MATLAB TOOLBOX FOR SELF-TUNING PREDICTIVE CONTROL OF TIME-DELAYED SYSTEMS

Radek Holiš, Vladimír Bobál
Department of Process Control
Faculty of Applied Informatics, Tomas Bata University in Zlin
Nad Stráněmi 4511, Zlin 76005, Czech Republic
E-mail: rholis@fai.utb.cz

KEYWORDS
Self-tuning Control, Model Predictive Control, MPC, Time-delay, MATLAB Toolbox

ABSTRACT

The designed MATLAB/SIMULINK Toolbox is dedicated to develop and design predictive Self-Tuning Control (STC) algorithm for the time-delayed systems. In practice, many processes can exhibit time-delay in their dynamic behavior, which is mainly caused by a time needed for transport of the energy, information, or mass. In lights of these facts, it is necessary to develop suitable algorithm and verify its correct dynamic behavior using simulation first, so this Toolbox can be used in advantage. This paper deals with the basic principles of Model Predictive Control (MPC), calculation of control law, design process of the predictive controller and recursive identification of control process using Recursive Least Squares Method (RLSM). There are also many cases when compensation of measurable disturbance is required, so this Toolbox allows compensating of this disturbance.

INTRODUCTION

Time-delayed systems appear in many processes in industry and other fields, including economical as well as biological systems (Camacho and Normey-Rico 2007). These processes are difficult to control using standard feedback controllers. When the relative time-delay is very large or a high performance of the control process is desired, we can choose MPC as the suitable algorithm for these types of processes. The predictive control strategy contains a model of the process in the structure of the controller. The first time-delay compensation algorithm was shown by (Smith 1957). This algorithm is known as the Smith Predictor (SP) and it contains a dynamic model of the time-delay process and it can be called as the first MPC algorithm. For more complex processes, containing time-delay and affected by a measurable disturbance, MPC strategy can be used (Maciejowski 2002).

The MPC is an attractive set of the control strategies widely used in the industry. The popularity of the MPC is mostly due to its leading to a safety operation of processes under all circumstances and ability to use constraints. The MPC is known as a control strategy where based on the measurements of plant's states at time, a mathematical model of the plant (often referred to as the prediction model) is being used for prediction of the evolution of the plant in the future.

The MPC with allowance to control of the time-delayed processes, ability of self-tuning, and possibility of the measurable disturbance compensation can be powerful and versatile algorithm for control of various processes.

This paper deals with the use of MPC for processes with time-delay with possibility of measuring disturbance compensation.

Strategy of MPC presents a series of advantages over other methods. The MPC can be used to control a great variety of processes, ranging from those with relatively simple dynamics to other more complex ones, including systems with long time-delay, unstable ones or non-minimum phase. The multivariable case can easily be dealt with. The additional advantage is that extension to the treatment of constraints is conceptually simple and these can be systematically included during the design process. This approach of control is a totally open methodology based on certain basic principles that is allowed for future extensions (Camacho and Normey-Rico 2007; Rossiter 2003; Haber et al. 2011).

The MPC has been deployed on slower processes in its early days (Kvasnica 2009).

It was caused by the large computational complexity of control algorithms and large time demands. Trends have expanded towards modifications of predictive control over the years. Nowadays, the MPC strategy can be used for controlling of very fast processes. These processes can have requirement for computation of control action in microseconds (e.g. explicit approach of MPC can be used). In practice, an excellent industrial survey reports many successful applications of the MPC in various industry areas (Qin and Badgewell 1997; Rawlings and Mayne 2009).

An extended version of the Generalized Predictive Control (GPC) algorithm is dedicated for design of the adaptive predictive controller in this paper.

This paper is arranged as follows. The extended GPC algorithm is described in the first section. The next section shows computation of the cost function for GPC and computation of control law. Brief description of the recursive identification procedure is introduced in the following section. The designed Toolbox is briefly described afterwards. The next section contains examples of the simulation control using designed Toolbox and the last section concludes this paper.

Proceedings 32nd European Conference on Modelling and Simulation ©ECMS Lars Nolle, Alexandra Burger, Christoph Tholen, Jens Werner, Jens Wellhausen (Editors)
ISBN: 978-0-9932440-6-3/ ISBN: 978-0-9932440-7-0 (CD)

EXTENDED VERSION OF THE GENERALIZED PREDICTIVE CONTROL ALGORITHM

The basic MPC structure with the extended GPC algorithm is schematically displayed in the Figure 1.

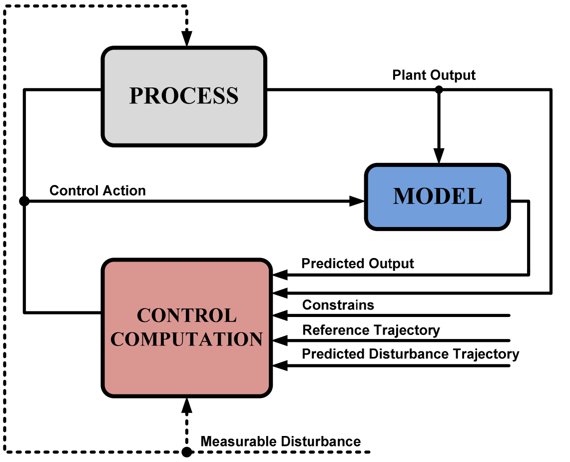

Figure 1 : Extended Structure of the MPC

The basic GPC algorithm minimizes a cost function that may be written as

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

where J is the function of the N_x, which represents N_1, N_2, and N_u. The N_1 and N_2 are the minimum and maximum horizons of cost function, and N_u is the control horizon of the cost function. This horizon should be chosen with regard to dynamics of controlled process to handle step response (Rossiter 2003; Moudgalya 2007). The $\hat{y}(k+i)$ is an optimum prediction of the system output. Coefficients $\delta(i)$ and $\lambda(i)$ are weighting coefficients and $w(k+i)$ is a vector of future reference sequence.
The goal of the predictive control is to calculate the future incremental control action of the $\Delta u(k)$, $\Delta u(k+1)$,... The cost function J (1) is minimized to obtain the final control law. This is realized by minimizing with respect to Δu, where the predictions $\hat{y}(k+i)$ are first expressed as a function of the past data and the future control actions $\Delta u(k+i-1)$. Thus, J can be considered as a function of the future control sequence. The control horizons as well as weighting factors are the tuning parameters, which can be changed to modify steepness and rapidity of the control course as required (Camacho and Normey-Rico 2007).
The horizons N_1 and N_2 are computed as $N_1 = d+1$ and $N_2 = N_u + d$, because of the time-delay characteristics of the process. In practice, N_1 and N_2 are hardcoded in the algorithm and N_u is the only changeable parameter.
The modified mathematical model of the Controlled Auto-Regressive Integrated Moving Average (2) (CARIMA) is used by the GPC to compute the predictions. It is the typical CARIMA model extended by the vector $v(k)$, which represents measurable disturbance sample

$$A(z^{-1})y(k) = z^{-d}B(z^{-1})u(k-1) + z^{-dv}D(z^{-1})v(k) + \frac{C(z^{-1})}{\Delta}e_s(k) \quad (2)$$

where d corresponds to number of steps of the time-delay for process, dv represents number of steps of the time-delay for disturbance, $e_S(k)$ is the white noise, and $\Delta = 1 - z^{-1}$. The polynomial $D(z^{-1})$ represents character of disturbance and the polynomial $C(z^{-1})$ describes character of the noise. This character is difficult to determine; therefore, polynomial $C(z^{-1})$ is chosen to be equal to one (Camacho and Bordons 2004; Fikar and Mikleš 2008; Clarke et al. 1987a).
Consider equation (2) multiplied by Δ. Then, predication model can be represented as follows, where output can be predicted as

$$\hat{y}(k+1) = \sum_{i=1}^{na+1} \tilde{a}_i y(k+1-i) + \sum_{i=1}^{nb} b_i \Delta u(k-d-i) + \sum_{i=1}^{nd} d_i \Delta v(k+1-dv-i) \quad (3)$$

where na, nb, and nd are degrees of polynomials $A(z^{-1})$, $B(z^{-1})$, and $D(z^{-1})$. The white noise $e_S(k)$ and its future values are considered to be equal to zero for the prediction of the future output values.
In case where time-delay is present, following equation should be used when equation (3) is applied recursively for $i = 1, 2, ..., N_u$ (Clarke et al. 1987b).

$$\mathbf{\hat{y}} = \mathbf{Gu} + \mathbf{Hu_1} + \mathbf{Sy_1} + \mathbf{H_{V1}v_1} + \mathbf{H_{V2}v_2} \quad (4)$$

Matrices **G**, **H** and **S** are constant matrices of dimensions $N_u \times N_u$, $N_u \times nb$, and $N_u \times (na+1)$, respectively. Matrices $\mathbf{H_{V1}}$ and $\mathbf{H_{V2}}$ are of dimensions $N_u \times (nd-1)$ and $N_u \times N_u$. Matrix $\mathbf{H_{V1}}$ can be used only in case when degree of polynomial $D(z^{-1})$ is equal to 2 or higher.
Following equation corresponds to the free response of the system that is the output that would be obtained if the control signal was kept constant.

$$\mathbf{f} = \mathbf{Hu_1} + \mathbf{Sy_1} + \mathbf{H_{V1}v_1} \quad (5)$$

Forced response of the system can be written as the next equation

$$\mathbf{f_r} = \mathbf{Gu} + \mathbf{H_{V2}v_2} \quad (6)$$

Vectors $\mathbf{u_1}$, $\mathbf{y_1}$, $\mathbf{v_1}$, and $\mathbf{v_2}$ are defined in the following equations. Sum of the free response (5) and the forced response (6) leads to overall response of the system defined by the following equation

$$\mathbf{\hat{y}} = \mathbf{f} + \mathbf{f_r} \quad (7)$$

Control Law Computation and Cost Function

The predicated output $\hat{y}$, expressed in the previous paragraph, is part of the equation (1). It is evident that J is the cost function of $\mathbf{y_1}$, $\mathbf{u}$ and $\mathbf{u_1}$. The individual elements of the summation of the cost function in the equation (1) can be written in a matrix form. This cost function can be defined as

$$J = (\mathbf{Gu} + \mathbf{Hu_1} + \mathbf{Sy_1} - \mathbf{w})^T \mathbf{Q_\delta} (\mathbf{Gu} + \mathbf{Hu_1} + \mathbf{Sy_1} - \mathbf{w}) + \\ + \mathbf{u}^T \mathbf{Q_\lambda} \mathbf{u} \tag{8}$$

where $\mathbf{Q_\delta}$ and $\mathbf{Q_\lambda}$ are the diagonal weighting matrices of size $N_u \times N_u$ with elements $\delta(j)$ and $\lambda(j)$, respectively. Although, in practice, the most common choice is to set $\delta(j)$ and $\lambda(j)$ constants on the horizon. In a fact, the values of these weighting factors must be normalized in order to obtain a correct weighting of the different errors and controller outputs.
After some manipulations J can be written as

$$J = \mathbf{u}^T (\mathbf{Q_\lambda} + \mathbf{G}^T \mathbf{Q_\delta G})\mathbf{u} + 2(\mathbf{Hu_1} + \mathbf{Sy_1} - \mathbf{w})^T \mathbf{Q_\delta Gu} + \\ + (\mathbf{Hu_1} + \mathbf{Sy_1} - \mathbf{w})^T \mathbf{Q_\delta} (\mathbf{Hu_1} + \mathbf{Sy_1} - \mathbf{w}) \tag{9}$$

Minimizing J with respect to $\mathbf{u}$, it means $\frac{\partial J}{\partial \mathbf{u}} = 0$, leads to

$$\mathbf{Mu} = \mathbf{P_0 y_1} + \mathbf{P_1 u_1} + \mathbf{P_2 w} \tag{10}$$

where $\mathbf{M} = \mathbf{G}^T \mathbf{Q}_\delta \mathbf{G} + \mathbf{Q}_\lambda$ is of dimension $N_u \times N_u$, $\mathbf{P}_0 = -\mathbf{G}^T \mathbf{Q}_\delta \mathbf{S}$ of dimension $N_u \times (na+1)$, $\mathbf{P}_1 = -\mathbf{G}^T \mathbf{Q}_\delta \mathbf{H}$ of dimension $N_u \times nb$ and $\mathbf{P}_2 = \mathbf{G}^T \mathbf{Q}_\delta$ of dimension $N_u \times N_u$.
In a receding horizon algorithm only the current value of the $\Delta u(k)$ is computed, so if $\mathbf{m}$ is the first row of matrix $\mathbf{M}^{-1}$, then $\Delta u(k)$ is given by

$$\Delta u(k) = \mathbf{mP_0 y_1} + \mathbf{mP_1 u_1} + \mathbf{mP_2 w} \tag{11}$$

Equations (8) – (10) deal with the case of no disturbance rejection. For the expression of the final control law containing compensation of the measurable disturbance, $\Delta u(k)$ is computed as the following control law form

$$\Delta u(k) = \mathbf{mP_0 y_1} + \mathbf{mP_1 u_1} + \mathbf{mP_2 w} + \mathbf{mP_{V1} v_1} + \mathbf{mP_{V2} v_2} \tag{12}$$

where $\mathbf{P_{V1}} = -\mathbf{G}^T \mathbf{Q}_\delta \mathbf{H_{V1}}$ is of dimension $N_u \times (nd+1)$ and $\mathbf{P_{V2}} = -\mathbf{G}^T \mathbf{Q}_\delta \mathbf{H_{V2}}$ of dimension $N_u \times N_u$.
$\mathbf{H_{V1}}$ and $\mathbf{H_{V2}}$ are matrices including the coefficients of the system step response to the disturbance.
Future values of the disturbance can be determined only in certain cases, e.g. be measurement or generally in case, when it is related to the process load. In other cases, it can be predicted using means, trends, past data, other information, or by combination of specified items. If this is the case, the term corresponding to future deterministic disturbance can be computed (Schwarz et al. 2010). After introducing vectors $\mathbf{y_1}$, $\mathbf{u_1}$, $\mathbf{w}$, $\mathbf{v_1}$ and $\mathbf{v_2}$, final control law is defined as

$$\Delta u(k) = \mathbf{mP_0} \begin{bmatrix} \hat{y}(k+d) \\ \hat{y}(k+d-1) \\ \vdots \\ \hat{y}(k+d-na) \end{bmatrix} + \mathbf{mP_1} \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-nb) \end{bmatrix} + \\ + \mathbf{mP_2} \begin{bmatrix} w(k+d+1) \\ w(k+d+2) \\ \vdots \\ w(k+d+N_u) \end{bmatrix} + \mathbf{mP_{V1}} \begin{bmatrix} \Delta v(k-1) \\ \Delta v(k-2) \\ \vdots \\ \Delta v(k-nd+1) \end{bmatrix} + \\ + \mathbf{mP_{V2}} \begin{bmatrix} \Delta v(k) \\ \Delta v(k+1) \\ \vdots \\ \Delta v(k+N_u-1) \end{bmatrix} \tag{13}$$

If the future load disturbance is constant and equal to the last measured value (i.e. $\Delta v(k) = 0$), the last term of the equation (13) vanishes (Pawlowskaa et al. 2012).
It is evident that matrices $\mathbf{H_{V1}}$ and $\mathbf{H_{V2}}$ are dependent on the relative difference between number of steps of time-delay of input-output and disturbance-output which is defined as

$$\rho = d - dv \tag{14}$$

This leads to three types of structures for matrices $\mathbf{H_{V1}}$ and $\mathbf{H_{V2}}$ based on the value of ρ:

- $\rho < 0$ and $\rho = 0$

$$\mathbf{H_{VX}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 \\ h_1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & 0 & 0 & 0 & 0 \\ h_{Nu+\rho} & \cdots & h_1 & 0 & 0 & 0 \end{bmatrix} \left.\vphantom{\begin{matrix}0\\0\\0\\0\end{matrix}}\right\} |\rho| \tag{15}$$

- $\rho > 0$

$$\mathbf{H_{VX}} = \begin{bmatrix} h_{\rho+1} & h_\rho & \cdots & h_1 & 0 & 0 \\ h_{\rho+2} & h_{\rho+1} & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & h_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & h_2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N+\rho} & h_{N+\rho-1} & \cdots & \cdots & h_\rho & h_{\rho+1} \end{bmatrix} \left.\vphantom{\begin{matrix}0\\0\\0\\0\end{matrix}}\right\} \rho \tag{16}$$

where h_i are the coefficients of $\mathbf{H_{V1}}$ and $\mathbf{H_{V2}}$ matrices obtained from the delay free disturbance response moved up/down according to value of ρ (Pawlowskaa et al. 2012).

RECURSIVE IDENTIFICATION

The identification of systems deals with the problem of creating mathematical models of dynamical systems based on data observed from the system. It is an alternative procedure for obtaining a model in case, when it is not possible to determine a set of differential equations that describes the dynamic behavior of the system. The MPC requires an internal model of the system; therefore, really precise model of process is necessary for correct behavior of predictive algorithm.

Identification of control processes can be divided into two groups, which are used most often. The first group is Offline (one-time) Identification Methods (OfIM) and the second is Online (ongoing) Identification Methods (OnIM).

The OfIM type as well as the OnIM type can be used during the real-time control of processes. The estimated parameters obtained from the OfIM are usually selected as a starting point for the STC. They can be also chosen as the internal model throughout the control procedure when the process does not change its dynamic behavior much and adaptive control is not required. These STCs can utilize an auto-tuning or adaptive approach in many practical applications (Bitmead et al. 1990). The most known adaptive approach is to use OfIM recursively.

Offline Identification Methods

The well known OfIM is the MATLAB function *fminsearch*. It finds the minimum of the entered function without restricting conditions. The entered function can be single variable or multivariable type. This function uses the simplex search method for finding the minimum of a function. This is a direct search method that does not use numerical or analytic gradients. However, the most known method for the identification of the discrete transfer function model parameters is the Least Squares Method (LSM) based on the idea of linear regression. This identification algorithm can be carried out in a recursive manner as well in an order to use it for STC. The LSM is based on minimizing the sum of squared subtraction of measured and model output value.

The LSM is defined as the vector $\hat{\mathbf{\Theta}}$ that minimizes the quadratic error

$$\hat{\mathbf{\Theta}} = \left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{y} \tag{17}$$

Note, that $\hat{\mathbf{\Theta}}$ is a vector of estimated model parameters, which has dimension $2n$, $\mathbf{F}$ is a matrix of dimension $N - n - d \times 2n$, $\boldsymbol{y}$ is a data vector of dimension $N - n - d$, where N is a number of measured data, n is an order of system, and d is a number of steps of time-delay. The $\mathbf{F}$ depends on past inputs and outputs and that this condition can be fulfilled if the input signal sequence is adequately chosen in such a way that the obtained vectors are linearly independent. (Camacho and Normey-Rico 2007).

Online Identification Methods

The OnIM are mainly used to adjust the estimates of the process parameters from initial estimates in the each sampling period. Since the approach, when calculation of estimated parameters is performed each sampling period, these methods are capable to react on changes in a dynamic behavior of system as well as they are able to compensate slightly non-liner behavior of the system.

One of the advantages of the process parameter estimation using the LSM is fact, that this algorithm can be used recursively. The parameter vector computed at step k can be computed as a function of the parameter vector estimated at step $k – 1$.

The recursive least squares method (RLSM) is the most known recursive method and it uses the AutoRegressive eXogenous (ARX) model (Bobál et al. 2005).

$$y(k) = \mathbf{\Theta}^T(k)\mathbf{\Phi}(k) + e_s(k) \tag{18}$$

where $\mathbf{\Theta}$ is a vector of model parameters

$$\mathbf{\Theta}^T(k) = \left[a_1 \quad a_2 \ldots a_n \quad b_1 \quad b_2 \ldots b_n\right] \tag{19}$$

and $\mathbf{\Phi}$ is a regression vector

$$\begin{aligned}\mathbf{\Phi}^T(k) = [-y(k-1) \quad -y(k-2)\ldots -y(k-n) \\ u(k-d-1) \quad u(k-d-2)\ldots u(k-d-n)]\end{aligned} \tag{20}$$

Final RLSM algorithm can be defined as

$$\hat{\mathbf{\Theta}}(k) = \hat{\mathbf{\Theta}}(k-1) - \frac{\mathbf{C}(k-1)\mathbf{\Phi}(k)}{1+\xi(k)}\hat{e}(k) \tag{21}$$

where $\mathbf{C}$ is covariance matrix and

$$\xi(k) = \mathbf{\Phi}^T(k)\mathbf{C}(k-1)\mathbf{\Phi}(k) \tag{22}$$

The RLSM can be modified by weighting of the past data and forgetting of them to always work with the most actual and relevant data. Application of the RLSM with exponential forgetting results in a more realistic situations. Parameters of the control law are being continuously adjusted in order to track time-varying properties of the controlled plant (Bobal et al. 2005; Skormin 2016). Final algorithm is defined as

$$\hat{\mathbf{\Theta}}(k) = \hat{\mathbf{\Theta}}(k-1) + \frac{\mathbf{C}(k-1)\mathbf{\Phi}(k)}{\varphi^2+\xi(k)}\hat{e}(k) \tag{23}$$

where covariance matrix is defined as follows

$$\mathbf{C}(k) = \frac{1}{\varphi^2}\left[\mathbf{C}(k-1) - \frac{\mathbf{C}(k-1)\mathbf{\Phi}(k)\mathbf{\Phi}^T(k)\mathbf{C}(k-1)}{\varphi^2+\xi(k)}\right] \tag{24}$$

The RLSM with adaptive directional forgetting eliminates disadvantages of the RLSM with exponential forgetting. It forgets old information only in the direction in which new data bring new information, which also helps to avoid the estimator windup effect.

The RLSM with exponential forgetting as well as with adaptive directional forgetting has been chosen as an algorithm for the STC algorithm used in the introduced Toolbox (Bobal et al. 2005; Skormin 2016).

TOOLBOX DESCRIPTION

The Toolbox for the STC GPC of time-delayed processes with measurable disturbance compensation is depicted in the Figure 2. This Toolbox was developed using the MATLAB R2014b. Basic setting of Toolbox is possible in the *init.m* file, which is an initialization routine. This routine is executed automatically once the simulation is started using the MATLAB/SIMULINK.

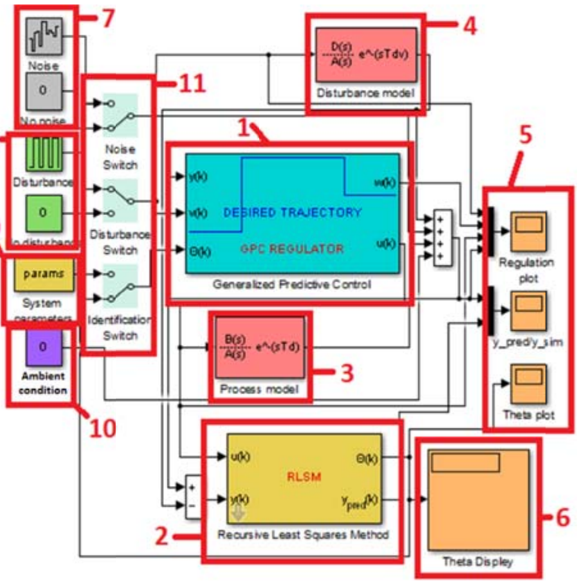

Figure 2 : STC GPC Toolbox MATLAB Scheme

The Toolbox consists of the following parts:

1 – GPC controller block
2 – RLSM identification block (used for STC)
3 – Controlled process model
4 – Disturbance model
5 – Resulting charts of control courses and predicates
6 – Estimated parameters of process from RLSM
7, 8 – Noise signal and disturbance signal
9 – System parameters setting
10 – Ambient condition setting signal
11 – Activation/deactivation switches of 9, 10, and 11

The GPC controller block 1 contains three tabs, see Figure 3. First tab is used for the setting of the Sample time, Dead times (time delays), Control horizon and Weighting parameters. Second tab is intended to design desired trajectory and last tab can set properties of disturbance compensation.

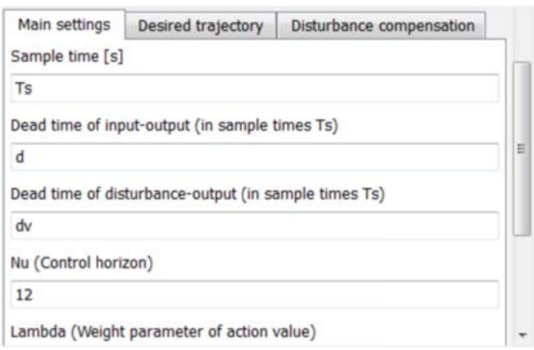

Figure 3 : Main Settings of GPC Window

The measurable disturbance compensation can be enabled/disabled based on the checkbox as it is visible on the Figure 4. If disabled, No Disturbance Compensation (NDC) approach is used. In case when disturbance compensation is enabled, user can choose one of the two possible ways of its compensation. First, the Normal Measurement Disturbance (NMD) means that disturbance is measured every sample interval and output is compensated based on the present and past data of disturbance. Second option is to use the Predicted Vector of Disturbance (PVD), where course of the disturbance over time is known during the whole simulated control process. Moreover, disturbance is measured for overcompensation of invalid data as well. For example, disturbance can be first measured and used for the control or disturbance vector can be statistically computed based on the past data, etc. Usage of the PVD significantly improves whole control process in terms of control quality.

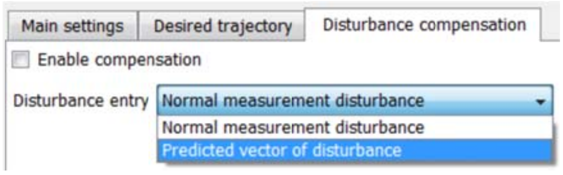

Figure 4 : GPC Disturbance Compensation Window

The RLSM identification block 2 is allowed for the self-tuning purposes by the Identification Switch 11. Otherwise, identified parameters of process are constants during the control process. The RLSM block allows to set Sample/dead time in the first tab. Second tab is designed for setting of the Type of identification (RLSM, RLSM with exponential forgetting, or RLSM with adaptive directional forgetting). Other boxes can modify the RLSM parameters.

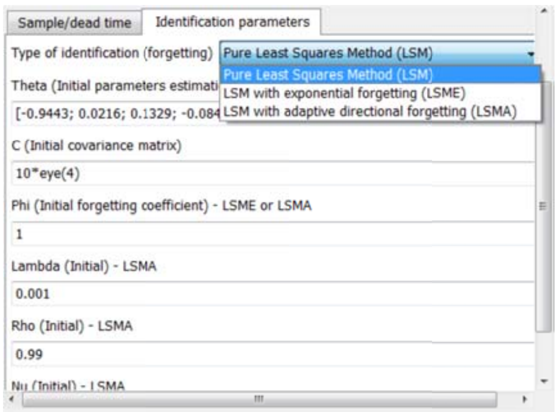

Figure 5 : RLSM Settings for the STC Window

To execute the simulation, set parameters of the controlled system in the *init.m* file first. Then, enable/disable Noise and external Disturbance using Switches 11. Choose STC with RLSM or non STC algorithm using the Identification Switch. Set Ambient temperature, if required. Set GPC controller block 2 and RLSM identification block according to description above. Run the simulation and display charts using 5.

SIMULATION VERIFICATION OF TOOLBOX

The designed STC GPC Toolbox was verified by simulation on three examples – GPC algorithm without STC, STC GPC algorithm, and GPC algorithm with disturbance compensation.

The Controlled process model (item 3 of the Figure 2) is represented as the following continuous transfer function, which was used for all simulations.

$$G_S(s) = \frac{B(s)}{A(s)} e^{-sT_d} = \frac{2}{20s^2 + 9s + 1} e^{-10s} \qquad (25)$$

The following second order linear discrete transfer function was used for the simulation purposes as a model for estimated parameters of controlled process for GPC and RLSM.

$$G_S(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-d} \qquad (26)$$

GPC Algorithm without STC

First, the GPC algorithm without STC was verified. Very precise estimated model (27) of the controlled system was used.

$$G_S(z^{-1}) = \frac{0.1490z^{-1} + 0.1104z^{-2}}{1 - 1.2770z^{-1} + 0.4066z^{-2}} z^{-5}, \quad T_S = 2s \qquad (27)$$

Following parameters were used for all simulations:

$d = 5$, $\delta = 1$, $\lambda = 1$, $N_1 = 6$, $N_u = 10$, and $N_2 = 15$

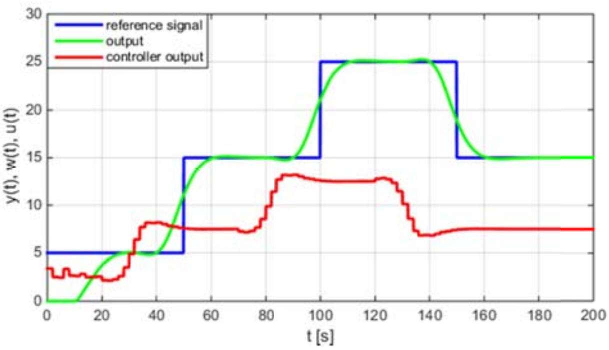

Figure 6 : GPC without STC – Exact Identification

From the Figure 6 it is obvious that the control quality is very good after start-up phase and overshoots are not significant. Next, the exact estimated parameters were changed subsequently

$$G_S(z^{-1}) = \frac{0.2000z^{-1} + 0.1000z^{-2}}{1 - 1.5000z^{-1} + 0.5000z^{-2}} z^{-5} \qquad (28)$$

where inaccurate identification is performed.

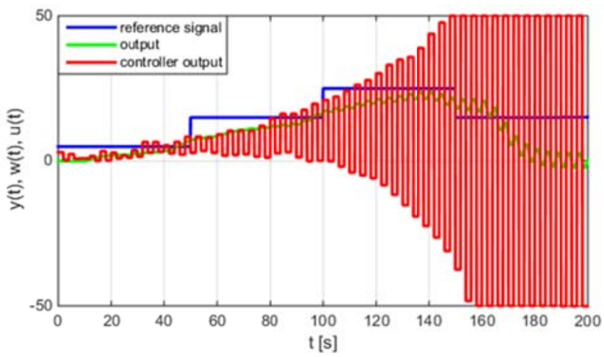

Figure 7 : GPC without STC – Inaccurate Identification

The Figure 7 shows that when identification process is underestimated and parameters are not accurate, whole control process becomes unstable and it cannot be effectively controlled using GPC.

STC GPC Algorithm

Disadvantages of inaccurate parameters estimation can be eliminated by STC GPC algorithm. The RLSM with adaptive directional forgetting and the controlled system model (28) was used for the simulation purposes.

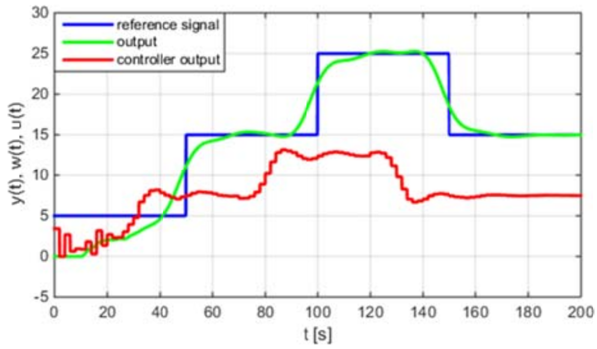

Figure 8 : STC GPC – Control Courses

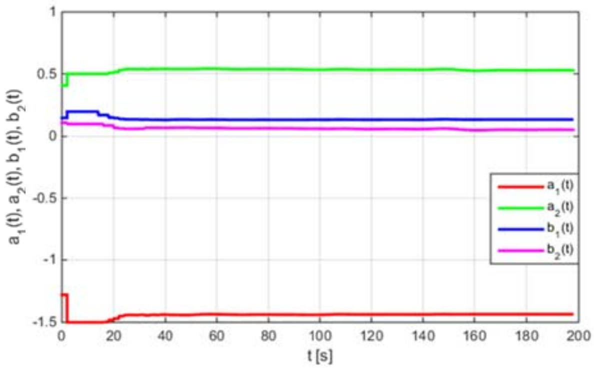

Figure 9 : STC GPC – Parametrs Evolution

The Figure 8 depicts that when GPC STC is used, even initial inaccurate system parameter estimation does not prevent high control quality. System parameters evolution is captured on the Figure 9.

GPC Algorithm – Disturbance Compensation

The controlled system model (27) was used to verify functionality of the GPC with NDC, NMD, and PVD. Control courses are depicted on the Figure 10.

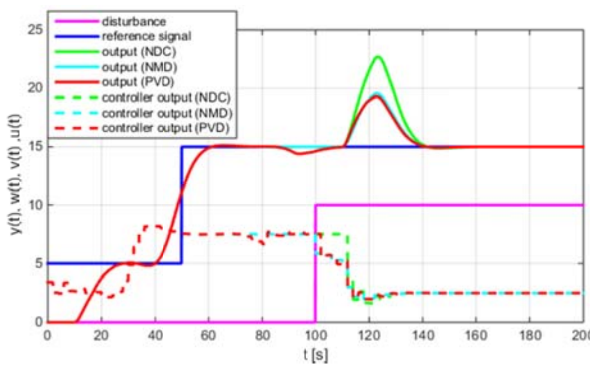

Figure 10 : GPC – Disturbance Compensation

CONCLUSION

This paper has presented an extended STC GPC Toolbox with possibility to compensate the measurable disturbance. The Toolbox has been created in the MATLAB/SIMULINK environment with a purpose to create a simulation suitable for the design and verification of adaptive control of time-delay systems with usage of the MPC strategy.

The GPC algorithm as itself without using the STC is able to control processes in a really good quality after start-up phase and overshoots are not significant. However, this is possible only in case when process parameters are estimated very precisely. Otherwise, in case when these parameters are not estimated with sufficient precision, control process becomes unstable. First option is to use suitable identification process for parameters estimation or use the STC algorithm. The STC algorithms have several advantages, e.g. initial parameter estimation can be only raw, slightly nonlinear process can be controlled using the STC, and influence of an unexpected conditions during the control process can be eliminated. The simulation shows that RLSM with adaptive directional forgetting can be suitable STC algorithm for the MPC, in general.

Incorporation of the disturbance compensation into the control law can have really positive effect on overall control processes. An overshoot caused by the disturbance can be eliminated when it is measured and predicted. The GPC with NMD and PVD improves the control quality and reduces the overshoot in comparison with the GPC with NMC.

Toolbox is maintained by the Department of Process Control, Faculty of Applied Informatics, Tomas Bata University in Zlin and for its downloading, feel free to contact authors or mentioned department.

ACKNOWLEDGEMENT

This work was supported by the Czech Republic Ministry of Education - grant IGA/CebiaTech/2018/002.

REFERENCES

Bitmead, R.R., Gevers, M. and V. Hertz. 1990.*Adaptive optimal control. The thinking man's GPC*, Prentice Hall, Englewood Cliffs, New Jersey.

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

Camacho E.F. and C. Bordons. 2004. *Model predictive control*, Springer Verlag, London.

Camacho E.F. and J.E. Normey-Rico. 2007.*Control of dead-time processes*, 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.; C. Mohtadi and P.S. Tuffs. 1987b. "Generalized predictive control, part II: extensions and interpretations", Automatica 23,149-160.

Fikar, M. and J. Mikleš. 2008. *Process modelling, optimisation and control*, Springer-Verlag, Berlin.

Haber, R., Bars, R.and U. Schmitz. 2011. *Predictive control in process engineering: From the basics to the applications*. Weinhaim: Willey-VCH Verlag.

Kvasnica, M. 2009. *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools*, VDM Verlag.

Maciejowski, J. M. 2002.*PredictiveControl, with Constraints*,Pearson Education.

Moudgalya, K.M. 2007.*Digital control*. Chichester: John Wiley.

Pawlowskaa A., Guzmána J. L., Normey-RicoJ. E. and M. Berenguela. 2012."Improving feedforward disturbance compensation capabilities in Generalized Predictive Control". Journal of Process Control 22, 527-539.

Qin, S. J. and T. A. Badgewell. 1997. *An overview of industrial model predictive control technology*. In: Chemical Process Control – V, volume 93, no. 316, 232-256. AIChe Symposium Series – American Institute of Chemical Engineers.

Rawlings, J. B. and D. Q. Mayne. 2009. *Model Predictive Control: Theory and Design*. Nob Hill Pub.

Rossiter, J.A. 2003.*Model based predictive control: a practical approach*, CRC Press.

Schwarz, M. H., Cox, C. S. and J. Börcsök. 2010."A Filtered Tuning Method for a GPC Controller". In: University of Kassel, Germany, 180-185.

Skormin, A. V. 2016. *Introduction to Process Control: Analysis, Mathematical Modeling, Control and Optimization,* Springer Verlag, London.

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

AUTHOR BIOGRAPHIES

RADEK HOLIŠ studied at the Tomas Bata University in Zlín, Czech Republic, where he obtained his master's degree in Automatic Control and Informatics in 2014. He now attends Ph.D. study in the Department of Process Control, Faculty of Applied Informatics of the Tomas Bata University in Zlín. His research interest focuses on modeling and simulation of discrete technological processes, adaptive control and model predictive control. He currently works at Honeywell HTS-CZ, Brno in Aerospace division as Software Design Engineer. His email address is rholis@fai.utb.cz.

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.