost once at the end. The entries of the cost function are the difference between x\_t and x0\_ref, the weighting matrices Q and R as well as the control signal u\_t. Regarding the terminal cost the difference between the last state and the reference state is penalized by matrix P.

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning

Figure 6 - Object function

1. The terminal constraints are included by adding inequality constraints for the final state x\_N. The inequality constraints upper bound is set by using the offsets of the polytope and the lower is set to minus infinity. The following inequality holds:

En bild som visar text, skärmbild, Teckensnitt, programvara

Automatiskt genererad beskrivning

Figure - Terminal constraints

1. The variable param\_s concatenates vertically the vectors, x0, x0\_ref and u0 in form of a single input vector for the casadi solver. The three input vectors are the start values. The param\_s is handed over to the solver as a part of a dictionary to formulate the nonlinear program (nlp). See figure 9.

En bild som visar text, Teckensnitt, skärmbild

Automatiskt genererad beskrivning

Figure 8 - Declaration of param\_s

En bild som visar text, skärmbild, Teckensnitt

Automatiskt genererad beskrivning

Figure 9 – Dictionary for the solver

1. The variable x0 is used to receive the first control input u0 to control the system. See figure 10.

En bild som visar text, Teckensnitt, skärmbild, Grafik

Automatiskt genererad beskrivning

Figure 10 - Usage of x0

**Q4:**

The values for energy use, as well as the position and attitude integral errors obtained when running the simulation with and without expanding the boundaries of can be found in table 1.

Table 1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Percentual change |
| Energy use | 156.2 | 160.5 | +2.75% |
| Position integral error | 3.94 | 3.55 | -9.9% |
| Attitude integral error | 0.9 | 0.88 | -2.2% |

From the table it can be seen that an increase of the boundaries of with a factor of 3 results in an increase of energy usage while the integral errors decrease. This is reasonable since a larger set of control inputs has the possibility to decrease the errors further on a limited horizon compared to a small set. But it comes with the cost of being more energy consuming. The percentual changes show that error reduction is higher than the increase of used energy.

En bild som visar text, diagram, linje, Parallell

Automatiskt genererad beskrivning

Figure 11 - Simulation with u\_lim

En bild som visar text, diagram, linje, Parallell

Automatiskt genererad beskrivning

Figure 12 - Simulation with 3\*u\_lim

From the simulations it can also be seen that the position is reached slightly faster when the boundary of the input signal is increased by a factor of 3.

**Q5:**

When setting the terminal set to the problem becomes infeasible. By comparing the invariant set obtained by (figure 13) with the invariant set in Q1 (figure 14), it has decreased significantly which makes the initial state appear outside the set. As a result, there is no solution for the MPC problem.

En bild som visar text, skärmbild, skärm, programvara

Automatiskt genererad beskrivning

Figure 13 - Invariant set with terminal set as zero

En bild som visar skärmbild, text, skärm, lila

Automatiskt genererad beskrivning

Figure 14 - invariant set from Q1

**Q6:** The problem becomes feasible again when changing the horizon length from to . But the computational time is increased. With the computational time is approximately 18 seconds. This can be compared to the computational time of which is 44 seconds.

Regarding the time to solve the MPC problem it took around 0.04 to 0.11 seconds for to solve it. For it took around 0.12 to 0.2 seconds to solve it which is an increase of approximately factor 2.

**Q7:**

* **By multiplying R with a factor 10**, the energy used is decreased to 100.61. This decrease is reasonable since the control penalty is increased and therefore the optimal solution is found with smaller control inputs. The integral error of the position is 5.82 and for the attitude 0.91, which is an increase in both cases. As a drawback the astrobee reaches its final position after approximately 6 seconds compared to 4.5 seconds for the standard case. It also uses a smaller force input with a maximum of 0.13 N which can be compared to 0.3 N for the standard case.
* **By adding 100 to the velocity components of Q,** the energy used is reduced to 32 with the drawback of increased positional integral error to 18 and attitude integral error of 7. The astrobee will not reach its final position within simulation time of 20 seconds. The maximum velocity is decreased . This can be compared to the standard case when the maximum of approximately .
* **By adding 100 to the positional components of Q,** the system tries to reach the desired states faster. As a result, we overshoot the goal states, e.g., by in . The energy consumption is .
* **By increasing all elements of Q with 100,** we increase the penalty on every state. Hence, the states in total are penalized more than the control input. Since the velocity is penalized higher than before, the energy consumption is decreased to . The overshot of the position states is decreased, e.g., to of .
* **Conclusion:** Tuning the MPC problem is quite like tuning an LQR-problem in term of tuning rules and intuition. For example, by increasing the penalty on the position and attitude states the controller reaches the desired states faster with a higher tendency to overshoot. By penalizing the control inputs the energy consumption can be highly decreased at the cost of getting a slower system response.