Active Set Method Based Model Predictive Control for a Ball and Beam System

Handan Nak[1], Şirin Akkaya[1], and Erhan Yumuk[1]

[1] Department of Control and Automation Engineering, Istanbul Technical University, Istanbul, Turkey
handan.nak@itu.edu.tr, sakkaya@itu.edu.tr, yumuk@itu.edu.tr

Abstract

Model predictive control is an optimal control method that requires the solution of a quadratic programming problem with constraints at each sampling time. In this study, an active set method is used to solve the problem. The whole control algorithm is written through Matlab editor so that it can be run in embedded systems. A ball and beam system model is utilized to investigate the performance of the controller structure. The simulations are conducted with two different cost functions. The results illustrate that the constructed control system exhibits satisfactory performance in the sense of system response and constraints.

1. Introduction

Model predictive control (MPC) is a well-known model based optimal control method for systems with constraints. MPC has used by most of the academic researches and industrial applications in these days, however, its history dates back to 1970s. Most primitive examples of MPC are model predictive heuristic control method [1], dynamic matrix control method [2] and quadratic matrix control method [3]. The first comprehensive exposition of generalized predictive control is presented in [5] and [6]. The formulation of MPC based on the state space approach is presented in [7]. A discrete time MPC design using Laguerre functions are in a discrete form is given in [8]. A solution to the problem of robust MPC of constraint linear discrete time systems in the presence of bounded disturbances is given in [9], and a neural network based MPC is given [10]. As the original MPC algorithm is developed for linear systems, however, it is extended for non-linear systems in time [11], [12]. Past achievements, some current developments and a few avenues for future research are given in [13]. Some implementation of MPC for magnetic levitation system is presented in [14], for ball and beam system in [15], [16].

In classical MPC design, the control problem that contains the linear model of the system, constraints and a cost function is transformed into a quadratic programming (QP) problem. And the control signal is obtained by solving the QP problem in each step by utilizing an optimization algorithm. Active set method (ASM) and interior-point method (ISM) are the most commonly employed approaches for solving QP, and the other algorithms are generally based on these methods. The computational complexity of ISM is larger compared with ASM per iteration. ASM has less complexity and converges faster when the number of variables and the constraints are small [17].

In this study, a MPC that uses ASM algorithm is designed for linear time invariant systems. The ASM algorithm is written by hand by the help of Matlab m-file editor. The performance of the controller is investigated by using a ball and beam model in Matlab Simulink. The motivation of the study is to acquire an MPC algorithm with optimization code that can be compiled and used in embedded systems.

The paper is organized as follows: the general information about MPC, problem description and primal active set method is given in Section 2. The modelling of ball and beam system and simulation results are given in Section 3 and the conclusions are highlighted in Section 4.

2. Model Predictive Control

Model predictive control is a model based optimal control method that solves the constrained finite-horizon optimization problem by predicting the future behavior of system variables using the current state of the system at each sampling time. The predictions along the prediction and control horizon are calculated in order to minimize a cost function that generally depends on error and control signal. Only the first element of the obtained optimal control sequence is applied to the real system and the whole algorithm is repeated by measuring or observing the system output at the next sampling time.

In the method, the cost function to be optimized depends on error and control signals along prediction and control horizon, respectively. The optimal control sequence that minimizes the cost function is obtained along the control horizon by using the prediction of system states. Only the first element of the sequence is applied to the real system and the whole algorithm is repeated by measuring or estimating the system output at the next sampling time. The receding horizon control strategy provides the system a feedback and in this way, it is possible allows to compensate the modeling errors and the disturbances that affect to the system [18], [19].

Basically, a MPC loop consists of a system model, a cost function and a optimization tool. There are two essential parameters in the loop: Prediction horizon K_y and control horizon K_u. Whereas the prediction horizon refers to the length of horizon to be predicted, the control horizon defines the number of elements in the candidate control sequence to be applied to the system during the prediction horizon. Therefore, the inequality $K_u \leq K_y$ must always be satisfied and the elements after the K_u^{th} of candidate control sequence must be equal to the K_u^{th} element of the sequence. The basic structure of MPC is shown in Fig. 1 [20], [21].

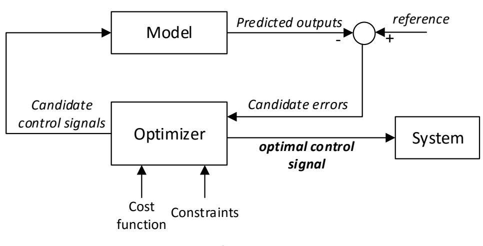

Fig. 1. The basic structure of the model predictive control.

Briefly, an MPC loop consists ofhas three basic steps:
1. Step 1: The prediction vector is obtained from the system model using the measure or estimated system states and optimized candidate control sequence.
2. Step 2: The candidate control sequences that minimizes the cost function are determined evaluated by a suitable optimization method. The cost function is often chosen as error between the reference and output and a structure penalizing control sequence signal.
3. Step 3: The first element of the optimized control sequence is applied to the real system. Then, the next sample is taken and Step 1 is repeated.

2.1. Problem Description

A discrete-time system single input single output linear time invariant system is considered. Such a discrete-time system is expressed in the state space form as follows:

$$\begin{aligned} x[n+1] &= A\,x[n] + B\,u[n] \\ y[n] &= C\,x[n] \end{aligned} \tag{1}$$

where $x[n] \in \mathbb{R}^n$, $u[n] \in \mathbb{R}$, $y[n] \in \mathbb{R}$ are state variables, input and output signals, respectively. *A*, *B* and *C* matrices are the system matrices discretized with sampling time T_s. The general control problem, under assumption that the reference signal ($r[n]$) is known in advance, is described as to find optimal control signal ($u[n]$) that minimizes the tracking error of controlled system for future values along prediction horizon with subject to the constraints on control signal and system states

$$\begin{aligned} u_{min} \le u[n+k] \le u_{max}, \quad & k = 0,1,\cdots,K_u \\ |u[n+k] - u[n+k-1]| \le \Delta u_{max}, \quad & k = 1,\cdots,K_u \\ x_{min} \le x[n+k] \le x_{max}, \quad & k = 0,1,\cdots,K_y \end{aligned} \tag{2}$$

Accordingly, the cost function can be written as

$$f(\boldsymbol{u}) = \sum_{k=1}^{K_y} (r[n+k] - \hat{y}[n+k])^2 + \lambda \sum_{k=1}^{K_u} (u[n+k] - u[n+k-1])^2 \tag{3}$$

where y̑$[n]$ is predicted output, λ is a parameter which penalizes changes in control signal. System output can be predicted for each step using (1). When the prediction of output is continued along prediction horizon, the cost function is written in the form

$$f(\boldsymbol{u}) = \boldsymbol{u}^T(\boldsymbol{M}^T\boldsymbol{M} + \lambda\,\boldsymbol{L})\boldsymbol{u} - (2(\boldsymbol{R_n} - \boldsymbol{Z}\,\hat{\boldsymbol{x}}[n])^T\boldsymbol{M} + 2[\lambda\,u[n-1] \quad 0 \quad \cdots \quad 0])\,\boldsymbol{u} + S \tag{4}$$

where

$$\boldsymbol{L} = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \ddots & 0 & 0 \\ 0 & 0 & -1 & \ddots & -1 & 0 \\ 0 & 0 & 0 & \ddots & 2 & -1 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}_{(K_u+1)\times(K_u+1)}$$

$$\boldsymbol{M} = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ C\,A\,B & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\,A^{K_u}B & C\,A^{K_u-1}B & C\,A^{K_u-2}B & \cdots & C\,B \\ C\,A^{K_u+1}B & C\,A^{K_u}B & C\,A^{K_u-1}B & \cdots & \sum_{i=0}^{1} C\,A^iB \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C\,A^{K_y-1}B & C\,A^{K_y-2}B & C\,A^{K_y-3}B & \cdots & \sum_{i=0}^{K_y-K_u-1} C\,A^iB \end{bmatrix},$$

$$\boldsymbol{Z} = \begin{bmatrix} C\,A \\ C\,A^2 \\ \vdots \\ C\,A^{K_u+1} \\ C\,A^{K_u+2} \\ \vdots \\ C\,A^{K_y} \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} u[n] \\ u[n+1] \\ \vdots \\ u[n+K_u] \end{bmatrix}, \boldsymbol{R_n} = \begin{bmatrix} r[n+1] \\ r[n+2] \\ \vdots \\ r[n+K_u+1] \\ r[n+K_u+2] \\ \vdots \\ r[n+K_y] \end{bmatrix},$$

$$S = (\boldsymbol{R_n} - \boldsymbol{Z}\,\hat{\boldsymbol{x}}[n])^T(\boldsymbol{R_n} - \boldsymbol{Z}\,\hat{\boldsymbol{x}}[n]) + \lambda\,u^2[n-1]$$

Note that *S* does not depend on the control variables, thus it is not included in the optimization process [20], [22]. In some cases, the cost function in (4) is not sufficient for fast system response. As a solution to this problem, it is possible to multiply the tracking error term with time in (4), as it is in ITSE (integral time-square error) index. In this case, the cost function becomes

$$f(\boldsymbol{u}) = \boldsymbol{u}^T(t\,\boldsymbol{M}^T\boldsymbol{M} + \lambda\,\boldsymbol{L})\boldsymbol{u} - (2\,t\,(\boldsymbol{R_n} - \boldsymbol{Z}\,\hat{\boldsymbol{x}}[n])^T\boldsymbol{M} + 2[\lambda\,u[n-1] \quad 0 \quad \cdots \quad 0])\,\boldsymbol{u} + S \tag{5}$$

where *t* denotes time. As a result, MPC problem can be expressed as

$$\begin{aligned} & \min_u f(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T G\,\boldsymbol{u} + g^T\,\boldsymbol{u} \\ & u_{min} \le u[n+k] \le u_{max}, \; k = 0,1,\cdots,K_u \\ \text{subject to } & |u[n+k] - u[n+k-1]| \le \Delta u_{max}, \; k = 1,\cdots,K_u \\ & x_{min} \le x[n+k] \le x_{max}, k = 0,1,\cdots,K_y \end{aligned}$$

where $G = 2(\boldsymbol{M}^T\boldsymbol{M} + \lambda\,\boldsymbol{L})$ or $G = 2\,t\,(\boldsymbol{M}^T\boldsymbol{M} + \lambda\,\boldsymbol{L})$ and $g^T = -(2(\boldsymbol{R_n} - \boldsymbol{Z}\,\hat{\boldsymbol{x}}[n])^T\boldsymbol{M} + 2[\lambda\,u[n-1] \quad 0 \quad \cdots \quad 0])$ or $g^T = -(2\,t\,(\boldsymbol{R_n} - \boldsymbol{Z}\,\hat{\boldsymbol{x}}[n])^T\boldsymbol{M} + 2[\lambda\,u[n-1] \quad 0 \quad \cdots \quad 0])$ depends on the type of cost function.

This is a quadratic programming (QP) problem with inequality constraints. The solution of the problem is in the form

$$\boldsymbol{u}^* = [u[n] \quad u[n+1] \quad \cdots \quad u[n+K_u]] \tag{6}$$

and the first element of the control vector is applied to the system at each sampling time. Note that the optimization time must be always less than sampling time.

2.2. Primal Active Set Method

The QP with inequality constraints is defined as

$$\begin{aligned} & \min_u f(x) = \frac{1}{2}\,x^T\,G\,x + g^T x \\ & \text{subject to } {a_i}^T x \ge b_i \end{aligned}$$

where $G \in \mathbb{R}^{n\times n}$ is a symmetric and positive definite matrix, and the region $\Omega = \{x \in \mathbb{R}^n : {a_i}^T x \ge b_i\,, i \in I\}$ is called as feasible region. The basic principal of primal active set method is to evalute a feasible sequence $\{x_k \in \Omega\}$ which decreases the value of the cost function, $f(x_k+1) < f(x_k)$. The method starts with a feasible initial point and ensures all iterations remain in the feasible region.

In the method, for each step, the set of inequality constraints provided as equality is utilized.This set is named as current active set and denoted by $\mathcal{A}(x_k)$, $\mathcal{A}(x_k) = \{i \in I : a_i^T x_k = b_i\}$. For the feasible point x_k the corresponding working set is denoted by $\mathcal{W}_k$. $\mathcal{W}_k$ is a subset of the active set $\mathcal{A}(x_k)$, $\mathcal{W}_k \subset \mathcal{A}(x_k)$, and its colums are selected from linearly independent columns of $\mathcal{A}(x_k)$.

Then, the equality constraint QP subproblem with the feasible point x_k and working set $\mathcal{W}_k$ at the k[th] iteration defined as follows:

$$\min_{p \in \mathbb{R}^n} f(p) = \frac{1}{2} p^T G p + (G x_k + g)^T p$$
$$\text{subject to } a_i^T p = 0,\ i \in \mathcal{W}_k$$

where p is improving direction.. The Karush-Kuhn-Tucker (KKT) conditions that solve the above problem are defined as

$$\begin{aligned} &G x_k + g - \sum_{i \in I} a_i \mu_i^* = 0 \\ &a_i^T p^* = b_i \qquad \mu_i^* \geq 0 \qquad i \in \mathcal{W}_k \\ &a_i^T p^* \geq b_i \qquad \mu_i^* = 0 \qquad i \in I \backslash \mathcal{W}_k \end{aligned} \tag{7}$$

where μ_i is Lagrange multiplier. To decide how far to move along the direction p^*, the step length α_k in the range [0, 1) is calculated by the following formula:

$$\alpha_k = min\left(1, \min_{i \in I \backslash \mathcal{W}_k : a_i^T p^* < 0} \frac{b_i - a_i^T x}{a_i^T p^*}\right) \geq 0 \tag{8}$$

The step length provides greatest reduction of the objective function in the p^* direction without leaving the feasible region. Then, the next feasible point (x_{k+1}) is calculated, $x_{k+1} = x_k + \alpha_k p^*$.

If $\alpha_k < 1$, it means that the uni step in the optimal improving direction (p^*) intersects with a constraint that is not in the set $\mathcal{W}_k$ and thus the intersecting constraint is added to the next working set ($\mathcal{W}_{k+1}$). On the other hand, if at least one of the Lagrange multipliers (μ_i) is negative when $p^* = 0$, the constraint corresponding to the most negative Lagrange multiplier is removed from the working set. In such a case, the solution is not optimal because the KKT conditions is not satisfied and removing the constraint with the most negative Lagrange multiplier reduces the cost function [23], [24].

These steps are repeated until $p^* = 0$ which is optimal solution satisfied with conditions $\mu_i^* \geq 0, i \in \mathcal{W}_k$. The detailed algorithm of the method is given in Table 1.

3. Simulation Studies

3.1. System Modeling

In this section we desire to control a ball and beam system. The objective of ball and beam system is to stabilize the ball to a desired position along the beam. As schematically illustrated in Fig. 2, a ball and beam system is comprised of two plants: a servo motor with transmission elements and beam element that allows the ball to roll freely with nonlinear dynamics. By controlling the position of the servo motor, hence the motor voltage, beam angle is adjusted to balance the ball to a desired position [25].

Table 1. Primal active set method algorithm.

Initialization: Specify a feasible point (x_0) and then create the corresponding working set $\mathcal{W}_0$ oluştur.

while NOT STOP do *%Calculation of the improving direction%*

Calculate the improving direction, p^*, by solving QP problem with equality constraints.

$$\min_{p \in \mathbb{R}^n} f(p) = \frac{1}{2} p^T G p + (G x_k + g)^T p$$
$$\text{subject to } a_i^T p = 0,\ i \in \mathcal{W}_k$$

if $\|p^*\| = 0$ **then** %Calculation *of Lagrange multipliers %*

Calculate Lagrange multipliers

$$\sum_{i \in \mathcal{W}} a_i \mu_i = G x + g$$
$$i \in I \backslash \mathcal{W} \text{ ise } \mu_i = 0$$

if $\mu_i \geq 0$ for $\forall\, i \in \mathcal{W}$ **then**

STOP *% optimal solution is founded %*

else *% removing constraint%*

$x = x$

Remove constraint $\mathcal{W}_j$ for $\mu_j < 0$

end

else *%calculation of the step length %*

$$\alpha = min\left(1, \min_{i \in I \backslash \mathcal{W}_k : a_i^T p^* < 0} \frac{b_i - a_i^T x}{a_i^T p^*}\right)$$

$$J = \arg\min_{i \in I \backslash \mathcal{W}_k : a_i^T p^* < 0} \frac{b_i - a_i^T x}{a_i^T p^*}$$

if $\alpha < 1$ **then** *%appending a constraint%*

$x = x + \alpha p^*$

Append constraint $\mathcal{W}_j$

else

$x = x + p^*$

$\mathcal{W} = \mathcal{W}$

end

end

end

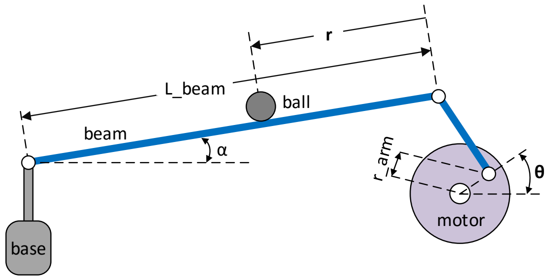

Fig. 2. Ball and beam schematic diagram.

The linearized (around origin) dynamic motion equation of ball and beam system is writen as

$$\begin{aligned} \frac{d^2\theta(t)}{dt^2} &= -\frac{1}{\tau}\frac{d\theta(t)}{dt} + \frac{K}{\tau}u(t) \\ \frac{d^2 r(t)}{dt^2} &= K_{bb}\,\theta(t) \end{aligned} \tag{9}$$

where $\theta(t)$ is the motor angular position, $u(t)$ is the motor voltage, $r(t)$ is the ball position, K is the motor and transmission steady state gain, τ is the motor time constant and K_{bb} is the gain between ball position and motor angle. The states of system are defined as $[x]^T = [x_1 \quad x_2 \quad x_3 \quad x_4]^T = [\theta(t) \quad \dot{\theta}(t) \quad r(t) \quad \dot{r}(t)]^T$, system input and output are defined as motor voltage and ball position, respectively. Then the state space representation of the system can be written as

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{\tau} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ K_{bb} & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{K}{\tau} \\ 0 \\ 0 \end{bmatrix} u \qquad (10)$$
$$y = [0 \quad 0 \quad 1 \quad 0]\, x$$

The system also has the following constraints on system input and states due to the mechanical and electrical structure of the system:

$$\begin{aligned} u_{min} &\leq u(t) \leq u_{max} \\ \theta_{min} &\leq \theta(t) \leq \theta_{max} \\ x_{min} &\leq x(t) \leq x_{max} \end{aligned} \qquad (11)$$

As seen from (10), system matrix, A has two eigenvalue at the origin, which destroys the stability of system. Therefore, before applying MPC, the system must be prestabilized with a state feedback matrix, L that makes $A - B\,L$ matrix Hurwitz. Then, the control signal, with MPC and state feedback, applied to the system will become $u - L\,x$. Once system is stabilized, next step is to discretize the system. Here, since ball and beam system is a relatively slow system, sampling time is chosen as $T_s = 0.1$ for discritezation.

In MPC algorithm, prediction horizon K_y, control horizon K_u and penalty parameter λ are selected as 12, 3 and 0.3 respectively. The constant matrices M and Z and depending on them G matrix in the cost function are calculated as given in previous section. Note that the vector g, is updated at each step according to the current state, the previous control signal and the reference vector.

Final step is to transform the constraints on control signal and states into the form $a_i{}^T u[n+k] \geq b_i$ for $k = 0,1,\cdots,K_u$ to be used in primal active set method. For detailed explanation for these inequality expression transforms, see [20, 22].

With these adjustments, the QP with inequality constraints is established and the control signal applied to the system at each step is calculated by means of the written primal active set method algorithm.

3.2. Simulation Results

To investigate the performance of written MPC algorithm, the simulation diagram in Fig. 3 is constructed in Matlab/Simulink. The parameters of ball and beam system are taken from Quanser experiment set [25] and given in Table 2.

The cost function evaluation and primal active set algorithm are integrated to the simulation through embedded Matlab functions. For prestabilization, the gain matrix $L = [3.67 \quad 0.08 \quad 28.86 \quad 22.54]$ is utilized.

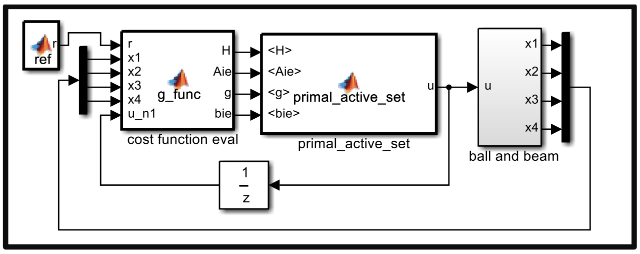

Fig. 3. Block diagram of ball and beam control system with MPC.

Table 2. Ball and beam system parameters.

Parameter	Value
Motor steady state gain, K	1.53 $[rad/(s\,V)]$
Time constant, τ	0.0248 $[s]$
Linearized gain of beam dynamics, K_{bb}	0.418 $[m/(s^2\,rad)]$
Min, max motor voltage, u_{min}, u_{max}	± 12 $[V]$
Min, max motor position, θ_{min}, θ_{max}	$\pm\,\pi/4$ $[rad]$
Min, max ball position, x_{min}, x_{max}	0 $[cm]$ and 0.5 $[cm]$

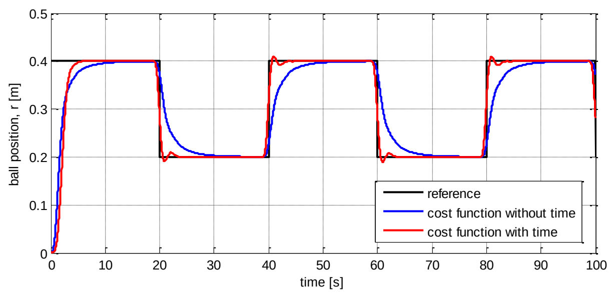

Fig. 4. Ball position using cost function with and without time.

In simulations, a square wave with frequency of 0.025 Hz and amplitude of 0.2 is used as reference ball position. The simulations are performed for two cost functions, cost function without time given in (4) and cost function with time given in (5). The results for ball position, motor voltage (control signal) and motor angular position are given in Figs. 4 – 6, respectively. In both cases motor angular position and voltage are within their boundaries.

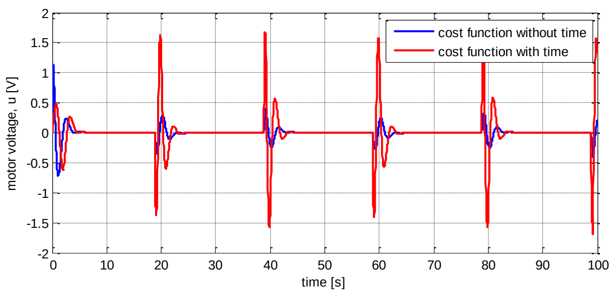

Fig. 5. Motor voltage using cost function with and without time.

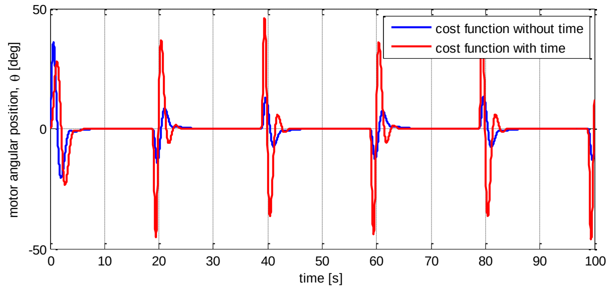

Fig. 6. Motor angular position using cost function with and without time.

As seen from figures, in case of cost function with time system response is faster but has a small overshoot as expected. In the first cycle, as the simulation has just started, MPC algorithm cannot predict the first reference change in advance and therefore, the settling time is larger. Nevertheless, this phenomenon is not experienced clearly in case of cost function without time, because the term that penalizes control signal dominates the tracking error in the cost function.

It is worth noting that, when considering the real system the motor voltage needs to exceed a certain value to move the beam. Therefore, in case of cost function without time the change of control signal is not reasonable.

4. Conclusion

In this study, a model predictive controller is designed for linear systems based on the active set algorithm as the optimization method. The whole algorithm is written in Matlab editor by authors, and it is suitable for run in embedded systems. For the case study, a ball and beam system is practiced. Simulations are conducted to verify the performance of the proposed structure and found that the transient and steady state response is satisfactory and constraints are provided. Significant improvements of the transient response and admissible control signal for real system are obtained when time term is added to the cost function in MPC algorithm. As future work, it is planned to apply this MPC algorithm to Quanser ball and beam system.

7. References

[1] J. A. Richalet, A. Rault, J. L. Testud and J. Papon, “Model predictive heuristic control: applications to an industrial process,” *Automatica*, vol. 14, no. 5, pp. 413-428, 1978.

[2] C. R. Cutler, B.C. Ramaker “Dynamic matrix control: a computer control algorithm”, *In Proceedings of the Joint Automatic Control Conference*, San Francisco, CA, 1980.

[3] C. R. Cutter, A. Morshedi, J. Haydel, “An industrial perspective on advanced control”, *In AICHE annual meeting*, Washington, DC, 1983.

[4] C. E. Gachia, A. M. Morshedi, “Quadratic programing solution of dynamic matrix control,” *Chemical Engineering Communications*, vol. 46, pp. 73 – 87, 1986

[5] D. W. Clarke, C. Mohtadi, P. C. Tuffs, “Generalized predictive control–part 1: the basic algorithm,” *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[6] D. W. Clarke, C. Mohtadi, P. C. Tuffs, “Generalized predictive control–part 2: extensions and interpretations,” *Automatica*, vol. 23, no. 2, pp. 149–163, 1987.

[7] M. Morari, J. H. Lee, C. E. Garcia, “State-Space Interpretation of Model Predictive Control,” *Automatica*, vol. 30, no. 4, pp. 707-717, 1994.

[8] L. Wang, “Discrete model predictive controller design using Laguerre functions,” *Journal of process control*, 14(2), 131-142, 2004.

[9] D. Q. Mayne, et all, “Robust model predictive control of constrained linear systems with bounded disturbances,” *Automatica*, vol. 41, no. 2, pp. 219-224, 2005.

[10] L. Cheng, W. Liu, Z. G. Hou, J. Yu, M. Tan, “Neural-network-based nonlinear model predictive control for piezoelectric actuators,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7717-7727, 2015.

[11] E. F. Camacho and C. Bordons, “Nonlinear model predictive control: an introductory review,” Lecture Notes in Control and Information Sciences, vol. 358: 1–16, 2007.

[12] S. Iplikci, “Runge–Kutta model-based adaptive predictive control mechanism for non-linear processes,” *Transactions of the Institute of Measurement and Control*, vol. 35, no. 2, pp. 166-180, 2013

[13] D. Q. Mayne, “Model predictive control: Recent developments and future promise.” *Automatica*, vol. 50, no. 12, pp. 2967-2986, 2014.

[14] L. Rušar, A. Krhovják, V. Bobál, “Predictive control of the magnetic levitation model,” *International Conference on Process Control*, pp. 345-350, 2017.

[15] D. Zhe, Z. Geng, L. Guoping, “Networked nonlinear model predictive control of the ball and beam system,” *Chinese Control Conference*, pp. 469-473, 2008

[16] N. Hara, M. Takahashi, K. Konishi, “Experimental evaluation of model predictive control of ball and beam systems,” *In American Control Conference*, pp. 1130-1132, 2011.

[17] H. J. Ferreau, P. Ortner, P. Langthaler, L. del Re, M. Diehl, “Predictive control of a real-world diesel engine using an extended online active set strategy,” *Annual Reviews in Control*, vol. 31, no. 2, pp. 293-301, 2007.

[18] M. N. Zeilinger, “Real-time model predictive control,” Ph.D. dissertation, ETH Zurich, 2011.

[19] S. Kouro, P. Cortes, R. Vargas, U. Ammann, and J. Rodriguez, “Model predictive control - A simple and powerful method to control power converters,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1826–1838, June 2009.

[20] S. İplikçi, “Optimization Techniques and Model Predictive Control – MPC,” Lecture Notes, 5. EMO Akademic Camp, 23-26 Mart 2017.

[21] P. Orukpe, “Model predictive control fundamentals,” *Nigerian Journal of Technology*, vol. 31, no. 2, pp. 139–148, 2012.

[22] B. Bahtiyar, “FPGA ile gerçek-zamanlı Runge-Kutta model öngörülü kontrol,” Ph.D. dissertation, Pamukkale University, 2015.

[23] E. L. Hansen, C. Völcker, “Numerical Algorithms for Sequential Quadratic Optimization,” Ph.D. dissertation, Technical University of Denmark, 2007.

[24] N. Jorge and J. W. Stephen, “Numerical optimization,” Springerverlang, USA, 1999.

[25] Quanser, “SRV02 Ball and Beam - User Manual,” Document number 709, Revision 2.2.