

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.

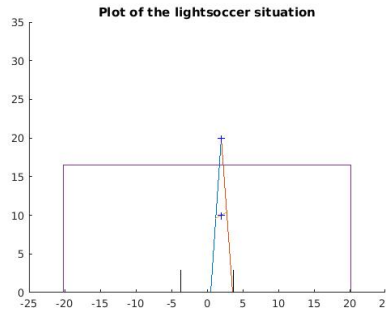


Figure 1: Light soccer situation with keepe rat (2,10), attacker at (2,20) and the keeper being at an angle of $\pi/2$

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.

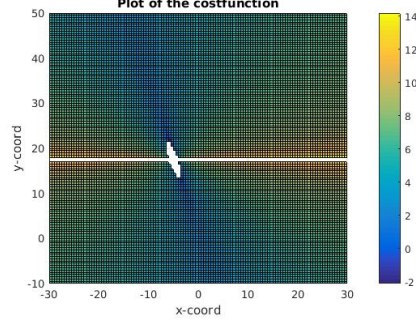


Figure 2: A topview of the costfunction, $\theta = 0$, the colors give the levels of the costfunction.

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

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

2 Dynamic keeper model

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

$$\begin{aligned} Ad &= \begin{pmatrix} 1 & 0 & 0 & Ts & 0 & 0 \\ 0 & 1 & 0 & 0 & Ts & 0 \\ 0 & 0 & 1 & 0 & 0 & Ts \\ 0 & 0 & 0 & 1 - \frac{Ts\rho}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 - \frac{Ts\rho}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 - \frac{Ts}{100I} \end{pmatrix} & Bd &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{Ts}{m} & 0 & 0 \\ 0 & \frac{Ts}{m} & 0 \\ 0 & 0 & \frac{Ts}{I} \end{pmatrix} \\ Cd &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} & Dd &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{aligned}$$

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

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

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

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 similar to the stepresponse of the continuous system. Computing the reference states with different discretizations also gave similar results. As the discretized model using the euler rule was already analyzed, we will continue using that model.

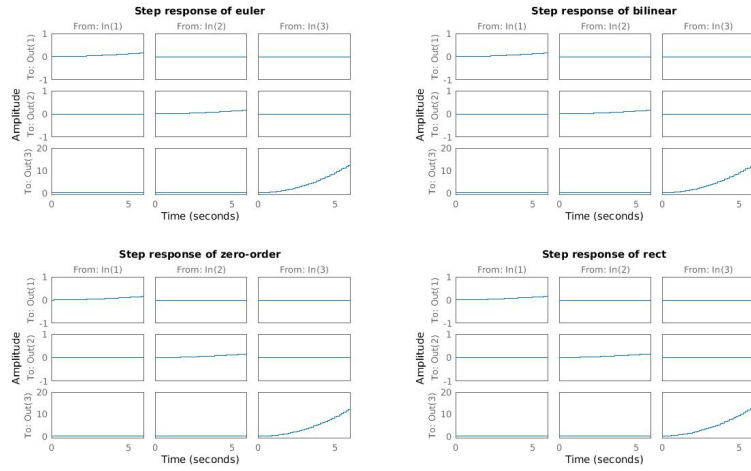


Figure 3: Stepresponses of four discretized versions of the original model

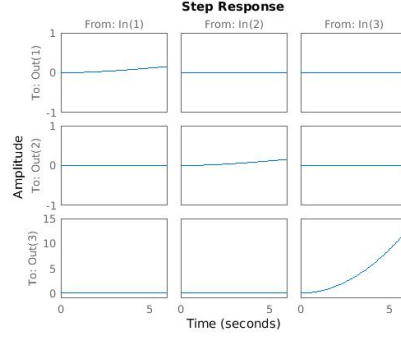


Figure 4: Stepresponse of the continuous model

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.

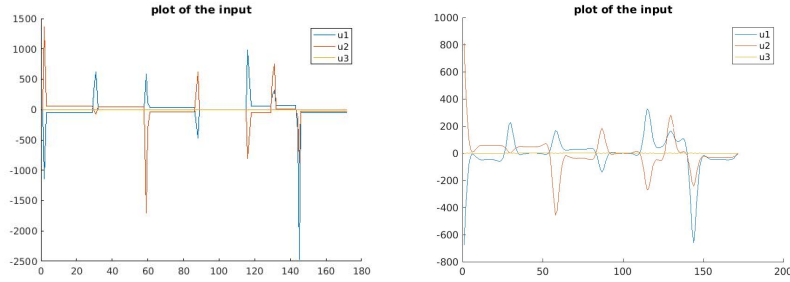


Figure 5: Optimal input calculated without and with a regularization term. For the input on the right the regularization term $\mu = 0.000001$

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.

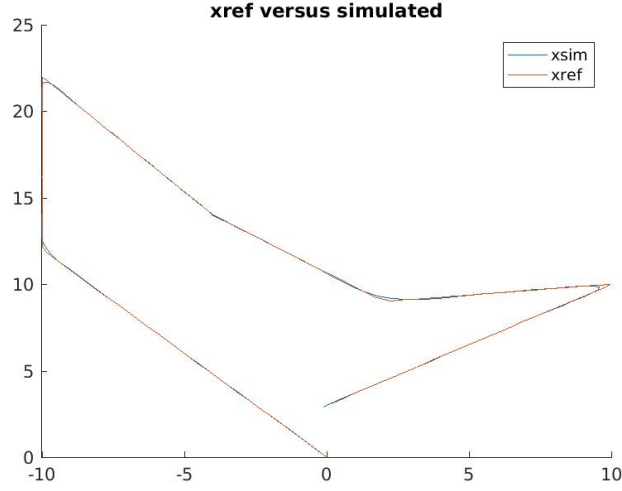


Figure 6: Plot of original reference path and the newly calculated references

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.

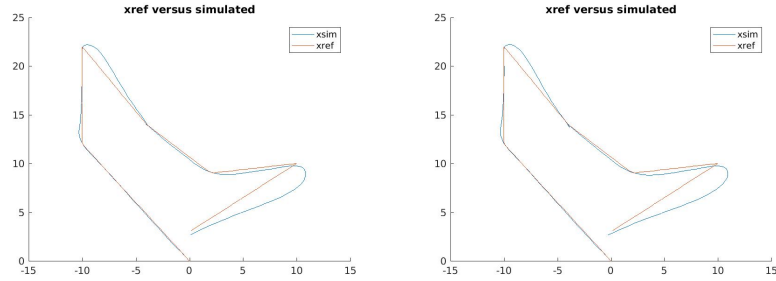


Figure 7: On the left the simulated trajectory with K_{lqr1} and on the right K_{lqr2}

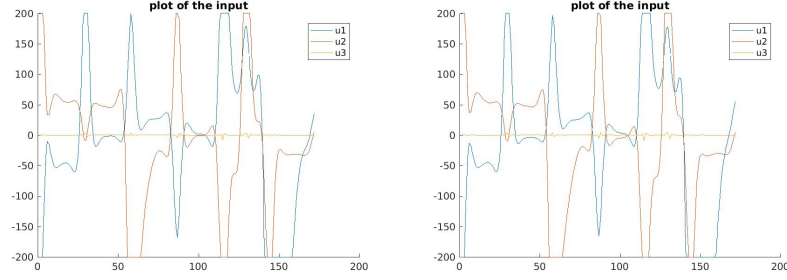


Figure 8: On the left the input with K_{lqr1} and on the right K_{lqr2}

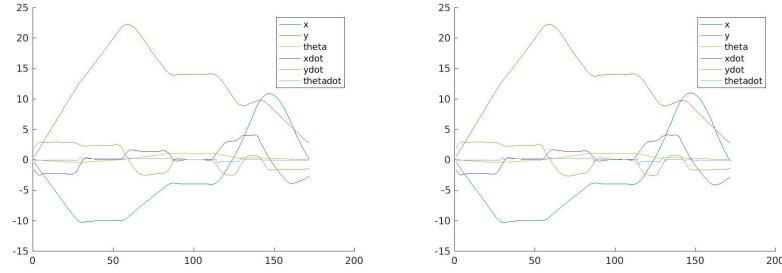


Figure 9: On the left the states of the simulated trajectory with K_{lqr1} and on the right K_{lqr2}

The cost of the simulation shows that the cost of the second controller is lower, namely 33.1349 versus 34.9471 respectively. The trajectories however look very similar. The fact that the plots are similar 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. The trajectory using a prediction horizon of 12 is shown below.

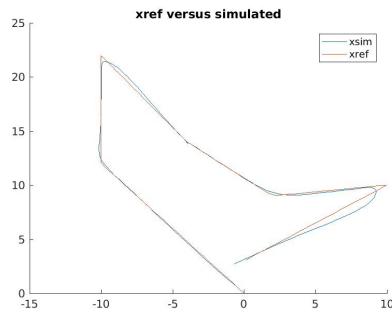


Figure 10: Simulate trajectory and the reference trajectory

We also plot the control actions and the state variables:

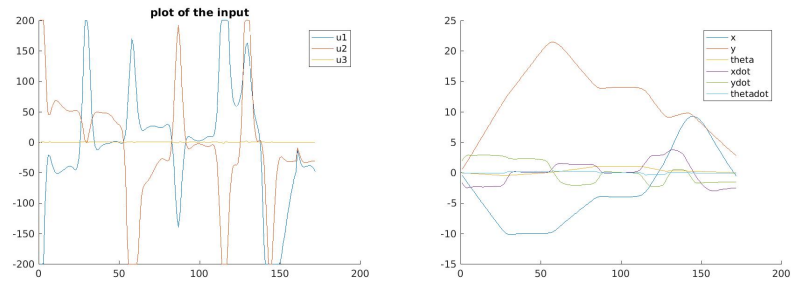


Figure 11: On the left control actions and on the right the states for an MPC controller with $N = 12$.