Predictive Control: Assignment 5

Olof Forsberg

***Abstract*— This text implements Model Predictive Control to a Crane. The crane is a pendulum with a mass. The problem of controlling the mass to follow a square is solved. In section A MPC without constraint and with soft constraints is implemented. Sektion B implements a Linearized Program Solver to the same problem. From the results it is motivated to use MPC with constraints and an LP Solver is sufficient to solve the tracking problem.**

I. A. TRACKING A SQUARE

A. *Definition of Problems*

We want to control the mass of a crane to follow a rectangle see fig. 1. We represent a rectangle by two dots according to 1 and fig. 1. With the point in 1 we define a periodic reference signal with period T as in 2. In order to make the mass follow the square we implement MPC so our state trajectories follow the reference signal 2.

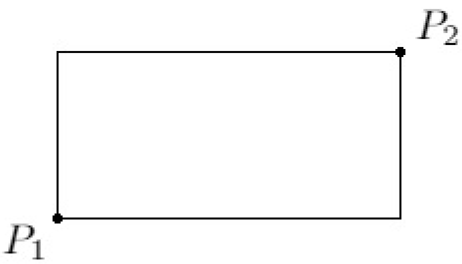

Fig. 1. Rectangle

$$P_1 = (x_1, y_1) \quad P_2 = (x_2, y_2) \tag{1}$$

$$r(t) = \begin{Bmatrix} \begin{pmatrix} x_2 \\ 0 \\ y_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & 0 \le t < \frac{T}{4} & \begin{pmatrix} x_2 \\ 0 \\ y_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \frac{T}{4} \le t < \frac{T}{2} \\ \begin{pmatrix} x_1 \\ 0 \\ y_2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \frac{T}{2} \le t < \frac{3T}{4} & \begin{pmatrix} x_1 \\ 0 \\ y_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} & \frac{3T}{4} \le t < T \end{Bmatrix} \tag{2}$$

The crane is described by a linearized model in 3

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y &= Cx \end{aligned} \tag{3}$$

B. *Optimal Control Problems*

In order to solve this tracking problem we use a soft constrained MPC to system 3, we can with this selection set limits on the angle of the pendulum. The result will be compared to a unconstrained RHC but this text will not go into detail about this controller. To implement the controller we are solving 4.

$$\begin{aligned} &\operatorname*{argmin}_{\bar{u}} \Big(\|x_N - x_e\|_P \\ &+ \sum_{k=0}^{N-1} (\|x_k - x_e\|_Q + \|u_k\|_R + \|s_{k+1}\|_{\boldsymbol{S}} + \rho \|s_{k+1}\|_1) \Big) \\ &s.t \quad x_{k+1} = Ax_k + Bu_k \\ &\check{u} \le u \le \hat{u} \quad k = 0, 1, ..., N-1 \\ &\check{c} - s_k \le Dx_k \le \hat{c} + s_k \quad k = 1, ..., N \end{aligned} \tag{4}$$

For the unconstrained RHC equation 4 changes. The constraints on u and x_k is taken away and every term with a s in the sum disappears.

C. *Optimization Problems*

$$\begin{aligned} &\operatorname*{argmin}_{\bar{u}} \frac{1}{2} \bar{u}^T H \bar{u} + (x_0 - x_e)^T G^T \bar{u} \\ &s.t \quad F\bar{u} \le \bar{b} + Jx_0 + Lx_e \end{aligned} \tag{5}$$

Our LQR in 4 can be converted to a optimisation problem on the form 5 see [2]. The unconstrained LQR takes on the same form in 5 but does not subject to the constraints.

D. *Parameters and Variables*

Our cost matrices will be the same in all implementations of controllers.

$$Q = \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

$$P = Q \quad R = \frac{1}{100} \mathbb{I}_8 \tag{7}$$

$$\boldsymbol{S} = \frac{1}{1000}\mathbb{I}_4 \tag{8}$$

$$\rho = 1000 \tag{9}$$

Parameter in 4 are set to

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{10}$$

$$\check{c} = \begin{pmatrix} 0 \\ 0 \\ -7^\circ \\ -7^\circ \end{pmatrix} \quad \hat{c} = \begin{pmatrix} 0 \\ 0 \\ 7^\circ \\ 7^\circ \end{pmatrix} \tag{11}$$

$$\check{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad \hat{u} = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \tag{12}$$

E. Implementation and Results

The points in 1 are selected to $(0.1, 0.1)$ and $(0.4, 0.4)$. The reference signal 2 is implemented in Matlab with functions pulstran and rectpuls, the period T in 2 is set to 20 s. In fig. 4 $r_1(t)$ and $r_3(t)$ are plotted in red on X and Y. Target x_e in 4 and 5 will now be signal 2.

Cost matrices are set according to 6,7. The sample time is selected to 0.15 s. And we select a horizon for our predictions to be 1.5 s. One important thing to consider here is to select the parameter so our solver find the solution within the sample time. Otherwise it will return a solution that is not optimal. The longest optimisation time was 0.4 s so with this selection we will be fine. We select the total time for the simulation to be 40 s our crane will do two turns.

System 3 is obtained first in continues time and then approximated in Matlab by the function c2d. Equation 4 is converted to the optimisation problem 5 [2]. Equation 5 is solved used mpcqpsolver in Matlab. We solve it for the specified horizon in several iterations. During each iteration the solution is implemented as our control output. This controller is implemented on two models in Matlab the linearised in 3 and a Nonlinear model of the crane in Simulink.

The result of how the mass is moving for the soft constrained QP can be seen in fig. 2 and fig. 3. We see that or QP can follow the square but struggles in the corners, if we dislike this behavior we can harden our angle constraints. In fig. 7 we can see that the input for x and y is on the boundary of our constraints around time zero. This is a reason to use a control model with constraints. If point two in 1 are further away say $(0.6, 0.6)$ this controller will violate our input constraints.

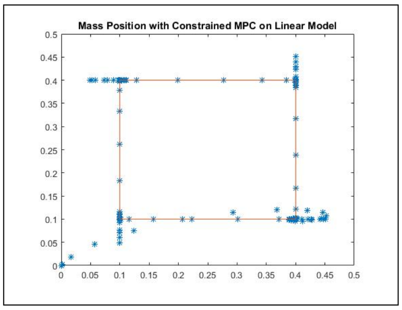

Fig. 2. Mass movement with Constrained MPC on Linear Model

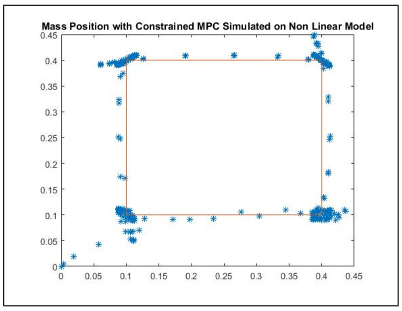

Fig. 3. Mass movement with Constrained MPC on Non Linear Model

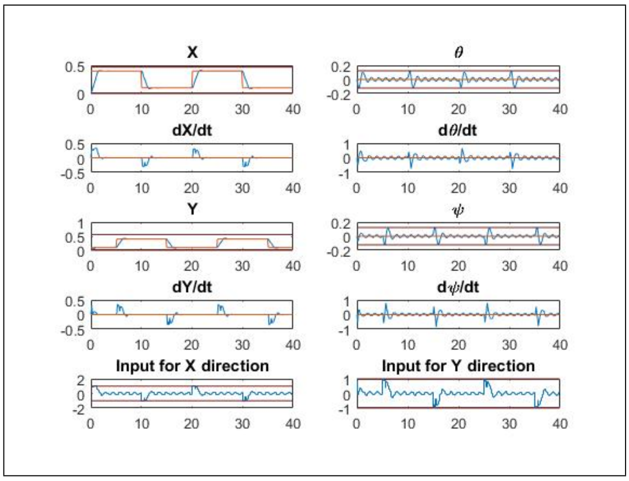

Fig. 4. Reference Signal

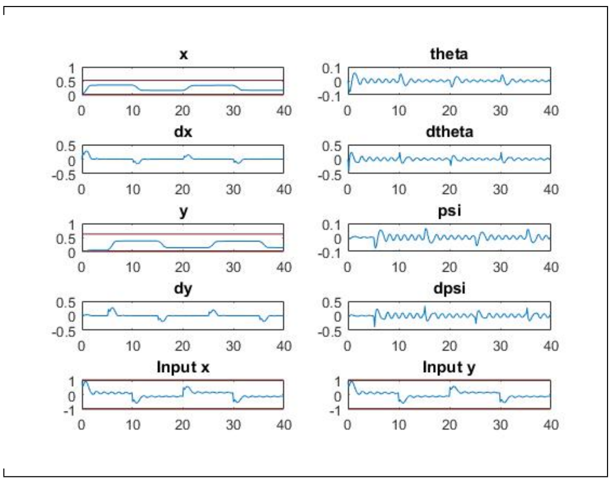

Fig. 5. Reference Signal

II. B. Implementing LP solver

An LP solver is implemented to track a reference point.

A. New Parameters

We introduce the parameters in 13 to use in 16

$$\boldsymbol{\hat{u}} = \begin{pmatrix} \hat{u} \\ \vdots \\ \hat{u} \end{pmatrix} \quad \boldsymbol{\check{u}} = \begin{pmatrix} \check{u} \\ \vdots \\ \check{u} \end{pmatrix} \tag{13}$$

B. Definition of Problems

$$\underset{\bar{u},x}{\operatorname{argmin}}\Big(\|P(x_N - x_e)\|_1 + \sum_{k=0}^{N-1} (\|Q(x_k - x_e)\|_1 + \|Ru_k\|_1) \Big) \tag{14}$$

$$s.t \quad x_{k+1} = Ax_k + Bu_k$$

$$\check{u} \leq u \leq \hat{u} \quad k = 0, 1, ..., N-1$$

For an LP we want to solve equation 14 and implement the first element of $\bar{u}$ in our controller.

C. Optimization Problems

$$\min_{\bar{u},x} \quad t \tag{15}$$

$$s.t \quad -t \leq (Ax - b) \leq t$$

To find the solution of 14 we can transform it to 15 and receive the optimization problem 16 see [1] or [3].

$$\min_{\bar{u}} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} \bar{u} \\ s \end{pmatrix} \tag{16}$$

$$s.t \quad \begin{pmatrix} \bar{Q}\Gamma & -I \\ \bar{R} & -I \\ -\bar{Q}\Gamma & -I \\ -\bar{R} & -I \\ I & 0 \\ -I & 0 \end{pmatrix} \begin{pmatrix} \bar{u} \\ s \end{pmatrix} \leq \begin{pmatrix} -\bar{Q}\Phi x_0 + \bar{Q}\Phi x_e \\ 0 \\ \bar{Q}\Phi x_0 - \bar{Q}\Phi x_e \\ 0 \\ \boldsymbol{\hat{u}} \\ -\boldsymbol{\check{u}} \end{pmatrix}$$

D. Implementation and Results

Parameters are set as 6, 7 we use the same reference signal 2. Both the horizon, sample time and duration of simulation in section I are used here. An LP solver is implemented in matlab by solving 15 with the function linprog. It is solved in iterations and the first element in $\bar{u}$ is used as output from our controller to the system. The controller is simulated with the linearised model 3 as in section I.

III. Conclusion

The unconstrained RHC was on the limit to violate the constraints defined in 12. For other squares it would and a choice of a constrained controller is motivated.

A LP solver works fine for this problem however it was more time consuming that the QP solver. The LP has less complexity but because it needs more constraints and it results in using more time [3]. If it is specifically interesting to not penalize big deviations for a problem then the LP is a good choice. One implementation is minimizing fuel LP [4].

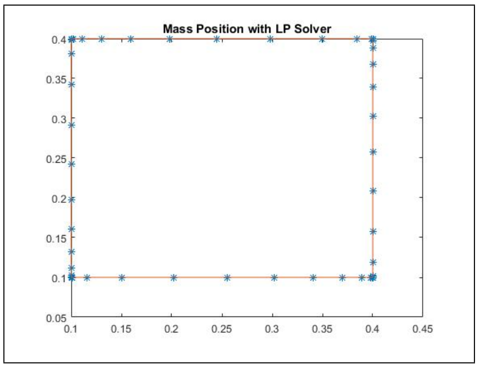

Fig. 6. Mass Movement with LP Solver

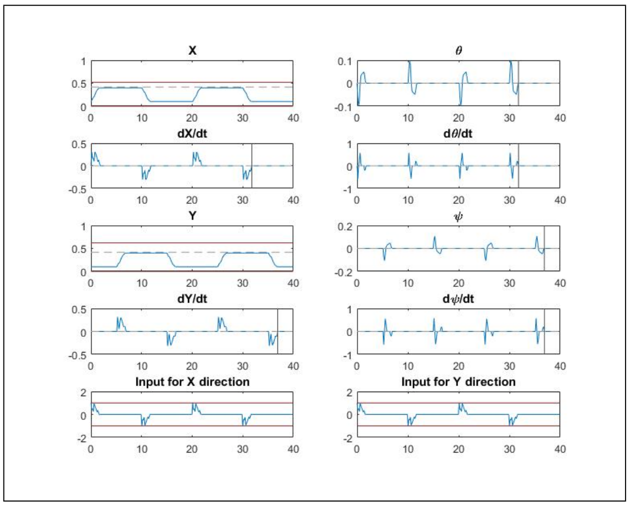

Fig. 7. Reference Signal

IV. DISCUSSION

To steer the mass in square movement is practical and when implemented able to solve problems. Consider a rectangle obstacle in the way of where we would like the mass to move. If we would define constraints in a way that our state is not allowed to be within the box we have a problem to apply MPC. This is because we defined a nonconvex set to solve MPC over, which is not possible [5]. A different method could then be to let the state trajectories follow a rectangle shape bigger than the obstacle until it reaches its target.

References

[1] Alberto Bemporad, Francesco Borrelli, Manfred Morari, et al. “Model predictive control based on linear programming˜ the explicit solution”. In: *IEEE Transactions on Automatic Control* 47.12 (2002), pp. 1974–1985.

[2] A. Bemporad et al. “The explicit solution of model predictive control via multiparametric quadratic programming”. In: *American Control Conference, 2000. Proceedings of the 2000*. Vol. 2. 2000, 872–876 vol.2. DOI: 10.1109/ACC.2000.876624.

[3] A. Dötlinger and R. M. Kennel. "Near time-optimal model predictive control using an L1-norm based cost functional". In: *Energy Conversion Congress and Exposition (ECCE), 2014 IEEE*. Sept. 2014, pp. 3504–3511. DOI: 10.1109/ECCE.2014.6953877.

[4] Edward N. Hartley et al. "Model predictive control system design and implementation for spacecraft rendezvous". In: *Control Engineering Practice* 20.7 (2012), pp. 695–713. ISSN: 0967-0661. DOI: http://dx.doi.org/10.1016/j.conengprac.2012.03.009. URL: http://www.sciencedirect.com/science/article/pii/S0967066112000664.

[5] L. Singh and J. Fuller. "Trajectory generation for a UAV in urban terrain, using nonlinear MPC". In: *American Control Conference, 2001. Proceedings of the 2001*. Vol. 3. 2001, 2301–2308 vol.3. DOI: 10.1109/ACC.2001.946095.