Model Predictive Control of a Ball and Plate laboratory model

Matej Oravec
Department of Cybernetics and Artificial Intelligence,
Faculty of Electrical Engineering and Informatics, Technical University in Košice, Slovakia
matej.oravec@tuke.sk

Anna Jadlovská
Department of Cybernetics and Artificial Intelligence,
Faculty of Electrical Engineering and Informatics, Technical University in Košice, Slovakia
anna.jadlovska@tuke.sk

***Abstract* — The papers presents an implementation of the predictive state space control algorithm, called Model Predictive Control (MPC). This control algorithm is verified on the Ball and Plate laboratory model, called B&P_KYB, for the reference trajectory tracking. The control algorithm is first verified using the derived nonlinear simulation model in Matlab/Simulink. Since simulation results are acceptable, an experiment is realized on the real laboratory model. The results of the experiment are demonstrated as the time response of the ball position and the voltage.**

Keywords— model predictive control, MPC, mechatronic system, Ball and Plate

I. Introduction

Model Predictive Control (MPC) is a modern method of nonlinear dynamic system control. MPC algorithm design for the nonlinear dynamic systems is discussed in publications such as [1], [2]. The application of the MPC algorithm to the different models of the dynamic systems is presented e.g. in articles [3], [4].

This papers presents the verification of the designed MPC control algorithm of the laboratory model *B&P_KYB* (Ball and plate system), which is located in the Laboratory of the Mechatronic Systems V142, Department of Cybernetics and Artificial Intelligence (DCAI), Faculty of Electrical Engineering and Informatics, Technical University of Košice, (*http://kyb.fei.tuke.sk/ laboratoria/miest/V142.php*).

At the DCAI, predictive control algorithms have been used for the control of laboratory models Helicopter *Humusoft CE 150* [7] and Hydraulic system [8]. Another laboratory model at the DCAI is the Ball and Plate model, *Humusoft CE 151*. Implementation of the MPC and other optimal state space algorithms to Ball and Plate model *Humusoft CE 151* was unsuccessful, which was caused by inaccuracies in the design of the mathematical model this system. The main objective of these papers is to design and to verify the MPC algorithm using the laboratory model *B&P_KYB* (ball and plate model) [11].

The components and construction of the laboratory model *B&P_KYB* are different from those of the model Ball and Plate - *Humusoft CE 151*. A web camera is used to capture the ball position in the laboratory model *B&P_KYB* and the algorithm for image processing is designed and implemented to acquire the image. The tilt of the plate is controlled by two servomotors, which communicate with control PC by single – microchip computer (more information about differences from the model Ball and Plate *Humusoft CE 151* can be found in [12]). Mathematical description of the *B&P_KYB* model is based on the analytical identification with respect of physical laws (experimental identification of the *B&P_KYB* model is presented in [11]). The exact mathematical model enables to implement state space control algorithms, such as the MPC algorithm for the laboratory model *B&P_KYB*.

The designed MPC algorithm is verified using the nonlinear simulation model *B&P_KYB* in a control experiment for the reference trajectory tracking. After the required results achieved at the nonlinear simulation model *B&P_KYB*, was the MPC algorithm verified on the real laboratory model. The results of the simulations are presented in the time response form for the chosen variables, such as the ball position and reference angle of the laboratory model.

II. Model Based Predictive Control Algorithm with Linear Predictor

Model Predictive Control (MPC) is the state predictive control algorithm, suitable for the control of the fast. unstable dynamic systems in the SISO or MIMO form [6]. This paper focuses on the implementation of the MPC algorithm for SISO systems.

In general, predictive control algorithms minimize the following functional:

$$J_{MPC} = \sum_{i=1}^{N_p} Q(i)[\hat{y}(k+i) - w(k+i)]^2 + \sum_{i=1}^{N_u} R(i)[u(k+i-1)]^2 , \quad (1)$$

where Q, R are weight matrices, N_p is horizon of the prediction and N_u is horizon of the control, k is number of sample (k = 1, 2, 3, ...).

The presented MPC algorithm uses a linear predictor for computing the prediction of state variables on the horizon N_p, according to [5]. To derivate of the linear predictor, it is important to specify a model of the dynamic system. If the model of the dynamic system is obtained by analytical identification, the result of the identification is the system of

978-1-4799-8221-9/15/$31.00 ©2015 IEEE

the nonlinear differential equations. Next, the linear predictor is defined by linearizing the nonlinear differential equations. The result of the linearization is the linear model of the nonlinear dynamic system in the state space form:

$$\begin{aligned} x(k+1) &= F\,x(k) + Gu(k) \\ y(k) &= C^T x(k) \end{aligned}, \tag{2}$$

while F is matrix of system dynamics (n_x x n_x), G is input matrix (n_x x n_u) and C^T is output matrix (n_y x n_x).

The derivation of the system states predictor and output predictor with respect to the prediction horizon N_p is described in [5], [6]. Linear predictor in matrix form based on the state space description of the system (2) is defined as:

$$\hat{y} = V_0\,x(k) + S_0\,u_k\,, \tag{3}$$

where the vector of predicted values is $\hat{y} = \begin{bmatrix} y(k) & y(k+1) & \cdots & y(k+N_p-1) \end{bmatrix}^T$, vector of control predicted values is $u_k = \begin{bmatrix} u(k) & u(k+1) & \cdots & u(k+N_p-1) \end{bmatrix}^T$ and the vector of the reference trajectory from step k is $w_k = \begin{bmatrix} w(k) & w(k+1) & \cdots & w(k+N_p-1) \end{bmatrix}^T$.

In the definition of the linear predictor in the matrix form (3), $V_0x(k)$ represents the free response of the system and S_0u_k represents the forced response of the system.

The matrices of free response V_0 and forced response S_0 of the system have form:

$$V_0 = \begin{pmatrix} C \\ CF \\ \vdots \\ CF^{N_p} \end{pmatrix}, \quad S_0 = \begin{pmatrix} D & 0 & \cdots & & 0 \\ CG & D & & & \\ CFG & CG & D & & \vdots \\ \vdots & & & \ddots & 0 \\ CF^{N_p-1}G & \cdots & CFG & CG & D \end{pmatrix}$$

If predictor (3) is substituted into the functional (1), we obtain the predictor in form:

$$\begin{aligned} J_{MPC} = {} & (V_0\,x(k) + S_0\,u_k - w_k)^T Q (V_0\,x(k) + S_0\,u_k - w_k) + \\ & + u_k^T R u_k \end{aligned} \tag{4}$$

Minimization of functional (4) with the condition $\frac{\partial J_{MPC}}{\partial u_k} \overset{!}{=} 0$ based on the derivation of the vectors according to [6], yields the optimal control law in the final form:

$$u_k = -(G^T Q G + R)^{-1} (G^T Q (Vx(k) - w_k))$$

Optimal control law u_k for the *B&P_KYB* laboratory model is computed by optimalization methods based on quadratic programming, which are implemented in the *Matlab* enviroment as the function *quadprog* (*Optimization Toolbox*).

Optimal control law u_k with constraints is given by the minimization of the relation:

$$\min_{\bar{u}_k} \left(\frac{1}{2} u_k^T H u_k + g^T u_k\right), \quad u_{\min} \le u_k \le u_{\max}\,, \tag{5}$$

while H (Hessian) and g (gradient) are defined as follows:

$$H = S_0^T Q S_0 + R \qquad g^T = (y_k - w_k)^T Q\,S_0 \tag{6}$$

The value of the control horizon N_u affects the size of the free response matrix S_0 if the matrix S_0 is multiplied from the right by the matrix:

$$I_S = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & & & \\ 1 & & & \\ 0 & 1 & & \vdots \\ & \vdots & & \\ & 1 & & \\ \vdots & & \ddots & 0 \\ & & & 1 \\ & & & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

The size of matrix I_S is $(m_u \cdot N_u) \times (m_u \cdot N_p)$, where m_u is the length of control vector u_k.

The equations (5), (6) are fundamental for the design of MPC algorithm with linear predictor which computes the control input for the dynamic system. The general scheme of the MPC algorithm is shown on Fig. 1.

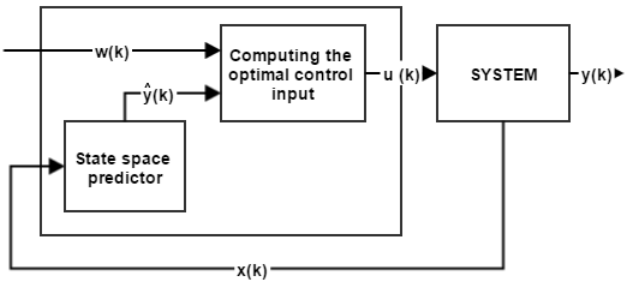

Fig. 1. Block scheme of the MPC algorithm

III. Implementation of Model Predictive Control to Laboratory Model B&P_KYB

The nonlinear simulation model of the *B&P_KYB* system (Fig. 2) was obtained by analytical identification based on relevance law of physics.

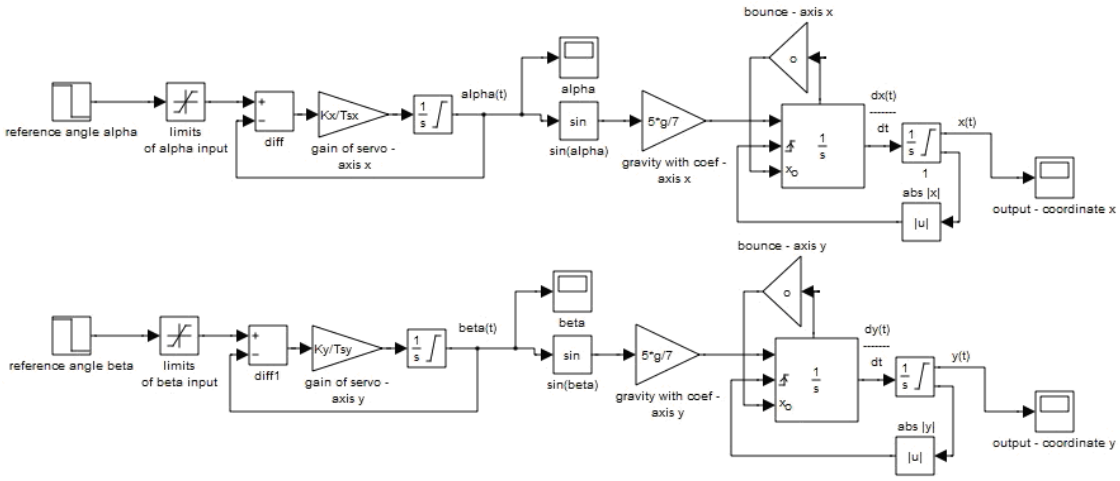

Fig. 2. Block scheme of the simulation *B&P_KYB* model in Simulink

The model was broken down into two subsystems, as the *Servomotors* subsytem and the *B&P* subsystem, shown in Fig. 3.

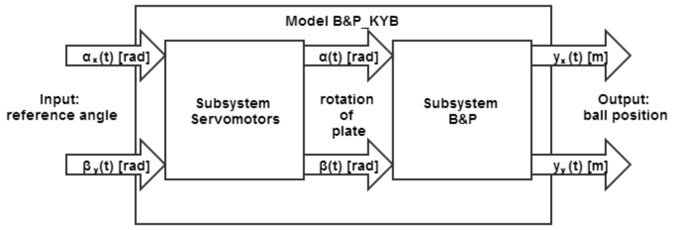

Fig. 3. Block scheme of the mechatronical laboratory model *B&P_KYB*

B&P subsystem is described by two nonlinear differential equations. Nonlinear differential equation for axis x is:

$$\ddot{y}_x(t) = \frac{5}{7} g \sin\alpha(t) \tag{7}$$

and for axis y:

$$\ddot{y}_y(t) = \frac{5}{7} g \sin\beta(t) \tag{8}$$

Subsystem *Servomotors* is described by two linear differential equations. Linear differential equation for x axis is:

$$\dot{\alpha}(t) = \frac{K_x}{T_{sx}} \cdot (\alpha_x(t) - \alpha(t)) \tag{9}$$

and for y axis is:

$$\dot{\beta}(t) = \frac{K_y}{T_{sy}} \cdot (\beta_y(t) - \beta(t)) \tag{10}$$

Physical variables and parameters of the nonlinear model are listed in Table 1.

TABLE I. PARAMETERS AND PHYSICAL VARIABLES

	Description	*Label*	*Units*
physical variables	ball position – x axis	$y_x(t)$	$[m]$
	ball position – y axis	$y_y(t)$	$[m]$
	plate tilt – x axis	$\alpha(t)$	$[rad]$
	reference angle – x axis	$\alpha_x(t)$	$[rad]$
	plate tilt – y axis	$\beta(t)$	$[rad]$
	reference angle – y axis	$\beta_y(t)$	$[rad]$
parameters	servomotor gain – x axis	K_{sx}	-
	servomotor time constant – x axis	T_{sx}	-
	servomotor gain – y axis	K_{sy}	-
	servomotor time constant – y axis	T_{sy}	-

The time responses to the same step input signal for the nonlinear simulation and real laboratory model are compared in Fig. 4.

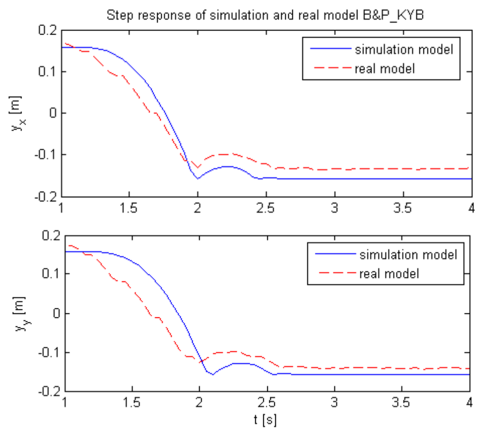

Fig. 4. Comparison of the simulation and real model open loop response

Comparison of models responses depicted in Fig. 4 shows that the behavior of the nonlinear simulation model is very similar to the laboratory model of *B&P_KYB*. This assumption allows us to use the *B&P_KYB* simulation model for verification of the MPC control algorithm.

To define the linear predictor, it is important to specify the linear state space form (3) of the *B&P_KYB* simulation model. The vector of the state variables is specified as (axis x):

$$\bar{x}(t)=\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}=\begin{bmatrix} y_x(t) \\ \dot{y}_x(t) \\ \alpha(t) \end{bmatrix}=\begin{bmatrix} position\,[m] \\ velocity\,[m\cdot s^{-1}] \\ angle\,[rad] \end{bmatrix}$$

The input vector has the form:

$$u(t)=[\alpha_x(t)]$$

The selected linearization point of the system is equivalent to the equilibrium point, whose values are $x_{XEP}=[x_1=0 \quad x_2=0 \quad x_3=0]$. The matrix of system dynamics could be defined by the Jacobi matrix:

$$A_x=\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}_{x_{XEP}}=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & k_g g \\ 0 & 0 & -\frac{K_x}{T_{sx}} \end{bmatrix}_{x_{XEP}}, \quad k_g=\frac{5}{7}$$

The input matrix is defined as:

$$B_x=\begin{bmatrix} \frac{\partial f_1}{\partial u} \\ \frac{\partial f_2}{\partial u} \\ \frac{\partial f_3}{\partial u} \end{bmatrix}_{x_{XEP}}=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{x_{XEP}}$$

The process of defining the state space description of the system for the direction of the y axis is the same for the x axis. The output matrices for both directions are specified as $C_x^T=C_y^T=[1 \quad 0 \quad 0]$ [12].

Synthesis of the MPC control algorithm with a linear predictor is based on the discrete state space form (2). Transformation from the continuous (matrices A, B) to the discrete form (matrices F, G) is done by the Matlab function *c2d* with the sample period $T_s = 0.05$ *s*.

IV. Design of MPC Algorithm for Laboratory Model B&P_KYB

The design of the MPC algorithm for the laboratory model *B&P_KYB* is described in the flowchart form, in Fig. 5.

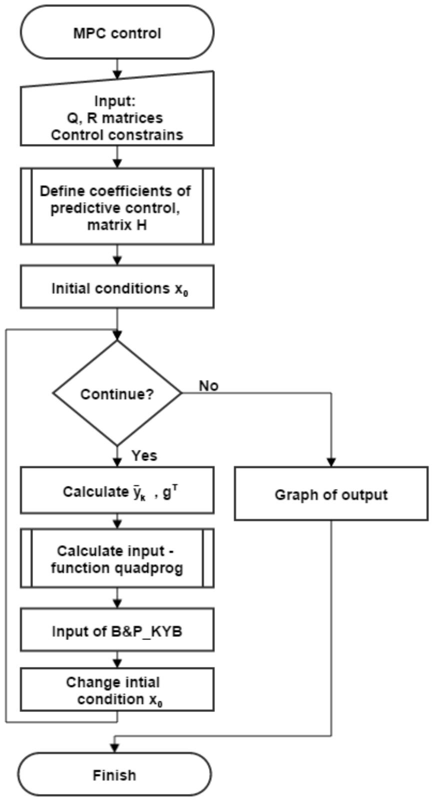

Fig. 5. The designed flow chart of the MPC algorithm

According to the flowchart (Fig. 5), MPC algorithm has been implemented as the *paramMpcC* function in the *Matlab* environment. The description of input and output parameters of the *paramMpcC* function is as follows:

[H, gT, S, V, A_obm, dU, options] = ***paraMpcC****(F, G, C, D, Q, R, Np, Nu, k)*

H – Hessian matrix specified in equation (6)

gT – gradient matrix specified in equation (6)

S – matrix of forced response (S_0)

V – matrix of free response (V_0)

A_obm – matrix of the model constraints

dU – minimum of the control different

options – options for *quadprog* function

F – matrix of system dynamics in the discrete form

G – matrix of system input in the discrete form

C – transposed matrix of the system output

D – feedforward matrix

Q – weight matrix for state variables in functional

R – weight matrix for system input in functional

Np – value of the prediction horizon

Nu – value of the control horizon

k – vector of the model parameters

The *paramMpcC* function can be used for computing the Hessian *H*, gradient *g*, free response matrix (V_0) and forced response matrix (S_0). The *paramMpcC* function returns all matrices and parameters which are important for the function *quadprog*.

The designed MPC algorithm is used for predictive control design of the *B&P_KYB* laboratory model. The control algorithm is verified for the circle or square reference trajectory tracking including constraints. The MPC control structure for the simulation is shown in Fig. 6. Simulation parameters are listed in Table II. Simulation results for reference trajectory tracking are illustrated in Fig. 8, Fig. 10. The result of the simulation shows that the MPC algorithm fulfills the requirements of the control quality for the chosen objective. The MPC algorithm was compared to the optimal state control algorithm (LQ) and has shown better results of the simulation for the reference trajectory tracking than the LQ algorithm. The simulation results of LQ algorithm are listed in [11]. Fig. 9 and Fig. 11 show that MPC control of the laboratory model *B&P_KYB* is smooth, which is very important for the criteria of quality.

TABLE II. SIMULATION PARAMETERS

Description	***Label***	***Value***	***Units***
simulation time	*T*	20	[*s*]
sample period	*dT*	0,05	[*s*]
prediction horizon	*Np*	20	samples
control horizon	*Nu*	1	samples

The comparison of the effect of MPC control algorithm on the nonlinear simulation and laboratory model *B&P_KYB* in simulation with the MPC is illustrated in Fig. 10. The MPC algorithm is very suitable for the mechatronic systems with the fast dynamics.

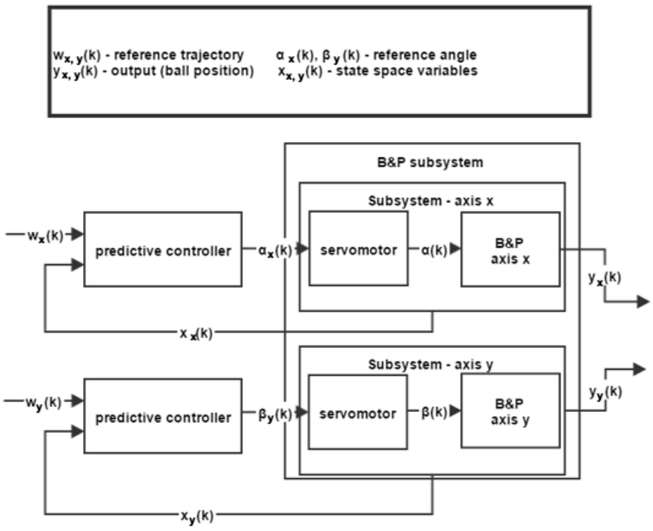

Fig. 6. Control scheme for reference trajectory tracking with using the MPC algorithm

To improve the experiments with the nonlinear simulation laboratory model *B&P_KYB*, we developed GUI application called *VirtModelsKKUI*, shown in Fig. 7. This application includes different types of control algorithms together with a developed 3D virtual laboratory model [13].

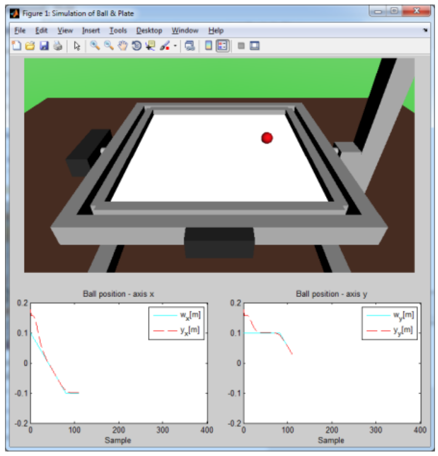

Fig. 7. GUI of the created application *VirtModelsKKUI*

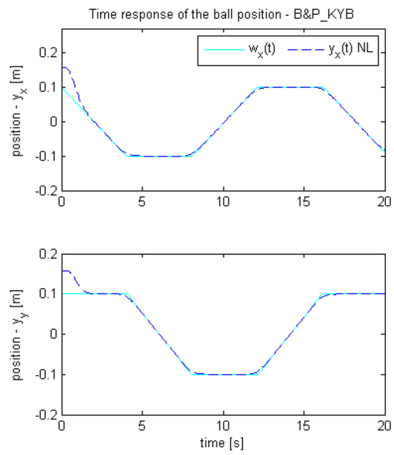

Fig. 8. Response of the ball position - square trajectory tracking

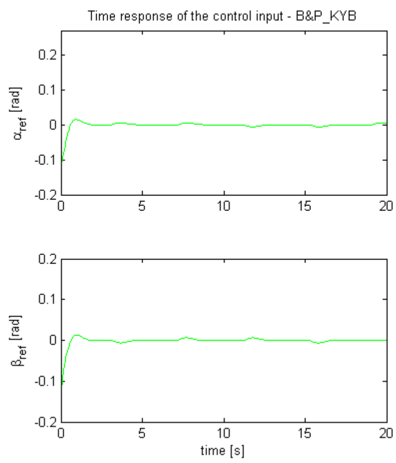

Fig. 9. Response of the control input - square trajectory tracking

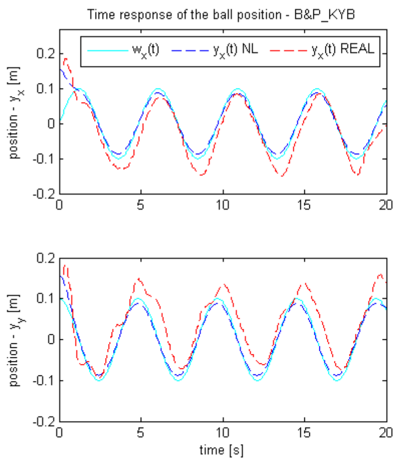

Fig. 10. Comparison of the nonlinear simulation (NL) and laboratory model (REAL) reference trajectory tracking – circle trajectory

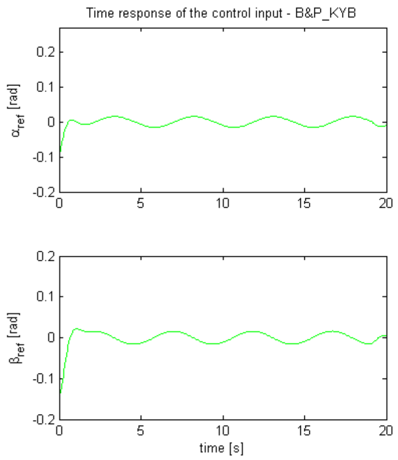

Fig. 11. Response of the control input - circle trajectory tracking (nonlinear simulation model)

V. Conclusion

The presented MPC algorithm has been implemented in the *Matlab* environment and verified for the *B&P_KYB* laboratory model control.

Designed predictive control algorithm for the laboratory model *B&P_KYB* is used for the reference trajectory tracking. Circle and square was chosen reference trajectories. The simulation results of the nonlinear simulation laboratory model have the required quality. Control is smooth, without strong oscillations, which is very important for the lifetime of servomotors. Used MPC control algorithm had a very good quality in control of the real laboratory model *B&P_KYB*, too. It can be concluded that the MPC control algorithm is a suitable predictive control algorithm for this type of dynamic systems.

Some of the results of the MPC algorithm design and implementation with application *VirtModelsKKUI* will be use in the subject *Control & Artificial Intelligence* (web page: *http://matlab.fei.tuke.sk/raui*).

Acknowledgment

This contribution is the result of the Vega project implementation – Dynamic Hybrid Architectures of the Multi-agent Network Control Systems (No. 1/0286/11), supported by the Scientific Grant Agency of Slovak Republic – 50%, and of the Research and Development Operational Program for project: University Science Park Technicom for innovative applications with knowledge technology support, (ITMS code 26220220182), co-financed by the ERDF – 50%.

References

[1] Camacho E.F., Bordons C., “Model Predictive Control”, 2. vyd, 2004. ISBN 978-0-85729-398-5.

[2] Morari M., Lee J.H., García C. E., “Model Predictive Control”, 2002. online: <http://s3.amazonaws.com/academia.edu.documents/31077928/knihaucebmpc.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1416305823&Signature=6aI4eMyKZ6ikj8tiJwd9ojwxuuI%3D&response-content-disposition=inline>

[3] Hara N., Takahashi M., Konishi K., “Experimental Evaluation of Model Predictive Control of Ball and Beam Systems”, American Control Conference, 2011.

[4] Chalupa P., Novák J., “Modeling and model predictive control of a nonlinear hydraulic system”, Computers & Mathematics with Applications Journal, , pp. 155–164, Volume 66, Issue 2, 2013.

[5] Roubal J., Havlena, V., “Range Control MPC Approach For Two-Dimensional System”, Proceedings of the 16tg IFAC World Congress, Volume 16, part 1, 2005.

[6] Roubal J., “Prediktivní regulátor. Příklady ze cvičení – Moderní teorie řizení” (in Czech). Online: <http://support.dce.felk.cvut.cz/pub/roubalj/teaching/MTR/seminars/MTR_cv8_mpc.pdf>

[7] Jajčišin Š., Jadlovská A., “Predictive Control Algorithms Verification on the Laboratory Helicopter Model”, Acta Polytechnica Hungarica (Journal of Applied Sciences), Vol. 9, No. 4, 2012. ISSN 1785-8860. Online: <http://www.uni-obuda.hu/journal/Jadlovska_Jajcisin_36.pdf>.

[8] Jajčišin Š., Jadlovská A., “Adaptive Predictive Control on the basis of Analytic and Experimental Identification of Controlled System”, Electrical Engineering and Informatics 3 : Proceeding of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice. Košice: TU – FEI, 2012. ISBN 978-80-553-0890-6.

[9] Jadlovská A., Jajčišin Š., “Návrh algoritmov prediktívneho riadenia s využitím nelineárnych modelov fyzikálnych systémov”, 1. vyd. Košice, ELFA, 2013. ISBN 978-80-8086-229-9.

[10] Krokavec D., Filasová A., “Discrete systems”. Košice, ELFA, 2006. ISBN 80-8086-028-9.

[11] Oravec M., “Control of Mechatronical Systems using .NET Applications”, Master thesis, pp. 112 (Supervisor: doc. Ing. Anna Jadlovská, PhD). Košice: TU – FEI, 2014.

[12] Oravec M., “Optimal State Control of the Mechatronical Laboratory Model B&P_KYB”, Proceedings of the Faculty of Electrical Engineering and Informatics of the Technical University of Košice, 2014. ISBN 978-80-553-1704-5

[13] Oravec M., “Design of Virtual Models of Mechatronics Systems with Simulink 3D Animation Toolobox”, Technical Computing, Bratislava, 2014.