Lightsoccer

Tijmen Wintjes

May 24, 2017

1 Formulation of the optimization problem

This report treats the problem sketched the figure. A soccer player is approaching a the goal whilst the keeper tries to protect as much of the goal as possible.

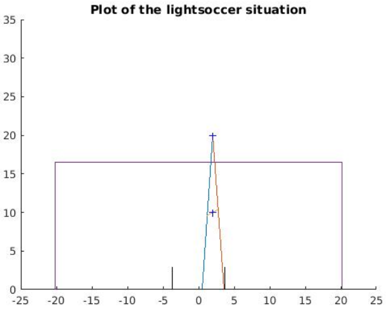

Figure 1: Light soccer situation with keepe rat (2,10), attacker at (2,20) and the keeper being at an angle of $\pi/2$

Given the position of the attacker we can calculate what the optimal

Given that we know where the attacker will be, we should calculate the optimal position of the keeper. To find the optimum placement of the keeper we juse the fuction:

$$J = x_l^2 + x_r^2$$

Where x_l and x_r are respectively the left and right uncovered parts of the goal. In the picture above $x_l = 4.17$ and $x_r = 0.02$

As small uncovered edges in the corners will lead to small costs, it is likely that the keeper will not fully cover a corner, but will rather choose to cover a bigger part of the other corner. In other words the keeper will stay more in the middle. This is okay, as it is hard for soccer player to aim precisely.

To get an idea of the costfunction a 3D plot was made. In the figure below a topview is presented. The white part in the picture is close to the attacker, too close such that a division by zero occurs in the calculation. This is okay as these values correspond with unfeasible points.

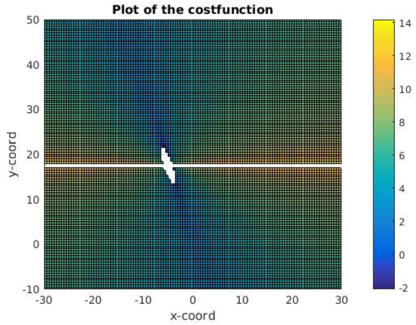

Figure 2: A topview of the costfunction, theta = 0, the colors give the levels of the costfunction.

The optimization problem to be solved is the following. W $max x_k = 19.66$

$$\min_{x_k,y_k,\theta} J(x_k, y_k, \theta)$$
$$s.t. - 19.66 \leq x_k \leq 19.66$$
$$0 \leq y_k \leq 16$$
$$- \pi/2 \leq \theta \leq \pi/2$$

2 Dynamic keeper model

The symbolic expression of the discrete matrices Ad, Bd, Cd and Dd are displayed in the figure below.

$$Ad = \begin{pmatrix} 1 & 0 & 0 & \mathrm{Ts} & 0 & 0 \\ 0 & 1 & 0 & 0 & \mathrm{Ts} & 0 \\ 0 & 0 & 1 & 0 & 0 & \mathrm{Ts} \\ 0 & 0 & 0 & 1 - \frac{\mathrm{Ts}\,\rho}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 - \frac{\mathrm{Ts}\,\rho}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 - \frac{\mathrm{Ts}}{100\,I} \end{pmatrix} \quad Bd = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\mathrm{Ts}}{m} & 0 & 0 \\ 0 & \frac{\mathrm{Ts}}{m} & 0 \\ 0 & 0 & \frac{\mathrm{Ts}}{I} \end{pmatrix}$$

$$Cd = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \quad Dd = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Filling in the numerical values for ρ m and I we get the following matrices.

$$Ad = \begin{pmatrix} 1 & 0 & 0 & \frac{3}{20} & 0 & 0 \\ 0 & 1 & 0 & 0 & \frac{3}{20} & 0 \\ 0 & 0 & 1 & 0 & 0 & \frac{3}{20} \\ 0 & 0 & 0 & \frac{67}{70} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{67}{70} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2797}{2800} \end{pmatrix} \qquad Bd = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{3}{1400} & 0 & 0 \\ 0 & \frac{3}{1400} & 0 \\ 0 & 0 & \frac{3}{28} \end{pmatrix}$$

$$Cd = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad Dd = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The eigenvalues of the matrix Ad are all within the unitcircle, making the discrete-time model obtained with Euler's rule stable. The system is controllable, as the rank of the controllability matrix is equal to the rank of the matrix Ad. The model is also observable as the rank of the observability matrix is equal to the rank of Ad. As the observability matrix and controllability matrix are of full rank the system is stabilizable and detectable. There are no transmission zeros in the system and it is of minimal form.

The two figures below show different discretizations of the system. The step responses are all very similair to the stepresponse of the continuous system. Computing the reference states with different discretizations also gave similair results. As the discretized model using the euler rule was already analyzed, we will continue using that model.

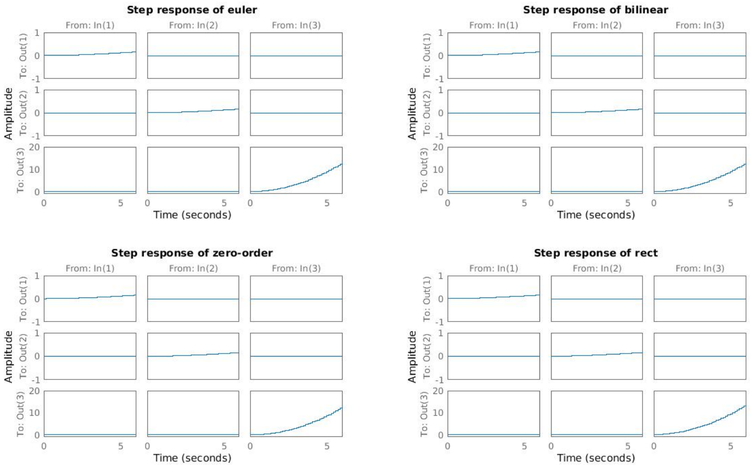

Figure 3: Stepresponses of four discreted versions of the original model

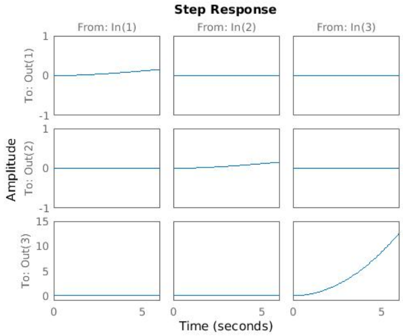

Figure 4: Stepresponse of the continuous model

3 Setpoints

The optimization problem exactly retrieves the input required to reach the reference states. The problem is however ill-conditioned, the matrix H used in the quadratic program has a condition number of $2.7006e + 12$. Calculating the input with this matrix leads to the input sequence in the figure below on the left. We would like a smoother input and therefore we add a regularization term $\mu \cdot I$ with μ a small number to H, $\mu = 0.000001$ already gives a much better result. See the input on the right.

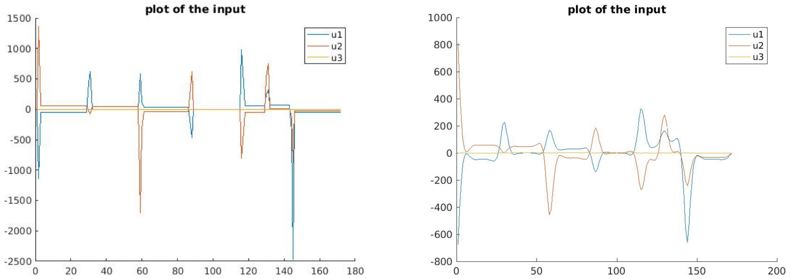

Figure 5: Optimal input calculated without and with a regularization term. For the input on the right the regularization term $\mu = 0.000001$

To calculate the reference state we simulate the system with our reference input. The figure below presents both the original reference path of the keeper and the newly calculated references.

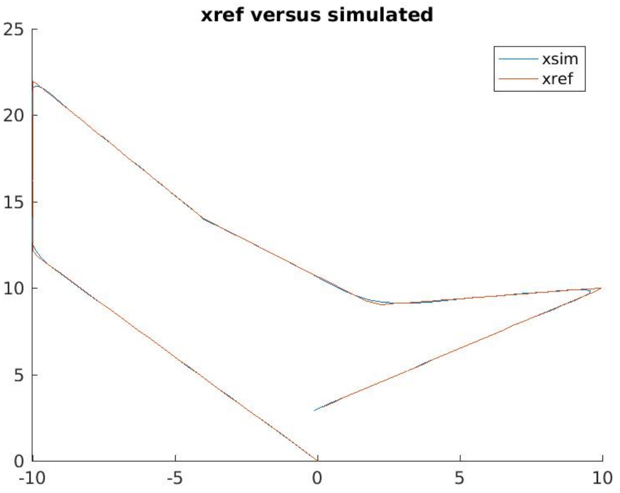

Figure 6: Plot of original reference path and the newly calculated references

4 LQR control

We now are ready to control our keeper with LQR control. Two controllers were created according to the specifications, denoted K_{lqr1} and K_{lqr2}.

$$K_{lqr1} = \begin{pmatrix} 86.6 & 0 & 0 & 117.0 & 0 & 0 \\ 0 & 86.6 & 0 & 0 & 117.0 & 0 \\ 0 & 0 & 8.6 & 0 & 0 & 9.26 \end{pmatrix}$$

$$K_{lqr2} = \begin{pmatrix} 89.9 & 0 & 0 & 89.4 & 0 & 0 \\ 0 & 89.9 & 0 & 0 & 89.4 & 0 \\ 0 & 0 & 38.9 & 0 & 0 & 7.92 \end{pmatrix}$$

In the figures below we subsequently plot the trajectories, the input signals and the states of both the controllers.

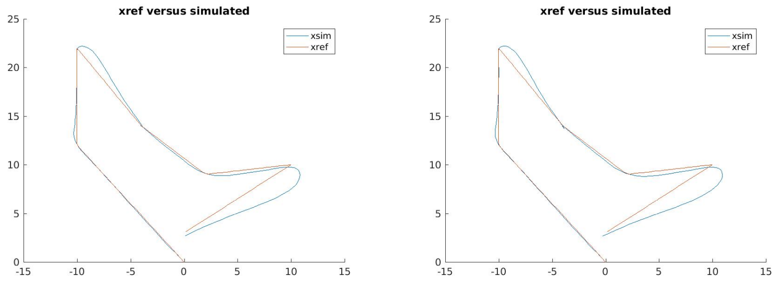

Figure 7: On the left the simulated trajectory with K_{lqr1} and on the right K_{lqr2}

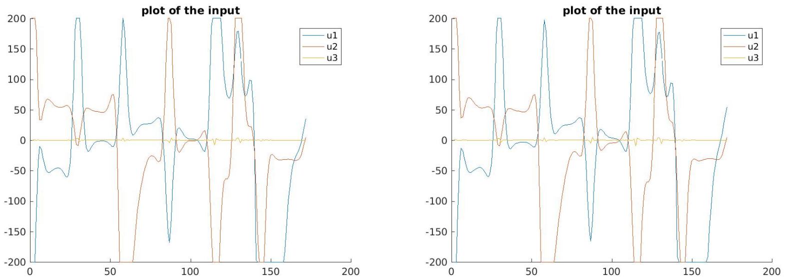

Figure 8: On the left the input with K_{lqr1} and on the right K_{lqr2}

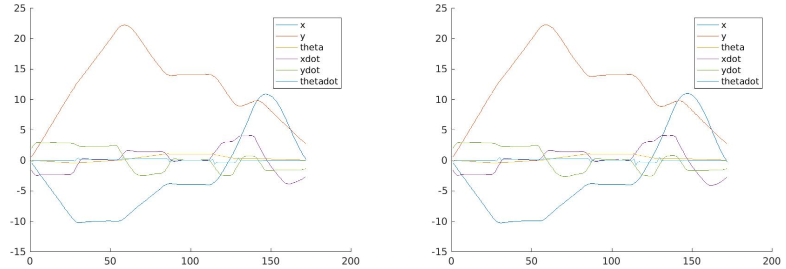

Figure 9: On the left the sates of the simulated trajectory with K_{lqr1} and on the right K_{lqr2}

The cost of the simulation shows that the cost of the second controller is lower, namely 603.06 versus 705.87 respectively. The trajectories however look very similair. The fact that the plots are similair can be expected since the entries that were removed in the second controller mostly corresponds with disturbance rejection, and in our simulation there are no disturbances.

5 MPC

Another approach we can take is that of the modelpredictive controller. Every timestep we will look ahead and calculate the subjectively "optimal" input to achieve the path closest to our reference states.

5.1 Without state constraints

Firstly we ignore the state constraints. The trajectory using a prediction horizon of 12 is shown below.

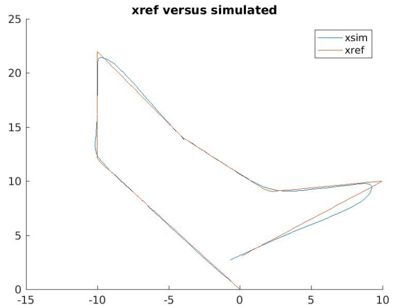

Figure 10: Simulate trajectory and the reference trajectory

We also plot the control actions and the state variables, the simulation cost for a horizon of 12 is equal to 311.47. This is already much better as the second LQR controller defined in the previous section.

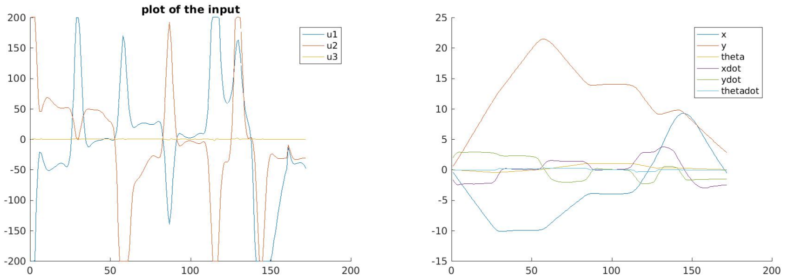

Figure 11: On the left control actions and on the right the states for an MPC controller with N = 12.

If we time how long the optimization problem takes a every iteration point we get the following graph. The periods where the computation time increases correspond with parts of the trajectory where the reference trajectory cannot be followed. This is as expected, because if it can be followed the solution is simply the reference input/output, if this is not possible another solution has to be find.

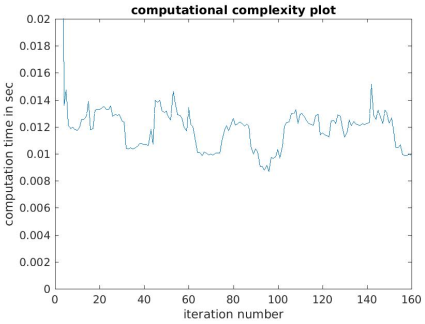

Figure 12: Plot of neccesary computation time for the optimazation for every iteration

We will do the same for a shorter and longer horizon. The found trajectories are the following.

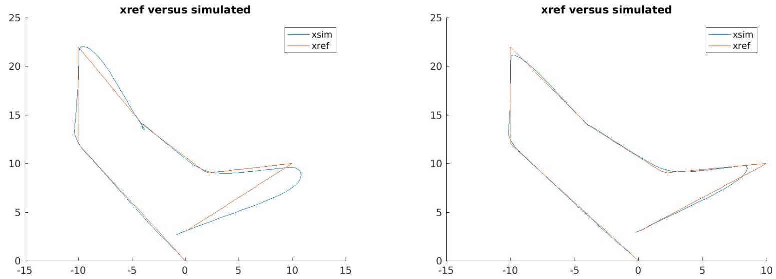

Figure 13: Trajectories with MPC for N=6 and N=24

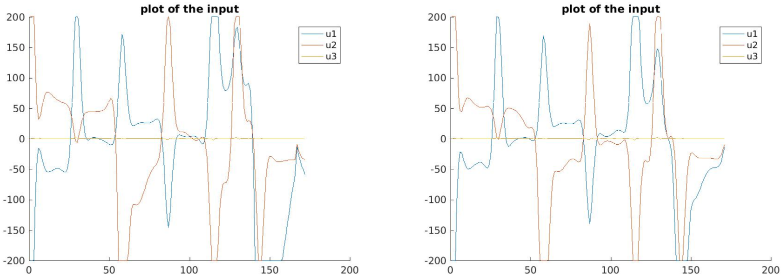

Figure 14: Inputs of the simulation for an MPC controller with on the left N=6 and right N=24

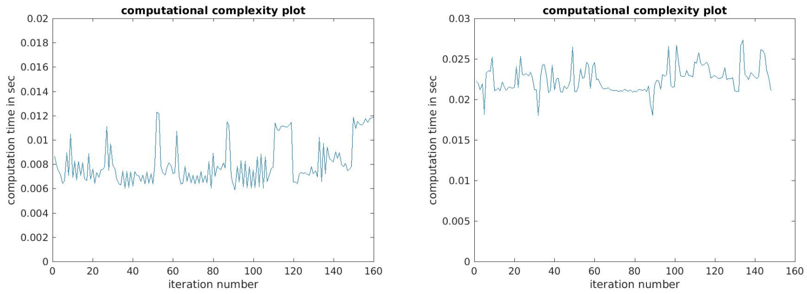

Figure 15: Computational time of the optimizations with on the left the timigns for an MPC controller with N=6 and on the rigth N=24

The cost of the trajectory with N=6 equals 569.72, the cost with N=24 equals 280.32.

5.2 With state constraints

Incorporating state constraints we get a different picture. For a horizon of order 12 we get the following trajectory, input and optimization timings. The cost of the simulation is 749.44.

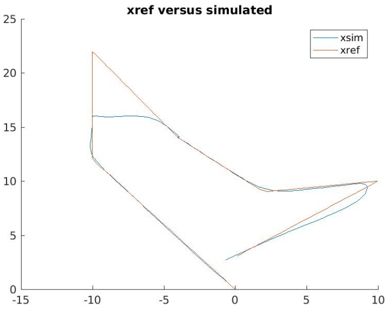

Figure 16: Plot of the trajectory for N=12 and state constraints

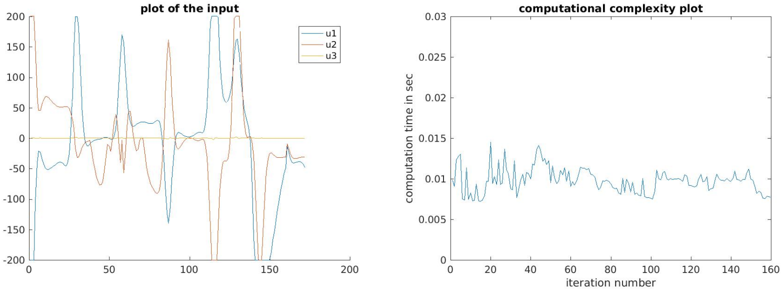

Figure 17: On the left the input of a state-constrainted mpc-control with N = 12, on the right the timings of the optimization problems at each iteration.

We repeat the experiments for a horizon of N=6 and N=24 and get the following results. We plot the trajectories, inputs and timings of the optimizations. Cost of the MPC-controller with N=6 is 987.66. Cost of the controller with N=24 is 723.73.

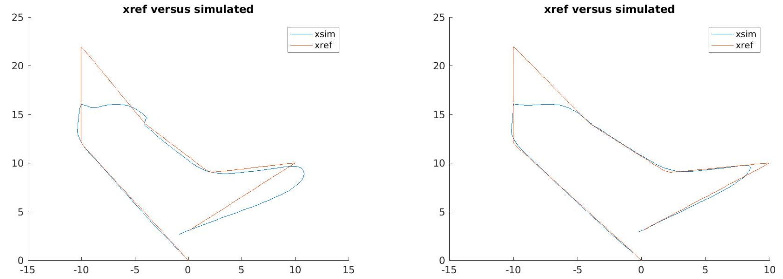

Figure 18: On the left the trajectory of a state-constrainted mpc-control with N = 6, on the right the same plot for N=24.

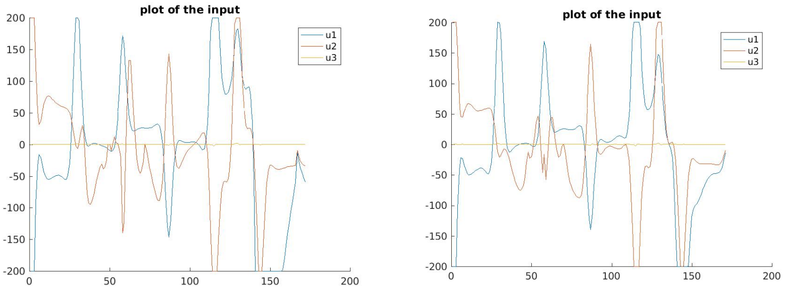

Figure 19: On the left the input of a state-constrainted mpc-control with N = 6, on the right the same plot for N=24.

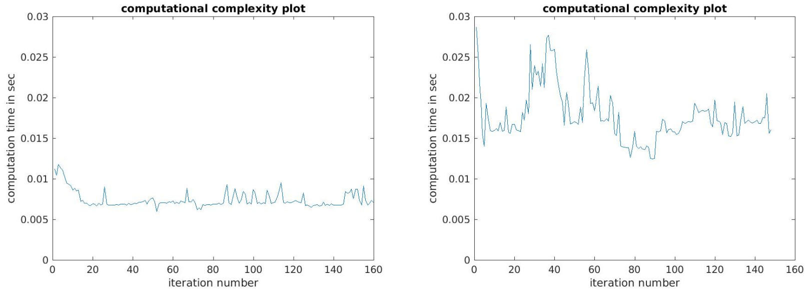

Figure 20: On the left the optimization timings of a state-constrainted mpc-control with N = 6, on the right the same plot for N=24.

No infeasibilities were observed, however, for a very short horizon this theoretically could happen. Out of curiosity we investigate and find that this indeed happens for a horizon of 2. The keeper's speed is too high to turn before going out of the penalty area and the controller does not notice it in time. Therefore the controller cannot find a input which adheres to the constraints and the optimization fails.

Increasing the maximal force we repeat the simulations. The cost of the trajectory with N=12 is 930.84. Choosing N = 6 we find a cost of 1034.6 and with N=24 we find a cost of 923.73.

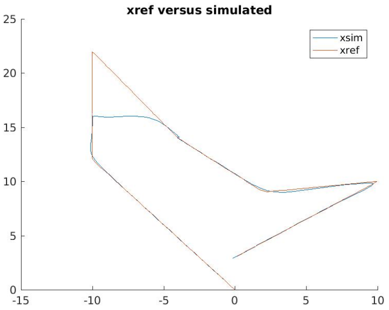

Figure 21: Plot of the trajectory for N=12 and relaxed maximum forces

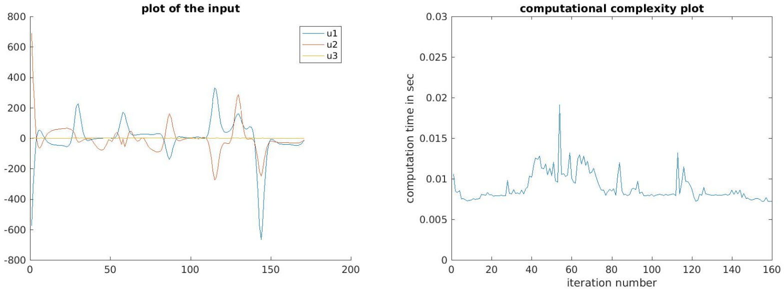

Figure 22: On the left the input of a state-constrainted mpc-control with N = 12, on the right the timings of the optimization problems at each iteration. The force constraints were relaxed to ten times their original value

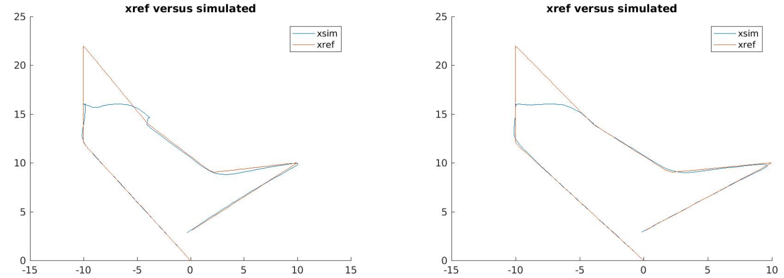

Figure 23: On the left the trajectory of a state-constrainted mpc-control with N = 6, on the right the same plot for N=24. The force constraints were relaxed to ten times their original value

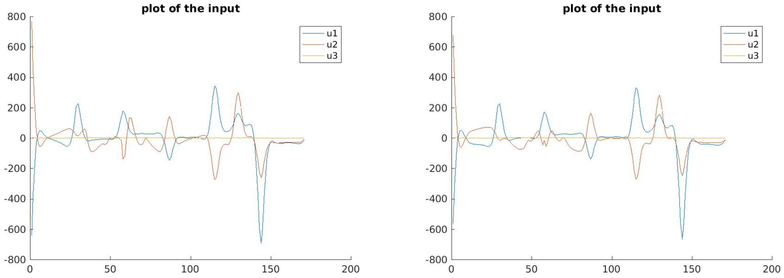

Figure 24: On the left the input of a state-constrainted mpc-control with N = 6, on the right the same plot for N=24. The force constraints were relaxed to ten times their original value

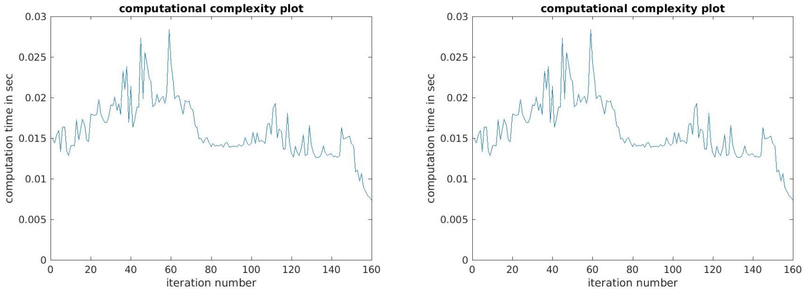

Figure 25: On the left the optimization timings of a state-constrainted mpc-control with N = 6, on the right the same plot for N=24. The force constraints were relaxed to ten times their original value