DESIGN AND SIMULATION OF SELF-TUNING PREDICTIVE CONTROL OF TIME-DELAY PROCESSES

Vladimír Bobál[1,2], Marek Kubalčík[2] and Petr Dostál[1,2]
Tomas Bata University in Zlín
[1]Centre of Polymer Systems, University Institute
[2]Department of Process Control, Faculty of Applied Informatics
T. G. Masaryka 5555
760 01 Zlín
Czech Republic
E-mail: bobal@fai.utb.cz

KEYWORDS

Time-delay systems, Model Predictive Control, CARIMA model, Quadratic criterion, Simulation

ABSTRACT

Majority industrial processes such as thermal, chemical biological, metallurgical, plastic etc., have time-delays. Therefore, the problem of the identification and optimal control of such systems is of great importance. These time-delay processes can be effectively handled by the Model-based Predictive Control method. The paper deals with design of an algorithm for self-tuning predictive control of such processes. The self-tuning principle is one of possible approaches to control of nonlinear systems or systems with uncertainties. Three types of processes were chosen for simulation verification of the designed self-tuning predictive controller. The program system MATLAB/SIMULINK was used for testing and verification of this predictive controller.

INTRODUCTION

Time delay is very often encountered in various technical systems, such as electric, pneumatic and hydraulic networks, chemical processes, long transmission lines, robotics, etc. The existence of pure time lag, regardless if it is present in the control or/and the state, may cause undesirable system transient response, or even instability. Consequently, the problem of controllability, observability, robustness, optimization, adaptive control, pole placement and particularly stability and robustness stabilization for this class of systems, has been one of the main interests for many scientists and researchers during the last five decades.

For control engineering, such processes can often be approximated by the FOTD (First-Order-Time-Delay) model. Time-delay in a process increases the difficulty of controlling it. However the approximation of higher-order process by lower-order model with time-delay provides simplification of the control algorithms. When high performance of the control process is desired or the relative time-delay is very large, the predictive control strategy is one of possible approaches. The predictive control strategy includes a model of the process in the structure of the controller. The first time-delay compensation algorithm was proposed by (Smith 1957). This control algorithm known as the Smith Predictor (SP) contained a dynamic model of the time-delay process and it can be considered as the first model predictive algorithm. First versions of Smith Predictors were designed in the continuous-time modifications, see e.g (Normey-Rico and Camacho 2007). Because most of modern controllers are implemented on digital platforms, the discrete versions of the time-delay controllers are more suitable for time-delay compensation in industrial practice. Most of authors designed the digital time-delay compensators with fixed parameters. However, the time-delay compensators are more sensitive to process parameter variations and therefore require an auto-tuning or adaptive (self-tuning) approach in many practical applications. Two adaptive modifications of the digital Smith Predictors are designed in (Hang et al. 1989; Bobál et al. 2011) and implemented into MATLAB/SIMULINK Toolbox (Bobál et al. 2012a; Bobál et al. 2012b).

Model predictive control (MPC) is becoming increasable popular method in industrial process control where time-delays are component parts of the system. However, an accurate appropriate model of the process is required to ensure the benefits of MPC. Furthermore, perturbations of a time-delay and parameters of an external linear model may induce complex behaviours (oscillations and instabilities) of the closed-loop system. Problems with time-variant model parameters can be solved using adaptive (self-tuning) approach.

MODEL PREDICTIVE CONTROL

Model Predictive Control, also known as Receding Horizon Control (RHC), attracts considerable research attention because of its unparalleled advantages.

These include (Lu 2008):

- Applicability to a broad class of systems and industrial applications.
- Computational feasibility.

Proceedings 27th European Conference on Modelling and Simulation ©ECMS Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang (Editors)
ISBN: 978-0-9564944-6-7 / ISBN: 978-0-9564944-7-4 (CD)

- Systematic approach to obtain a closed-loop control and guaranteed stability.
- Ability to handle hard constraints on the control as well as the system states.
- Good tracking performance.
- Robustness with respect to system modeling uncertainty as well as external disturbances.

The MPC strategy performs the optimization of a performance index with respect to some future control sequence, using predictions of the output signal based on a process model, coping with amplitude constraints on inputs, outputs and states. For a quick comparison of MPC and traditional control scheme, such as PID control, Fig. 1 shows the difference between the MPC and PID control schemes in which "anticipating the future" is desirable while a PID controller only has capacity of reacting to the past behaviours. The MPC algorithm is very similar to the control strategy used in driving a car (Lu 2008).

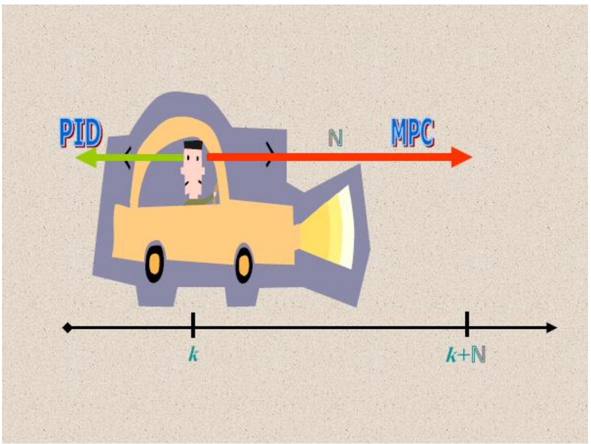

Figure 1: Difference between the MPC and PID control

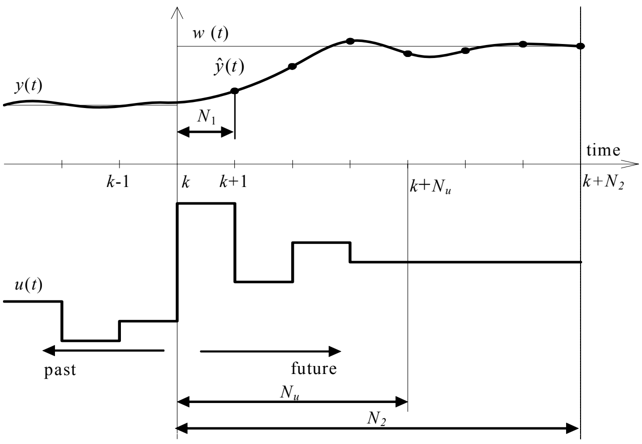

Figure 2: Principle of MPC

At current time k, the driver knows the desired reference trajectory for a finite control horizon, say (k, $k + N$), and by the taking into account the car characteristics to decide which control actions (accelerator, brakes, and steering) to take in order to follow the desired trajectory. Only the first control action is adopted as the current control law, and the procedure is then repeated over the next time horizon, say ($k + 1$, $k + 1+ N$). The term "*receding horizon*" is introduced, since the horizon recedes as time proceeds. The basic MPC strategy is shown in Fig. 2, where $u(t)$ is the manipulated variable, $y(t)$ is the process output and $w(t)$ is the reference signal, N_1, N_2 and N_u are called minimum, maximum and control horizons, respectively. The block diagram is shown in Fig. 3.

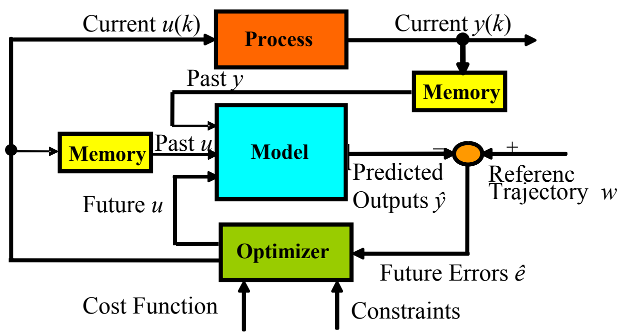

Figure 3: Block diagram of MPC

Calculation of Optimal Control

The designed control algorithm is based on the Generalised Predictive Control (GPC) method (Clarke et al. 1987a,b). The standard cost function used in GPC contains quadratic terms of control error and control increments on a finite horizon into the future (Camacho and Bordons 2004; Mikleš and Fikar 2008)

$$J = \sum_{i=N_1}^{N_2} \left[\hat{y}(k+i) - w(k+i)\right]^2 + \sum_{i=1}^{N_u} \left[\lambda(i)\Delta u(k+i-1)\right]^2 \tag{1}$$

where $\hat{y}(k+i)$ is the process output of i steps in the future predicted on the base of information available upon the time k, $w(k+i)$ is the sequence of the reference signal and $\Delta u(k+i-1)$ is the sequence of the future control increments that have to be calculated. Parameters N_1, N_2 and N_u are called minimum, maximum and control horizons. The parameter $\lambda(i)$ can be generally a sequence which affects future behaviour of the controlled process. The output of the model (prediction) is computed as the sum of the forced response $\boldsymbol{y}_n$ and the free response $\boldsymbol{y}_0$

$$\hat{\boldsymbol{y}} = \boldsymbol{y}_n + \boldsymbol{y}_0 \tag{2}$$

The free response is that part of the prediction which is determined by past values of the manipulated variable and past values of the systems output. The forced response is determined by future increments of the manipulated variable. The forced response is computed as the multiplication of the matrix $\boldsymbol{G}$ (Jacobian Matrix of the model) and the vector of future control increments $\Delta\boldsymbol{u}$, which is generally a priori unknown

$$\boldsymbol{y}_n = \boldsymbol{G}\Delta\boldsymbol{u} \tag{3}$$

where

$$\boldsymbol{G} = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\ g_2 & g_1 & 0 & \cdots & 0 \\ g_3 & g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & g_{N_2-2} & \cdots & g_{N_2-N_u+1} \end{bmatrix} \tag{4}$$

is matrix containing values of the step sequence.
It follows from (2) and (3) that the predictor in a vector form is given by

$$\hat{\boldsymbol{y}} = \boldsymbol{G}\Delta\boldsymbol{u} + \boldsymbol{y}_0 \tag{5}$$

By assumption that λ is scalar, the cost function (1) can be modified to the form

$$\begin{aligned} J &= \left(\hat{\boldsymbol{y}} - \boldsymbol{w}\right)^T \left(\hat{\boldsymbol{y}} - \boldsymbol{w}\right) + \lambda\Delta\boldsymbol{u}^T\Delta\boldsymbol{u} = \\ &= \left(\boldsymbol{G}\Delta\boldsymbol{u} + \boldsymbol{y}_0 - \boldsymbol{w}\right)^T \left(\boldsymbol{G}\Delta\boldsymbol{u} + \boldsymbol{y}_0 - \boldsymbol{w}\right) + \lambda\Delta\boldsymbol{u}^T\Delta\boldsymbol{u} \end{aligned} \tag{6}$$

Minimisation of the cost function (6) now becomes a direct problem of linear algebra. The solution in an unconstrained case can be found by setting partial derivative of J with respect to $\Delta\boldsymbol{u}$ as zero and yields

$$\Delta\boldsymbol{u} = \left(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{G}^T\left(\boldsymbol{w} - \boldsymbol{y}_0\right) \tag{7}$$

Equation (7) gives the whole trajectory of the future control increments and such is an open-loop strategy. To close the loop, only the first element is applied to the system and the whole algorithm is recomputed at time k+1. If we denote the first row of the matrix $\left(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{G}^T$ as $\boldsymbol{K}$ then the actual control increment can be calculated as

$$\Delta u(k) = \boldsymbol{K}\left(\boldsymbol{w} - \boldsymbol{y}_0\right) \tag{8}$$

DERIVATION OF PREDICTOR

An important task is computation of predictions for arbitrary prediction and control horizons. Dynamics of most of processes requires horizons of length where it is not possible to compute predictions in a simple straightforward way. Recursive expressions for computation of the free response and the matrix $\boldsymbol{G}$ in each sampling period had to be derived. There are several different ways of deriving the prediction equations for transfer function models. Some papers make use of Diophantine equations to form the prediction equations (Kwon et al. 2002) In (Rossiter 2003) matrix methods are used to compute predictions. We derived a method for recursive computation of both the free response and the matrix of the dynamics (Kubalčík and Bobál 2011; Bobál et al. 2010).

Computation of the predictor for the time-delay system can be obtained by modification of the predictor for the corresponding system without a time-delay. At first we will consider the second order system without time-delay and then we will modify the computation of predictions for the time-delay system.

Second Order System without Time Delay

The deterministic model is described by the discrete transfer function

$$G\left(z^{-1}\right) = \frac{B\left(z^{-1}\right)}{A\left(z^{-1}\right)} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{9}$$

Model (9) can be also written in the form

$$A\left(z^{-1}\right)y(k) = B\left(z^{-1}\right)u(k) \tag{10}$$

A widely used model in GPC is the CARIMA model which can be obtained from the nominal model (10) by adding a disturbance model

$$A\left(z^{-1}\right)y(k) = B\left(z^{-1}\right)u(k) + \frac{C\left(z^{-1}\right)}{\Delta}n_c(k) \tag{11}$$

where $n_c(k)$ is a non-measurable random disturbance that is assumed to have zero mean value and constant covariance and $\Delta = 1 - z^{-1}$. Inverted Δ is then an integrator. The difference equation of the second order CARIMA model without the unknown term $n_c(k)$ can be expressed as

$$\begin{aligned} y(k) &= (1 - a_1)y(k-1) + (a_{1'} - a_2)y(k-2) \\ &\quad + a_2 y(k-3) + b_1\Delta u(k-1) + b_2\Delta u(k-2) \end{aligned} \tag{12}$$

It was necessary to compute three step-ahead predictions in a straightforward way by establishing of lower predictions to higher predictions. The model order defines that computation of one step-ahead prediction is based on three past values of the system output. The three step-ahead predictions after modifications can be written in a matrix form

$$\begin{aligned} \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{bmatrix} &= \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \\ g_3 & g_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} \\ &+ \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \\ \Delta u(k-1) \end{bmatrix} \end{aligned} \tag{13}$$

where the individual matrix elements in (13) are

$$g_1 = b_1; \quad g_2 = b_1(1 - a_1) + b_2;$$

$$g_3 = (a_1 - a_2)b_1 + (1 - a_1)^2 b_1 + (1 - a_1)b_2$$

$p_{11} = (1-a_1); \quad p_{12} = (a_1 - a_2);$
$p_{13} = a_2; \quad p_{14} = b_2;$
$p_{21} = (1-a_1)^2 + (a_1 - a_2); \quad p_{22} = (1-a_1)(a_1-a_2) + a_2;$
$p_{23} = a_2(1-a_1); \quad p_{24} = b_2(1-a_1);$
$p_{31} = (1-a_1)^3 + 2(1-a_1)(a_1-a_2) + a_2;$
$p_{32} = (1-a_1)^2(a_1-a_2) + a_2(1-a_1) + (a_1-a_2)^2;$
$p_{33} = a_2(1-a_1)^2 + (a_1-a_2)a_2; \quad p_{34} = b_2(1-a_1)^2 + (a_1-a_2)b_2.$

It is possible to divide computation of the predictions to recursion of the free response and recursion of the matrix of the dynamics. Based on the three previous predictions it is repeatedly computed the next row of the free response matrix in the following way:

$$\begin{aligned} p_{41} &= (1-a_1)p_{31} + (a_1-a_2)p_{21} + a_2 p_{11} \\ p_{42} &= (1-a_1)p_{32} + (a_1-a_2)p_{22} + a_2 p_{12} \\ p_{43} &= (1-a_1)p_{33} + (a_1-a_2)p_{23} + a_2 p_{13} \\ p_{44} &= (1-a_1)p_{34} + (a_1-a_2)p_{24} + a_2 p_{14} \end{aligned} \tag{14}$$

The first row of the matrix is omitted in the next step and further prediction is computed based on the three last rows including the one computed in the previous step. This procedure is cyclically repeated. It is possible to compute an arbitrary number of rows of the matrix.

The recursion of the dynamics matrix is similar. The next element of the first column is repeatedly computed in the same way as in the previous case and the remaining columns are shifted to form a lower triangular matrix in the way which is obvious from the equation (13). This procedure is performed repeatedly until the prediction horizon is achieved. If the control horizon is lower than the prediction horizon a number of columns in the matrix is reduced. Computation of the new element is performed as follows:

$$g_4 = (1-a_1)g_3 + (a_1-a_2)g_2 + a_2 g_1 \tag{15}$$

Second Order System with Time-Delay

The nominal second order model with d steps of time-delay is considered as

$$G(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-d} \tag{16}$$

where d is a number of time-delay steps.

The CARIMA model for time-delay system without the unknown term $n_c(k)$ takes the form

$$\Delta A(z^{-1}) y(k) = z^{-d} B(z^{-1}) \Delta u(k) \tag{17}$$

In order to compute the control action it is necessary to determine the predictions from d+1 to d+N_2. The predictor (14) is then modified for an arbitrary number of time delay steps to

$$\begin{bmatrix} \hat{y}(k+3) \\ \hat{y}(k+4) \\ \hat{y}(k+5) \end{bmatrix} = \begin{bmatrix} p_{(1+d)1} & p_{(1+d)2} & p_{(1+d)3} \\ p_{(2+d)1} & p_{(2+d)2} & p_{(2+d)3} \\ p_{(3+d)1} & p_{(3+d)2} & p_{(3+d)3} \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \end{bmatrix} + \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \\ g_3 & g_2 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix}$$
$$+ \begin{bmatrix} g_{1+d-1} & g_{2+d-1} & p_{(1+d)4} \\ g_{2+d-1} & g_{3+d-1} & p_{(2+d)4} \\ g_{3+d-1} & g_{4+d-1} & p_{(3+d)4} \end{bmatrix} \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \Delta u(k-3) \end{bmatrix} \tag{18}$$

Recursive computation of the matrices is analogical to the recursive computation described for the second order system without time-delay.

RECURSIVE IDENTIFICATION PROCEDURE

The regression (ARX) model of the following form

$$y(k) = \boldsymbol{\Theta}^T(k)\boldsymbol{\Phi}(k) + n_c(k) \tag{19}$$

is used in the identification part of the designed controller predictive algorithm, where

$$\boldsymbol{\Theta}^T(k) = \begin{bmatrix} a_1 & a_2 & b_1 & b_2 \end{bmatrix} \tag{20}$$

is the vector of the parameters estimates and

$$\boldsymbol{\Phi}^T(k-1) = \begin{bmatrix} -y(k-1) & -y(k-2) & u(k-d-1) & u(k-d-2) \end{bmatrix} \tag{21}$$

is the regression vector. For calculating of the parameter estimates $\hat{\boldsymbol{\Theta}}(k)$ is utilized the recursive least squares method, its numerical stability is improved by means of LD decomposition and adaptation is supported by directional forgetting (Kulhavý 1987; Bobál et al. 2005).

SIMULATION VERIFICATION OF SELF-TUNING MPC

A simulation verification of the designed predictive algorithm was performed in MATLAB/SIMULINK environment. A typical control scheme, which was used, is depicted in Fig. 4. This scheme is used for systems with time-delay of two sample steps. Individual blocks of the Simulink scheme correspond to blocks of the general control scheme presented in Fig. 3. The controller block represents the controlled system. This block consists of the recursive identification, predictive and optimization parts. This block has two inputs (process output y_s and initial condition y_in) and three outputs (controller output u, generating of reference signal w and model parameter estimates $\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2$). It is possible to influence the output of the process by non-measurable variables – the white noise n_c and the step v disturbances.

The above mentioned predictive controller is not suitable for control of unstable processes. Therefore, three types of processes were chosen for simulation verification of digital self-tuning predictive controller algorithms. Consider the following continuous-time transfer functions:

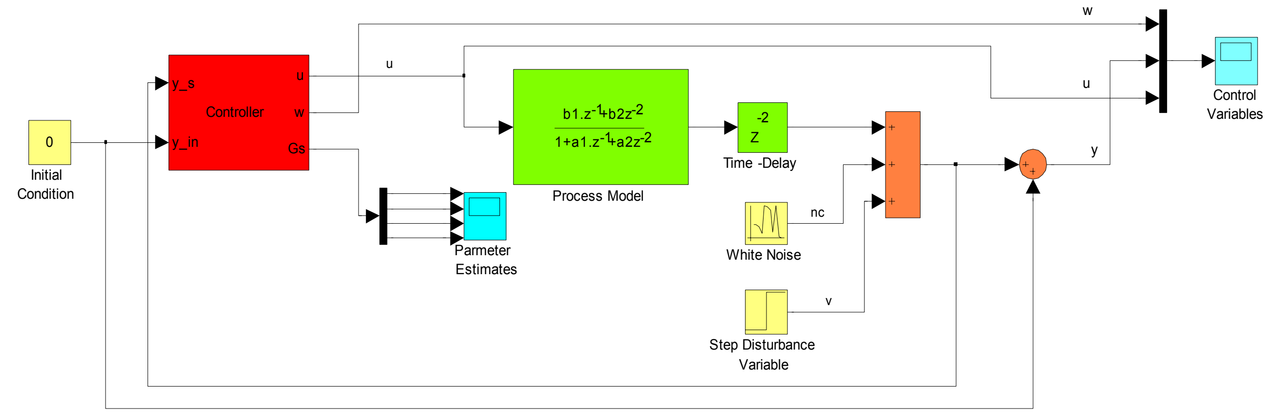

Figure 4: Simulink control scheme

1) Stable non-oscillatory $G_1(s)=\dfrac{2}{(s+1)(4s+1)}e^{-4s}$

2) Stable oscillatory $G_2(s)=\dfrac{2}{4s^2+2s+1}e^{-4s}$

3) Non-minimum phase $G_3(s)=\dfrac{2(1-5s)}{(s+1)(4s+1)}e^{-4s}$

Let us now discretize them with a sampling period $T_0 = 2\text{ s}$. The discrete forms of these transfer functions

$$G_1\left(z^{-1}\right)=\frac{0.4728z^{-1}+0.2076z^{-2}}{1-0.7419z^{-1}+0.0821z^{-2}}z^{-2}$$

$$G_2\left(z^{-1}\right)=\frac{0.6806z^{-1}+0.4834z^{-2}}{1-0.7859z^{-1}+0.3679z^{-2}}z^{-2}$$

$$G_3\left(z^{-1}\right)=\frac{-1.0978z^{-1}+1.7783z^{-2}}{1-0.7419z^{-1}+0.0821z^{-2}}z^{-2}$$

were used in the Simulink control scheme for the verification of the dynamical behavior of individual closed control loops.

Simulation control of model $G_1\left(z^{-1}\right)$

This model was chosen for the complex verification of self-tuning MPC properties for time-delay systems. The most important parameters in terms of quality process control are sampling period, penalization factor λ, initial parameter estimates $\hat{\boldsymbol{\Theta}}(0)$ (*a priori information*) and initial diagonal elements of the covariance matrix $\boldsymbol{C}_{ii}(0)$. The control performance is dependent also on the variance of the non-measurable noise σ^2.

At first, the model parameter estimates were chosen without *a priori* information

$$\hat{\boldsymbol{\Theta}}^T(0)=\begin{bmatrix}0.5 & 0.5 & 0.5 & 0.5\end{bmatrix}$$

$C_{ii}(0) = 1000$, $\sigma^2 = 0.01$, in time $t = 150$-300 s a step disturbance $v(t) = 5$ affected the systems output. Figs. 5 and 6 illustrate an influence of the penalization factor λ on the control performance.

The individual horizons were chosen for all experiments: $N_1 = 3$, $N_2 = 30$, $N_u = 28$.

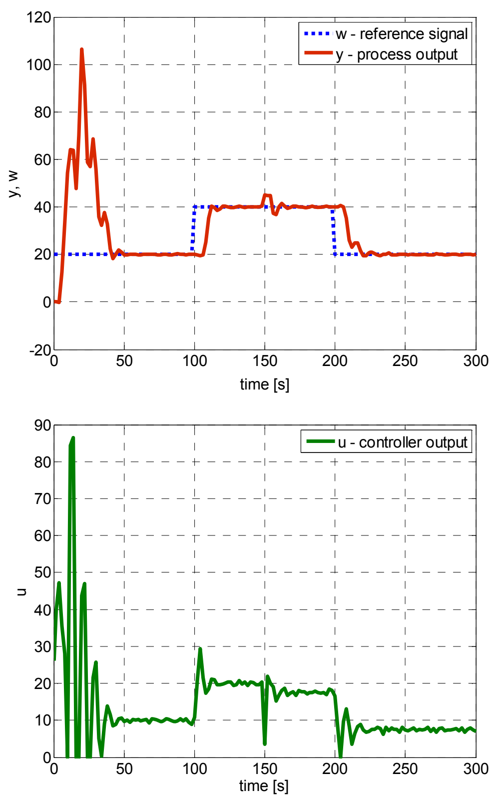

Figure 5: Control of the model $G_1\left(z^{-1}\right)$, $\lambda = 0.1$

The courses of the control variables oscillate in the initial control interval. When model parameter estimates are converged, the quality of the control process is very good. It is obvious that by increasing λ oscillations of the controller's output $u(k)$ are eliminated which has a positive influence on the course of the process output $y(k)$. In the subsequent

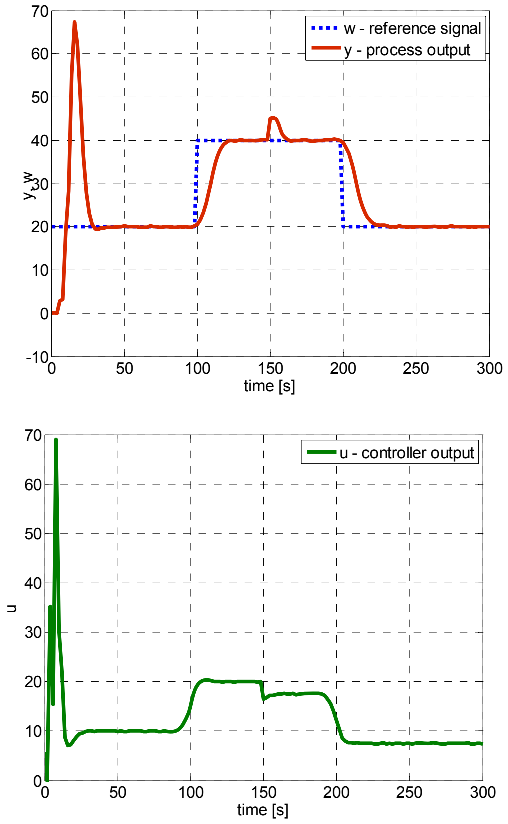

Figure 6: Control of the model $G_1\left(z^{-1}\right)$, $\lambda = 10$

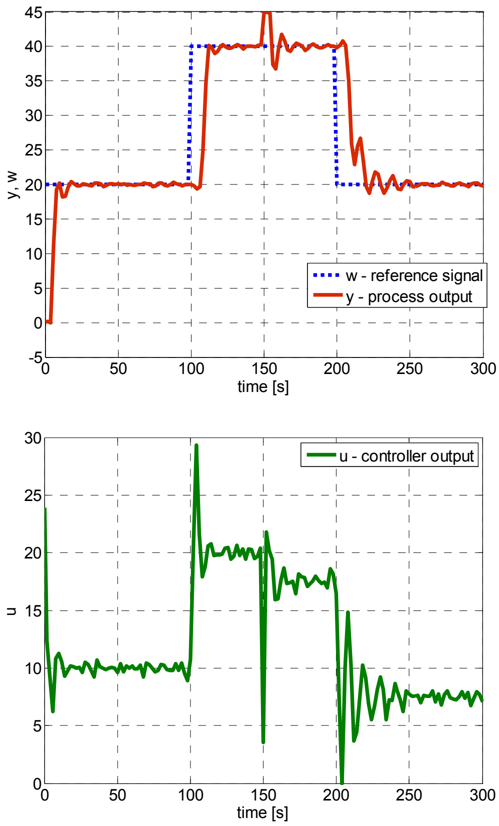

Figure 7: Control of the model $G_1\left(z^{-1}\right)$, $\lambda = 0.1$

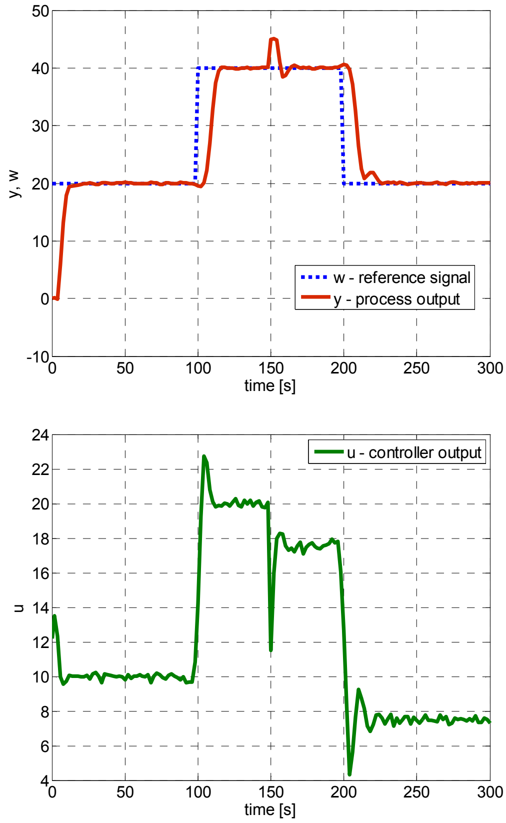

Figure 8: Control of the model $G_1\left(z^{-1}\right)$, $\lambda = 1$

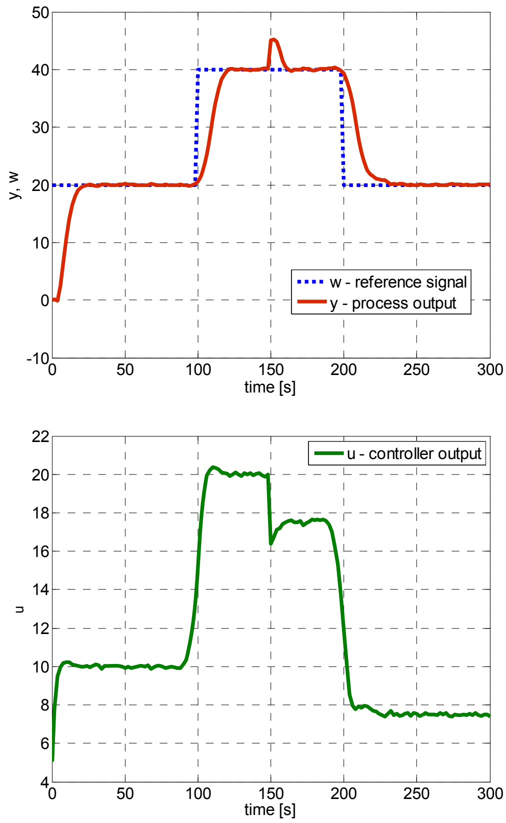

Figure 9: Control of the model $G_1\left(z^{-1}\right)$, $\lambda = 10$

experiment the model parameter estimates were chosen using *a priori* information

$$\hat{\boldsymbol{\Theta}}^T(0) = [-0.75 \quad 0.08 \quad 0.5 \quad 0.2]$$

$C_{ii}(0) = 10^{-3}$ (an assumption of the parameter estimates dispersion in a narrow interval). The other initial parameters were chosen to be the same as in the previous simulations – see Figs. 7 - 9. The courses of the control variables are satisfactory including the initial interval of control. The influence of the penalization factor λ is also evident on the control courses.

Simulation control of model $G_2(z^{-1})$

Simulation control of the model $G_2(z^{-1})$ (a stable oscillatory model) was realized upon similar conditions as in the previous case using *a priori* information. The initial vector of parameter estimates has the form

$$\hat{\boldsymbol{\Theta}}^T(0) = [-0.8 \quad 0.4 \quad 0.7 \quad 0.5]$$

The control courses are shown in Fig. 10, the quality of control is very good.

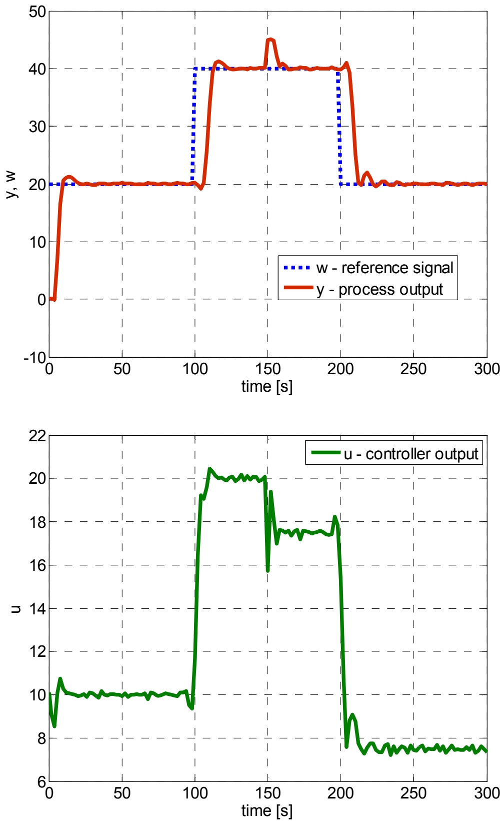

Figure 10: Control of the model $G_2(z^{-1})$, $\lambda = 1$

Simulation control of model $G_3(z^{-1})$

Simulation control of the model $G_3(z^{-1})$ (the non-minimum phase model) was realized upon similar conditions as in the previous cases using *a priori* information. The initial vector of parameter estimates has the form

$$\hat{\boldsymbol{\Theta}}^T(0) = [-0.75 \quad 0.08 \quad -1 \quad 1.8]$$

The control courses are shown in Fig. 11, the quality of control is very good.

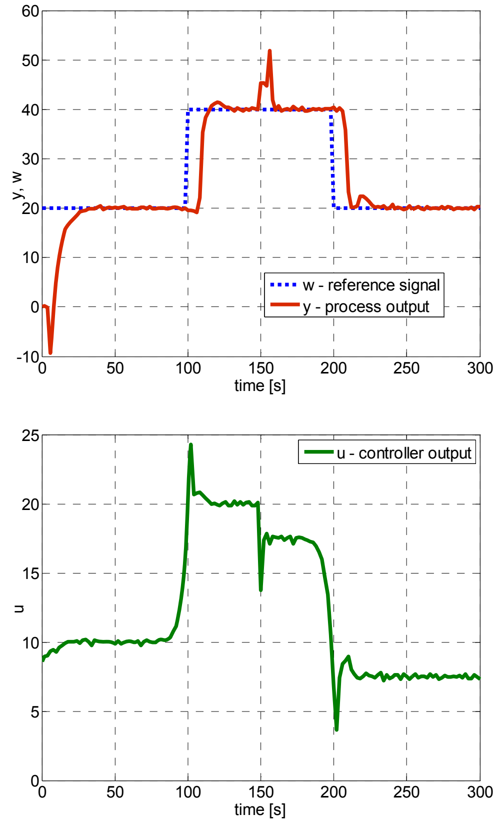

Figure 11: Control of the model $G_3(z^{-1})$, $\lambda = 1$

CONCLUSION

The contribution presents the self-tuning predictive control applied to time-delay processes. The predictive controller is based on the recursive computation of predictions by direct use of the CARIMA model. The computation of predictions was extended for time-delay systems. A linear model with constant coefficients used in pure model predictive control can not describe the control system in all its modes. Therefore, a self-tuning approach was applied. It consists of the recursive identification and the predictive controller. The model parameter estimates obtained from the identification procedure are used in the self-tuning predictive controller. MPC based on minimization of the quadratic criterion was derived and tested. Three models of control processes were used for simulation verification (the stable non-oscillatory, the stable oscillatory and the non-minimum phase). The designed predictive controller was successfully verified not only by simulation but also in real-time laboratory conditions for control of a heat exchanger.

ACKNOWLEDGMENT

This article was created with support of Operational Programme Research and Development for Innovations co-funded by the European Regional Development Fund (ERDF), national budget of Czech Republic within the framework of the Centre of Polymer Systems project (reg. number: CZ.1.05/2.1.00/03.0111).

REFERENCES

Bobál, V., Böhm, J., Fessl, J. and J. Macháček. 2005. *Digital Self-tuning Controllers: Algorithms, Implementation and Applications.* Springer-Verlag, London, 2005.

Bobál, V., Chalupa, P., Kubalčík, M. and P. Dostál. 2010. "Self-tuning predictive control of non-linear servomotor." *Journal of Electrical Engineering* 61, 265-372.

Bobál, V., Chalupa, P., Dostál, P. and M. Kubalčík. 2011. "Design and simulation verification of self-tuning Smith predictors." *International Journal of Mathematics and Computers in Simulation* 5, 342-351.

Bobál, V., Chalupa, P., Novák, J. and P. Dostál. 2012a. "MATLAB Toolbox for CAD of self-tuning of time-delay processes." In *Proc. of the International Workshop on Applied Modelling and Simulation*, Roma, 44 – 49.

Bobál, V., Chalupa, P. and J. Novák. 2012b. Toolbox for CAD and Verfication of Digital Adaptive Control Time-Delay Systems. Available from http://nod32.fai.utb.cz/promotion/Software_OBD/Time_Delay_Tool.zip.

Camacho, E. F. and C. Bordons. 2004. *Model Predictive Control.* Springer-Verlag, London.

Clarke, D. W., Mohtadi, C. and P. S. Tuffs. 1987a. "Generalized predictive control, part I: the basic algorithm." *Automatica* 23, 137-148.

Clarke, D. W., Mohtadi, C. and P. S. Tuffs. 1987b. "Generalized predictive control, part II: extensions and interpretations." *Automatica* 23, 149-160.

Hang, C. C., Lim, K. W. and B. W. Chong. 1989. "A dual-rate digital Smith predictor."*Automatica* 20, 1-16.

Kubalčík, M., and V. Bobál. 2011. "Techniques for predictor design in multivariable predictive control." *WSEAS Transactions on Systems and Control* 6, 349-360.

Kulhavý, R. 1987. "Restricted exponential forgetting in real time identification."*Automatica* 23, 586-600.

Kwon, W. H., Choj, H., Byun, D. G. S. Noh. 1992. "Recursive solution of generalized predictive control and its equivalence to receding horizon tracking control." *Automatica*, 28(6), 1235–1238.

Li, M. Ch. 2008. *Delay Identification and Model Predictive Control of Time Delayed Systems.* Ph.D. Thesis, Mc Gill University, Montreal, Canada.

Mikleš, J. and M. Fikar. 2008. *Process Modelling, Optimisation and Control.* Springer-Verlag, Berlin.

Normey-Rico, J. E. and E. F. Camacho. 2007. *Control of Dead-time Processes*, Springer-Verlag, London.

Rossiter, J. A. 2003. *Model Based Predictive Control: a Practical Approach.* CRC Press.

Smith, O.J. 1957. "Closed control of loops."*Chem. Eng. Progress*, 53, 217-219.

AUTHOR BIOGRAPHIES

VLADIMÍR BOBÁL graduated in 1966 from the Brno University of Technology, Czech Republic. He received his Ph.D. degree in Technical Cybernetics at Institute of Technical Cybernetics, Slovak Academy of Sciences, Bratislava, Slovak Republic. He is now Professor at the Department of Process Control, Faculty of Applied Informatics of the Tomas Bata University in Zlín, Czech Republic. His research interests are adaptive and predictive control, system identification and CAD for automatic control systems. You can contact him on email address bobal@fai.utb.cz.

MAREK KUBALČÍK graduated in 1993 from the Brno University of Technology in Automation and Process Control. He received his Ph.D. degree in Technical Cybernetics at Brno University of Technology in 2000. From 1993 to 2007 he worked as senior lecturer at the Faculty of Technology, Brno University of Technology. From 2007 he has been working as an associate professor at the Department of Process Control, Faculty of Applied Informatics of the Tomas Bata University in Zlín, Czech Republic. Current work cover following areas: control of multivariable systems, self-tuning controllers, predictive control. His e-mail address is: kubalcik@fai.utb.cz.

PETR DOSTÁL studied at the Technical University of Pardubice, Czech Republic, where he obtained his master degree in 1968 and PhD. degree in Technical Cybernetics in 1979. In the year 2000 he became professor in Process Control. He is now head of the Department of Process Control, Faculty of Applied Informatics of the Tomas Bata University in Zlín. His research interests are modelling and simulation of continuous-time chemical processes, polynomial methods, optimal and adaptive control. You can contact him on email address dostalp@fai.utb.cz.