Department of Automation and Applied Informatics
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

Laboratory of Process Control
Institute of Process Engineering and Plant Design
Faculty of Process Engineering, Energy and
Mechanical Systems
University of Applied Science Cologne

Fakhredin Arousi

PREDICTIVE CONTROL ALGORITHMS FOR LINEAR AND NONLINEAR PROCESSES

(PREDICTIVE PID CONTROL, DECOUPLING, VOLTERRA MODEL BASED NONLINEAR PREDICTIVE CONTROL)

Ph.D. Thesis

Scientific supervisors:

dr. Ruth Bars
Department of Automation and Applied Informatics
Faculty of Electrical Engineering and Informatics
Budapest University of Technology and Economics

Prof. Dr.-Ing. Robert Haber
Laboratory of Process Control
Institute of Process Engineering and Plant Design
Faculty of Process Engineering, Energy and
Mechanical Systems
University of Applied Science Cologne

Budapest, 2009

1. Introduction

In the last 30 years predictive control has gained wide acceptance in the process industries besides PID control [42]. Predictive control determines the control increment sequence minimizing a cost function considering the deviation between the reference signal and the output signal predicted in a future horizon. The future output signal is predicted based on the system model. The cost function can be minimized under constraints. Without constraints in linear case GPC (Generalized Predictive Control) gives an analytic solution for the control algorithm [21]. From the calculated control sequence only the first one is applied and in the next sampling point the procedure is repeated (receding horizon strategy). The tuning parameters of predictive control algorithms are the prediction horizon, the initial point of the prediction horizon, the control horizon (number of the allowed changes of the control signal during the calculation) and the weighting parameters in the cost function. Nowadays several industrial program packages support and ensure an appropriate environment for industrial applications of predictive control (e.g. [32]). These program packages provide possibilities for measurement and monitoring different process variables, execute unit step response measurements, identification, simulation, real-time control.

In case of systems containing big dead time significantly faster control performance can be achieved with predictive control than with PID control.

Nevertheless it is difficult to introduce advanced control algorithms in practice, as PID algorithms also work well, and generally control problems can be solved with them. Operators of industrial processes have a good expertise in PID controller tuning. Different tuning rules of thumb can be used, which are based mostly on the approximation of the system with a first-order model with dead time [17, 20, 36]. PID control algorithms can be equipped with predictive property if the error signal applied to calculate the control signal is not the actual error, but its predicted value in a future point or in a future horizon [34, 35]. In this control system the operator will see PID control with embedded predictive properties.

Industrial processes generally are complex, containing interacting MIMO controller loops. It is required to select appropriate pairing of the input and output variables to ensure more effective control. There are different decoupling techniques. Predictive control for MIMO processes ensures inherent decoupling if the cost function does not include terms punishing the control increments. But these terms are used to keep the control variables within their limit values. Enhancing decoupling properties of predictive control algorithms is important for practical applications [18, 39].

Most of the plants contain nonlinearities. The control algorithms generally consider linear, or linearised models around the working points. Predictive control algorithms based on nonlinear model of the plant would ensure better performance in the whole operating range, than algorithms using linear models. In the recent years there is an active research to develop nonlinear predictive control algorithms [23]. One direction is to use Wiener, Hammerstein or Volterra models to approximate the nonlinear characteristics and to develop predictive controllers based on these models [25, 26, 29, 30, 31, 37].

2. The objective of this thesis

The aim of this work was to contribute to the area of predictive control developing algorithms

- for predictive PI(D) control,
- for improving decoupling properties of MIMO predictive control,
- for nonlinear predictive control using Volterra models.

For applying predictive PI(D) controllers I used first- or second-order approximation of the plant with dead time. For these models the tuning of the parameters could be based on modified simple tuning rules of thumb. The aim was also to enhance the algorithms with extensions to improve tracking and robustness properties.

For dynamic decoupling I applied adaptive tuning of the weighting factors in the cost function. I used these decoupling techniques for TITO processes.

I have suggested an iterative optimization method to control nonlinear systems identified as Volterra models.

These algorithms contribute to better practical applicability of predictive control.

3. Research methodology

For the three problems theoretical considerations have lead to new control algorithms.

Some parts of the research were based also on previous diploma projects [16, 37]. Discussions with former PhD student Ulrich Schmitz were especially fertilizing. The common research project with the Sevilla University, Engineering Department of Systems and Automation in 2008 also has promoted the approach to the nonlinear Volterra model based predictive control algorithm.

The performance of the algorithms has been tested by programs written in MATLAB. Besides the simulation examples case studies have been also analyzed using data from laboratory experiments (two-tank system, ball and beam, Dept. of Automation and Applied Informatics, BME, distillation column, Laboratory of Process Control, Cologne).

The research work has been done in tight cooperation with the Laboratory of Process Control, Institute of Process Engineering and Plant Design, University of Applied Science, Cologne.

4. New Results

4.1 Thesis group 1: Predictive PI(D) control of linear SISO processes

PI(D) control algorithms are the most accepted in the process industries. The operators of industrial processes have expertise of tuning PI(D) controllers. PI(D) controllers could be equipped with predictive properties. In this case control properties of plants with big dead time could be improved significantly. Tuning rules of PI(D) control can be modified for predictive PI(D) control. Thus the operators can see a PI(D) controller with embedded predictive properties.

In lot of cases industrial plants are approximated by first- or second-order transfer functions with dead time. Several tuning rules of thumb have been derived especially for first-order plants. Predictive PI(D) control based on these approximate models would provide simple control algorithms.

4.1.1 *I have introduced a simplified control horizon idea in the original formulation of predictive PI(D) control. I have derived predictive PI(D) control algorithms for simple plants with first- and second-order transfer functions with dead time based on their prediction equations.*

4.1.2 *I suggested simple tuning rules for predictive PI(D) controllers modifying the Kuhn's tuning rule for predictive PI(D) control. I applied tuning based on GPC-PID equivalence and gave a new calculation method for extended horizon control. I enhanced predictive PI(D) controllers by set-point weighting property introduced by Åström and Hägglund [17] for PID control.*

4.1.3 *I analysed the possibilities for enhancing robustness of predictive PI(D) controller using noise filter (T polynomial) and Smith predictor filter. I introduced the theory elaborated in [19, 41] for the predictive PI(D) control.*

Related publications: [1,2,3,4,5,9,11,12,13,14]

Predictive PI(D) control structure

A predictive PID controller considers not only one predicted output signal, but a series of predicted output values. [34, 35] suggested *m* number of parallel connected PID controllers with inputs of the predicted error signal values. For all controller paths the same PID controller is applied. The block diagram of the predictive PID controller is shown in Fig.1, where $\hat{e}(k+d+1+n_e \mid k)$ denotes the predicted values of the error signal n_e+1 step ahead the dead time. $y_r(k+d+1+n_e)$ is the reference signal and $\hat{y}(k+d+1+n_e \mid k)$ the predicted output signal. n_{e1} is the first, while n_{e2} is the last point of the prediction horizon beyond the dead time.

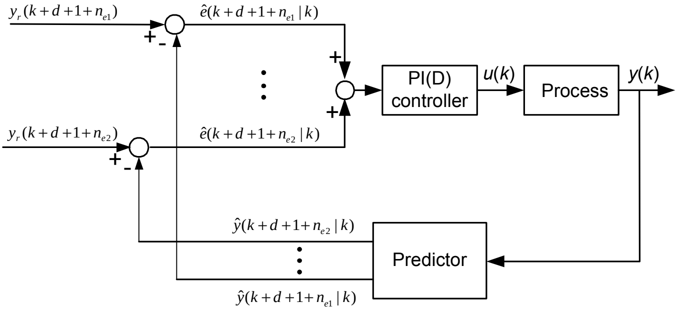
$y_r(k+d+1+n_{e1})$
$\hat{e}(k+d+1+n_{e1}\,|\,k)$
+
-
+
PI(D) controller
$u(k)$
Process
$y(k)$
+
$y_r(k+d+1+n_{e2})$
$\hat{e}(k+d+1+n_{e2}\,|\,k)$
+
-
$\hat{y}(k+d+1+n_{e2}\,|\,k)$
Predictor
$\hat{y}(k+d+1+n_{e1}\,|\,k)$

Fig.1. Predictive PI(D) controller with parallel paths

Discussion of 4.1.1

Predictive PI control algorithm

The output signal of a non-predictive discrete-time PI controller is expressed as $u(k) = k_{cp}e(k) + k_{ci}\sum_{i=1}^{k} e(i)$; k_{cp}, k_{ci} are the coefficients of the proportional and the integrating components of the controller. Taking the difference on both sides of the above equation at step *k* and *k*-1 leads to

$$\Delta u(k) = u(k) - u(k-1) = k_{cp}\left[e(k) - e(k-1)\right] + k_{ci}e(k) = p_0 e(k) + p_1 e(k-1)$$

In predictive PI control the manipulated variable is the sum of the controller outputs based on the predicted control errors with different prediction horizons. Applying the algorithm on a future error signal $d+1+n_e$ step ahead the actual time point the control increment is obtained as

$$\Delta u_{n_e}(k) = p_0\hat{e}(k+d+2+n_e\,|\,k) + p_1\hat{e}(k+d+1+n_e\,|\,k)$$

Let us introduce the following vector notations:

$$\mathbf{k}_{PI} = [p_1, p_0]^T$$

$$\hat{\mathbf{e}}_{PI}(k+d+1+n_e|k) = [\hat{e}(k+d+1+n_e|k), \hat{e}(k+d+2+n_e|k)]^T$$

With these notations $\Delta u_{n_e}(k) = \mathbf{k}_{PI}^T \hat{\mathbf{e}}_{PI}(k+d+1+n_e|k)$

In long-range predictive PI control the manipulated variable is obtained considering the sum of the predicted control errors. The control increment can be expressed as

$$\Delta u(k) = \mathbf{k}_{PI}^T \sum_{n_e=n_{e1}}^{n_{e2}} \hat{\mathbf{e}}_{PI}(k+d+1+n_e|k) = \mathbf{k}_{PI}^T \left(\mathbf{y}_{rPI,sum} - \hat{\mathbf{y}}_{PI,sum}\right)$$

where $\hat{\mathbf{y}}_{PI,sum} = \hat{\mathbf{y}}_{forcedPI,sum} + \hat{\mathbf{y}}_{freePI,sum}$ with

$$\hat{\mathbf{y}}_{freePI,sum} = \left[\sum_{n_e=n_{e1}}^{n_{e2}} \hat{y}_{free}(k+d+1+n_e|k) \quad \sum_{n_e=n_{e1}}^{n_{e2}} \hat{y}_{free}(k+d+2+n_e|k) \right]^T$$ and

$$\mathbf{y}_{rPI,sum} = \left[\sum_{n_e=n_{e1}}^{n_{e2}} y_r(k+d+1+n_e) \quad \sum_{n_e=n_{e1}}^{n_{e2}} y_r(k+d+2+n_e) \right]^T$$

For different systems the forced response can be calculated in the knowledge of the points of the step response, while the free response is obtained from the parameters of the model and from the past inputs and the actual and past output signals. As in the future error signal components future control increments also do appear, a closed form of the control increment can be obtained only with some simplification assumptions. With the assumption $\Delta u(k+i) = 0; \quad i = 1,2,...$ the forced response can be written as

$$\hat{\mathbf{y}}_{forced,PI,sum} = \left[\sum_{i=n_{e1}+1}^{n_{e2}} h_i \quad \sum_{i=n_{e1}+2}^{n_{e2+1}} h_i \right]^T \Delta u(k) = \mathbf{h}_{PI} \Delta u(k)$$

The control increment can be expressed as

$$\Delta u(k) = \left[1 + \mathbf{k}_{PI}^T \mathbf{h}_{PI}\right]^{-1} \mathbf{k}_{PI}^T (\mathbf{y}_{rPI,sum} - \hat{\mathbf{y}}_{freePI,sum})$$

I suggested two more strategies (introducing a simplified conception of control horizon in context of predictive PI(D) control) supposing equal control increments in the prediction horizon or a given n_u number for consecutive equal control increments. In the three cases (strategy 1, 2, 3) the calculation of $\mathbf{h}_{PI}$ is different.

Predictive PID algorithms are derived in a similar way.

I have derived the predictive PI(D) control for a first- and second-order process with dead time. In the prediction equations I used the special form of the models where the expression of the free response uses the model equations. Publications: [2,3,4,5,9,11,12].

Discussion of 4.1.2 Tuning of predictive PI(D) algorithms

Tuning based on equivalence between predictive PI(D) algorithm and the GPC algorithm

The GPC control algorithm gives an analytic solution minimizing a quadratic cost function without constraints. The equivalence of GPC and predictive PI(D) was derived in [34] and applied in [11].

I have derived a simple tuning rule for extended horizon prediction control.

The equivalent predictive PID parameters are obtained by the following relationship:

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} -h_{n_e+1}h_{n_e+3}/N & -h_{n_e+1}h_{n_e+2}/N & 1-h_{n_e+1}^2/N \\ -h_{n_e+2}h_{n_e+3}/N & 1-h_{n_e+2}^2/N & -h_{n_e+2}h_{n_e+1}/N \\ 1-h_{n_e+3}^2/N & -h_{n_e+3}h_{n_e+2}/N & -h_{n_e+3}h_{n_e+1}/N \end{bmatrix}^{-1} \begin{bmatrix} h_{n_e+1}/N \\ h_{n_e+2}/N \\ h_{n_e+3}/N \end{bmatrix}$$

where *h* denote the points of the step response, λ_u is the weighting factor of the control increments in the cost function of GPC and $N = \lambda_u + h_{n_e+1}^2 + h_{n_e+2}^2 + h_{n_e+3}^2$.

Tuning rules for aperiodic processes

There are different tuning rules for aperiodic processes. I have considered the application of these rules for predictive PID control. An aperiodic process can be described by the following transfer function:

$P(s) = \dfrac{K_P}{(1+sT_1)(1+sT_2)...(1+sT_P)} e^{-sT_d}$, where K_P is the static gain, $T_1, T_2, ..., T_P$ are the time constants, and T_d is the dead time. T_{SUM} is defined as the sum of the time constants: $T_{SUM} = T_d + \sum_{i=1}^{P} T_i$. [36] suggests a rule of thumb for the coefficients of the continuous PID control based on K_P and T_{SUM} which can be then transformed to discrete coefficents. Applying these rules for predictive control in T_{SUM} *I did not include the dead time* T_d*, as the prediction considers it already. The gain has to be divided by the number of parallel paths (m).* [17] gives tuning rules considering also set-point weighting to attenuate too high manipulated variable values. *I applied this structure for* *predictive PI(D) controllers using set-point weighting in all parallel paths.*

The structure of the control system is given in Fig. 2.

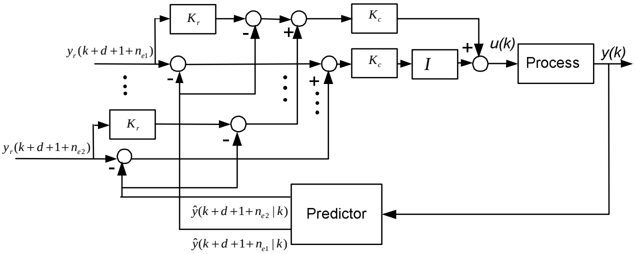

Fig. 2. Predictive PI(D) control with set-point weighting

Discussion of 4.1.3 Enhancing robustness of predictive PI(D) control using a noise model with polynomial T

The CARIMA process model is given by the following equation:

$$y(k)=\frac{B(z^{-1})}{A(z^{-1})}z^{-d}u(k)+\frac{T(z^{-1})}{A(z^{-1})(1-z^{-1})}v_u(k),$$ where $T(z^{-1})$ is a polynomial.

Generally $T(z^{-1})=1$ is considered, but if it is a polynomial, it can be treated as a filter. It can attenuate the component of prediction error caused by plant-model mismatch. If $T(z^{-1})$ is a polynomial, the prediction equations are valid for the filtered signals $y^F(z^{-1})=y(k)/T(z^{-1})$ and $\Delta u^F(z^{-1})=\Delta u(k)/T(z^{-1})$, respectively. In the prediction equation the free response is substituted by the filtered values. The control algorithm will then give the filtered value of the control increment, which has to be filtered with the inverse filter to get the actual control increment.

Enhancing robustness with Smith predictor filter

[19] and [41] suggest a structure similar to Smith predictor supplemented with a robustifying filter. I used this idea in case of predictive PI(D) controller, and also applied the noise filter and the Smith filter together.

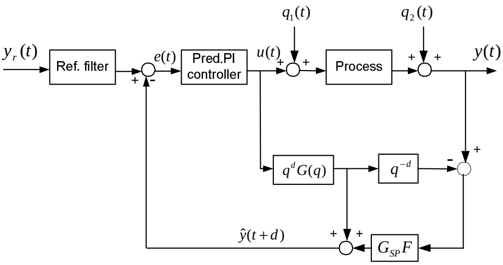

Fig. 3. Block diagram of predictive PI control enhanced by Smith predictor filter

Simulation example: The linear process has a static gain of K_P=1 and three equal time constants T_1=1/3. The sampling time is $\Delta T = 0.1$. The plant can be approximated by a first-order system with dead time. The transfer function of the plant is: $G(s) = \dfrac{1}{(1+0.333s)^3} \approx \dfrac{1}{1+1.25s} e^{-0.26s}$. The discrete tuning parameters according the T_{SUM} rule are $k_{cp} = 0.4603;\ k_{ci} = 0.0794$. The Smith predictor filter is chosen as $G_{SP}F = \dfrac{0.1}{1-0.9z^{-1}}$. The tuning parameters are: $n_{e1} = 1,\ n_{e2} = 5$. A positive unit step reference signal acts at time points 1 min and a negative step disturbance of amplitude -0.5 is applied at time point 15 or 30 min at the process input. Fig. 4 shows the controlled and the manipulated signals of the predictive and the non-predictive PI control algorithm with dead time d= 2 sec.

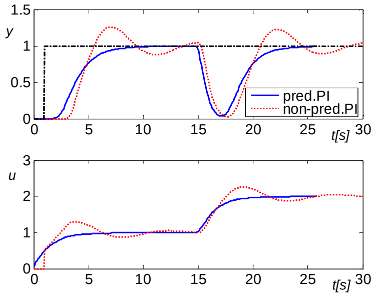

Fig.4. Output and control signals of non-predictive and predictive PI control

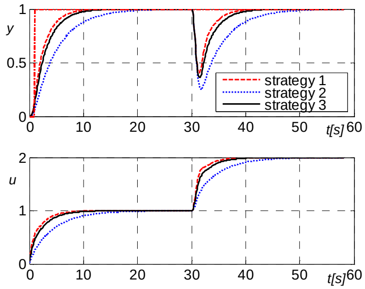

Fig.5. Output and control signals of predictive PI for different control strategies

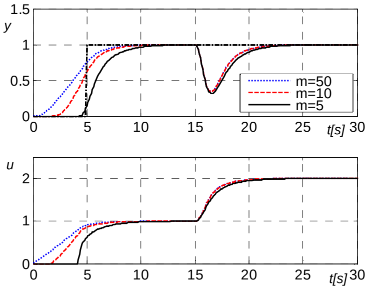

Fig.6. Output and control signals of predictive PI for different prediction horizons

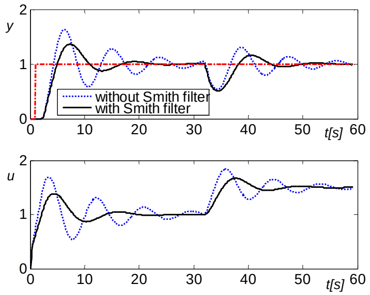

Fig.7. Predictive PI and predictive PI enhanced with Smith filter, in case of big dead time mismatch

Fig. 5 shows the controlled and the manipulated signals of the predictive PI control algorithm for different assumptions for the control horizon. Fig. 6 shows the controlled and the manipulated signals for three different prediction horizons, $n_{e2} = 5$, 10 and 50, respectively. Fig. 7 gives the controlled and the manipulated signals with big mismatch in the dead time, whose value was considered 18, while the controller was designed supposing value of 3. It is seen that the predictive PI control algorithm with the Smith filter is more robust.

4.2 Thesis 2: Enhancing decoupling properties of MIMO predictive controllers

I used some decoupling techniques for TITO processes. With the first method the reference signal change is decelerated in order to make the control slower and to reduce the coupling effect. A new filter design is recommended for calculating the modified reference signal which suppresses the effect of the disturbance in the other control variable whose set-point was kept constant. With the second method different control error weighting factors are used for the two controlled outputs.

Related publications: [7 ,15]

Discussion of 4.2

Decoupling predictive control by error dependent tuning of the weighting factors

I have analyzed also a method for dynamic decoupling. The weighting factor of the control error of the variable whose set-point is not changed should be increased related to the variable whose set-point is changed. The change of the weighting factors can be realized easily. The timing of the change of the weighting factor can be realized practically by synchronization to the set-point changes using a “set-point change detector”. Another way is to set the weighting factors as functions of the control error. If the weighting factor is set inverse proportional to the control error for both controlled variables then after a stepwise change of a reference signal the weighting factor of the output whose set-point was not changed will be higher than the weighting factor of the output whose set-point was changed. The new tuning procedure of the weighting factors works automatically, thus no extra synchronization is necessary.

Application

Distillation column model used for separating chemical petrol was simulated where the feed is chemical petrol from the desulfurization, the top product is light petrol and the bottom product is heavy petrol. The multivariable control was simulated without constraints. The sampling time was $\Delta T=1$ min. Control error prediction horizon of $n_{e11}=n_{e12}=0$ and $n_{e21}=n_{e22}=100$ were applied. The manipulated variable horizon was $n_u=25$, and the weighting of the control error and of the control increments for the first and second manipulated variables were $\lambda_{y1}=\lambda_{y2}=1$, $\lambda_{u1}=0.0012$ and $\lambda_{u2}=30.827$, respectively. The simulation is shown in Fig. 8. It is seen that the set-point change disturbs the other variable, which should be remained constant. This coupling effect can be suppressed by changing the control error weighting factor as a function of the control error. A good decoupling could be achieved by the same dependence of the weighting factors with $\lambda_{y1,max}=1$, $\lambda_{y2,max}=1$, $\lambda_{y1,damp}=25$ and $\lambda_{y2,damp}=25$. Fig. 9. shows the practically decoupled control.

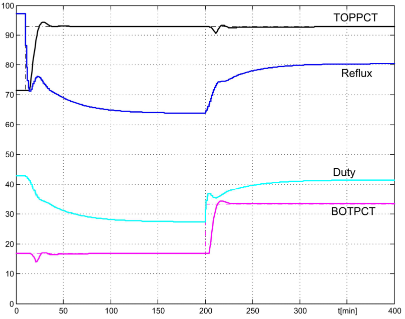

Fig. 8. Control of the TITO distillation column model with constant control error weighting factors (CV1: TOPPCT, CV2: BOTPCT, MV1: Reflux and MV2: Duty)

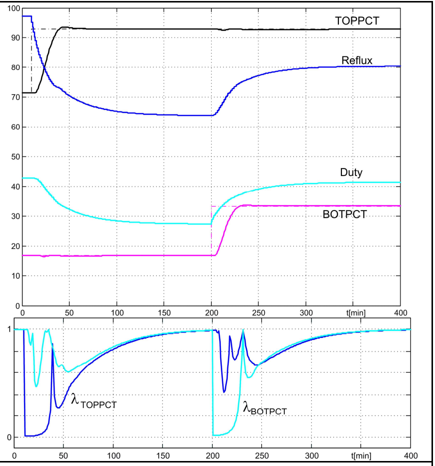

Fig. 9. Control of the TITO distillation column model with control error dependent control error weighting factors. Bottom: λ_{CV1}, λ_{CV2})

4.3 Thesis 3: A new predictive control algorithm for nonlinear systems described by the parametric Volterra model

I have suggested an iterative optimization method to control nonlinear systems identified as Volterra models. The method is based on the parametric Volterra model.

The optimization uses an iterative GPC-like technique, where the nonlinear prediction part is considered first as part of the free response.

The performance of the control algorithm is demonstrated on level control of a two-tank model.

Related publications: [10, 13]

Discussion of 4.3

In industrial practice the plants generally contain nonlinearities. Linear control algorithms provide good performance only in a small range around a working point. Control algorithms considering the nonlinear characteristics of the plant would provide better control behavior in the whole operating range, than the linear control algorithms. One way to model a smooth nonlinear plant is to approximate it by a Volterra model. A second-order Volterra model would give an acceptable approximation. Instead of using the non-parametric Volterra weighting function series in [25, 26] the parametric Volterra model is introduced in the following form:

$$A_1(q_1^{-1})y(k) = c_0 + B_1(q^{-1})u(k-d) + B_2(q_1^{-1}, q_2^{-1})u^2(k-d)$$

A_1, B_1 and B_2 are polynomials of the backward shift operator q^{-1} of degrees n_{a1}, n_{b1} and n_{b2}, respectively. In detailed form:

$$y(k) = -\sum_{j=1}^{n_{a1}} a_{1j} y(k-j) + c_0 + \sum_{i=1}^{n_{b1}} b_{1i} u(k-d-i) + \sum_{i=1}^{n_{b2}} \sum_{j=i}^{n_{b2}} b_{2ij} u(k-d-i) u(k-d-j)$$

Introducing the incremental form of the input signal and applying the predictive transformation derived in [28] the prediction form can be given as:

$$\hat{y}(k+d+1+n_e \mid k) = p_0^{(d+1+n_e)} + P_1^{(d+1+n_e)}(q^{-1})\Delta u(k+1+n_e)$$
$$+ P_2^{(d+1+n_e)}(q_1^{-1}, q_2^{-1})\Delta u^2(k+1+n_e)$$

or

$$\hat{y}(k+d+1+n_e \mid k) = \hat{y}_0(k+d+1+n_e \mid k) + \hat{y}_1(k+d+1+n_e \mid k)$$
$$+ \hat{y}_2(k+d+1+n_e \mid k)$$

with

$$\hat{y}_0(k+d+1+n_e \mid k) = p_0^{(d+1+n_e)}$$

$$\hat{y}_1(k+d+1+n_e \mid k) = [\, p_{1,n_e+1}^{(d+1+n_e)}, p_{1,n_e}^{(d+1+n_e)}, \ldots, p_{1,1}^{(d+1+n_e)}] \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+n_e) \end{bmatrix}$$

$$\hat{y}_2(k+d+1+n_e \mid k) = [\Delta u(k), \Delta u(k+1), \cdots, \Delta u(k+n_e)] \cdot$$
$$\cdot \begin{bmatrix} p_{2,n_e+1,n_e+1}^{(d+1+n_e)} & p_{2,n_e,n_e+1}^{(d+1+n_e)} & \cdots & p_{2,1,n_e+1}^{(d+1+n_e)} \\ 0 & p_{2,n_e,n_e}^{(d+1+n_e)} & \cdots & p_{2,1,n_e}^{(d+1+n_e)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_{2,1,1}^{(d+1+n_e)} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+n_e) \end{bmatrix}$$

If only n_u number of consecutive control increments are considered, vector P_1 contains only n_u elements, and the dimension of matrix P_2 is $n_u \times n_u$. $\hat{y}_0$ gives the free response, while $\hat{y}_1$ is the linear, and $\hat{y}_2$ is the quadratic part of the forced response. The prediction equations for long-range horizon have been also derived. A quadratic cost function J is minimized, which considers the quadratic deviation of the reference signal $\mathbf{y}_r$ and the output signal $\hat{\mathbf{y}}$ predicted in a future horizon and punishes also the squares of the control increments.

$$J = (\mathbf{y}_r - \hat{\mathbf{y}})^T \mathbf{\Lambda}_y (\mathbf{y}_r - \hat{\mathbf{y}}) + \Delta\mathbf{u}_f^T \mathbf{\Lambda}_u \Delta\mathbf{u}_f \Rightarrow \min_{\Delta u_f}$$

Using special manipulated variable parameterization strategies during the control horizon the optimization problem can be reduced to one-dimensional (scalar) case. This is the case e.g. if only one change in the manipulated signal is allowed and in the subsequent points the change is zero, or all increments are assumed to be equal. Without constraints the minimization leads to the solution of a third-degree algebraic equation. The control signal is chosen from among its solution, providing the smallest value for the cost function [27, 28]. [22] gives an optimization algorithm based on the nonparametric Volterra model, which has been developed further and applied in [24]. Based on this approach I have derived a new GPC-like predictive control algorithm using the parametric Volterra model.

The new iterative control algorithm

More and not necessarily equal changes in the control signal are allowed in the control horizon.

The output prediction can be described by a vector/matrix equation:

$$\hat{\mathbf{y}} = \hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_1 + \hat{\mathbf{y}}_2 = \hat{\mathbf{y}}_0 + \mathbf{P}_1^f \cdot \Delta\mathbf{u} + \hat{\mathbf{y}}_2 = \mathbf{P}_1^f \cdot \Delta\mathbf{u} + \hat{\mathbf{y}}_{free+nlin}$$

Here $\hat{\mathbf{y}}_{free+nlin} = \hat{\mathbf{y}}_0 + \hat{\mathbf{y}}_2$

contains not only the free response but also the unknown, nonlinear quadratic terms of the actual and future manipulated signal increments. Some initial values for the actual and future control increments are supposed. An iterative algorithm was introduced in [22, 38] for the predictive control of the finite-response, non-parametric Volterra model, which performs the following steps:

1. Assume initial values for the actual and future manipulated signal

increments ($\Delta\mathbf{u}$) (based e.g. on the last control step).

2. Calculate the "free + nonlinear response" term $\hat{\mathbf{y}}_{free+nlin}$.
3. Calculate the new manipulated variable sequence

   - in the unconstrained case by

   $$\Delta\mathbf{u} = [\mathbf{P}_1^{fT}\mathbf{\Lambda}_y\mathbf{P}_1^f + \mathbf{\Lambda}_u]^{-1}\mathbf{P}_1^{fT}\mathbf{\Lambda}_y(\mathbf{y}_r - \hat{\mathbf{y}}_{free+nlin})$$

   - in the constrained case e.g. by quadratic programming considering the constraints

   $$\Delta u_{\min} \le \Delta u(k+j) \le \Delta u_{\max}; \quad j = 0,1,\dots,n_u - 1$$
4. Check whether the new manipulated sequence differs from the previously calculated (or guessed) one. If the difference is larger than a predefined threshold vector, continue the iterative algorithm from Step 2 with the newly calculated future control signal sequence, otherwise terminate the iteration.

In [22, 38] a "mixed" predictive model description is used, that means incremental terms in the linear part ($\hat{\mathbf{y}}_1$) and non-incremental terms in the quadratic part ($\hat{\mathbf{y}}_2$). In [24] it is shown that this algorithm can be applied if the process is approximated and identified by a parametric Volterra model and the control is realized based on the nonparametric Volterra series, using transformation equations between the parametric Volterra model and the finite response Volterra series [24, 25]. The new algorithm presented here is based on the publications mentioned and has the following advantages:

- The control is based on the parametric Volterra model. (Parametric Volterra models approximate nonlinear dynamic processes with less parameters than the non-parametric models.)
- Both the linear and the quadratic terms of the controlled output are predicted based on the control signal increments.

Related publications: [10, 13].

Application

The sub-optimal algorithms were applied on a ball and beam model [33, 10] while the new nonlinear algorithm was applied on the model of a two-tank system [13]. Both systems are laboratory set-ups at the Department of Automation and Applied Informatics, BME. The parameters were obtained from simple measurements on the real systems.

Two-tank system

The identified Volterra model of the tank is given by the following relationship:

$$y(k) = 0.0017 + 1.5358y(k-1) - 0.6176y(k-2) + 0.0695u(k-1) - 0.0697u(k-2)$$
$$- 0.0195u^2(k-1) + 0.0593u(k-1)u(k-2) + 0.0163u^2(k-2)$$

From this model a quadratic polynomial static characteristics is calculated in the following form:

$$y_{steady} = 0.0208 - 0.0034u_{steady} + 0.6861u_{steady}^2$$

Fig.10. gives the predictive control of the two-tank system with the iterative and the suboptimal algorithms. Fig. 11. shows the effect of the prediction horizon with the iterative algorithm. The tuning parameters are $\lambda_u = 1$, $n_{e1} = 1$ and $n_{e2} = 1, 5, 10$. It is seen that with longer horizon the control signal starts before the change of the reference signal and the output signal reaches its required value smoothly. The suggested nonlinear predictive algorithm is promising, providing good control results in the whole operating range.

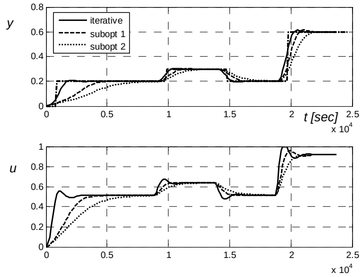

Fig. 10. Predictive control of the two-tank system with the different algorithms

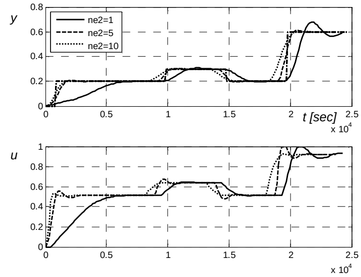

Fig. 11. Predictive control of the two-tank system with the iterative algorithm

Bibliography

[1] **Arousi, F.**, Bars, R., Haber, R. (2005). Predictive PI controller based on first-order dead time models. 5th Int. Carpathian Control Conference ICCC'2005, Miskolc-Lillafüred, Hungary, Vol. 1, 113-118.

[2] **Arousi, F**., Bars, R., Haber, R. (2006). Predictive PI controller based on first-order and second-order models with dead time. Workshop on System Identification & Control Systems in honor of L. Keviczky on his 60th Birthday, MTA Kiadvány, 167-181.

[3] **Arousi, F**., Bars, R., Haber, R. (2006). Optimization of PI controller for a predicted cost function assuming a first-order process with dead time. 13th IFAC Workshop on Control Applications of Optimisation, Paris-Cachan, France

[4] Schmitz, U., Haber, R., Bars, R., **Arousi, F**. (2006). Simple predictive PI controllers for aperiodic process, 7th Int. Conference on Process Control, PC'06, Kouty nad Desnou, Czech Repuplic, paper R175b1-12.

[5] **Arousi, F**., Bars, R., Haber, R., Schmitz, U. (2007). Predictive PI controller with set-point weighting. 8th Int. Carpathian Control Conference, ICCC'2007, Strbské Pleso, Slovak Republic, 32-35.

[6] Honc, D., **Arousi, F**., Haber, R. (2007). TITO decentralized predictive control using static decoupling compensators. Int. Carpathian Control Conference, ICCC'2007, Strbské Pleso, Slovak Republic, 179-182.

[7] Schmitz, U., Haber, R., **Arousi, F**., Bars, R. (2007). Decoupling predictive control by error-dependent tuning of the weighting factor. 16th Int. Conference on Process Control, PC'07, Strbské Pleso (Poprad), High Tatra, Slovak Republic

[8] Honc, D., Haber, R., **Arousi, F**. (2007). TITO decentralized PI control using static decoupling compensators. 16th Int. Conference on Process Control, PC'07, Strbské Pleso, Slovak Republic

[9] **Arousi, F**., Bars, R., Haber, R., Schmitz, U. (2007). Predictive PI controller with set-point weighting based on first-order model with dead time. 2th Automation and Applied Computer Science Workshop AACS'07, Budapest, Hungary, 239-252.

[10] **Arousi, F**., Bars, R., Haber, R. (2008). Suboptimal nonlinear Volterra based predictive control algorithms applied to the model of a pilot plant. 3th Automation and Applied Computer Science Workshop, AACS'07, Budapest, Hungary, 245-257.

[11] **Arousi, F**., Bars, R., Haber, R. (2008). Tuning of predictive PID algorithms according to GPC equivalence, 8th Int. Conference on

Process Control, PC'08, Kouty nad Desnou, Czech Republic, paper C080a1-11.

[12] **Arousi, F**., Schmitz, U., Bars, R., Haber, R. (2008). Robust predictive PI controller based on first-order dead time model. 17th IFAC World Congress, South Korea

[13] Haber, R., Bars, R., **Arousi, F**. (2009). An iterative GPC-like nonlinear predictive control algorithm for parametric Volterra model. ECC'09, Budapest, Hungary. Tuc15.5, 3383-3388.

[14] **Arousi, F**., Bars, R., Haber, R. (2007). Optimization of PI controller for a predicted cost function assuming a first-order process with dead time. International Journal of Tomography & Statistic (IJTS), Vol. 5, No. W07, Winter 2007, 155-160.

[15] Schmitz, U., Haber, R., **Arousi, F**., Bars, R. (2007). Decoupling predictive control by error-dependent tuning of the weighting factor. AT&P Journal PLUS 2, 131-140.

[16] **Arousi, F**. (2000). Predictive control algorithms applied for nonlinear systems, M.Sc. Diploma Project, BME

[17] Åström, K. J., Hägglund, T. (2006). Advanced PID Control. ISA – The Instrumentation, Systems and Automation Society

[18] Bego, O., Peric, N., Petrovic, I. (2000). Decoupling multivariable GPC with reference observation. 10th Mediterranean Electromechanical Conf., Vol. II, 819-822.

[19] Camacho, E. F., Bordons, C. (1999, 2004). Model Predictive Control, Springer-Verlag, 280.

[20] Chien, K.L., Hrones, J.A., Reswick, J.B. (1952). On the automatic control of generalized passive systems. Trans ASME. Vol. 74, 175-185.

[21] Clarke, D.W., Mohtadi, C., Tuffs, P. S. (1987). Generalized Predictive Control – Part I. The basic algorithm. Automatica, 24 (2), 137-148.

[22] Doyle, F. J., Pearson, R.K., Ogunnaike, B.A. (2001). Identification and control using Volterra models. Springer-Verlag

[23] Findeisen, R., Allgöwer, F., Lorenz, T. (2007). Assessment and future directions of nonlinear model predictive control. Springer-Verlag, ISBN 3540726985, 9783540726982

[24] Gruber, J. K., Bordons, C., Bars, R. and Haber, R. (2009). Nonlinear predictive control of smooth nonlinear systems based on Volterra models. Application to a pilot plant. Submitted for publication to International Journal of Robust and Nonlinear Control

[25] Haber, R., Keviczky, L.: Nonlinear System Identification Input-Output Modeling Approach. Volume 1: Nonlinear System Parameters Identification (1999). ISBN-0-7923-5857-0; ISBN 0-7923-5858-9 set

[26] Haber, R., Keviczky, L. (1999). Nonlinear System Identification-Input-Output Modeling Approach. Volume 2: Nonlinear System Structure Identification. ISBN-0-7923-5856-2; ISBN 0-7923-5858-9 set

[27] Haber, R. (1995). Predictive control of nonlinear dynamic processes. Applied Mathematics and Computation, Vol. 70. 169-184.

[28] Haber, R. (1992). Transformation of the absolute descriptions of linear and non-linear processes to their incremental forms. International Journal of Systems Science, 23, 6, 935-955.

[29] Haber, R., Bars, R., Lengyel, O. (1999a). Sub-optimal nonlinear predictive and adaptive control based on the parametric Volterra model. AMCS Applied Mathematics and Computer Science, Vol. 9, No 1, 161-173.

[30] Haber, R., Bars R., Lengyel, O. (1999b). Nonlinear predictive control algorithms with different input sequence parameterizations applied for the quadratic Hammerstein and Volterra models. In Allgöwer, F., and A. Zheng (editors): Nonlinear Predictive Control. Birkhäuser Verlag, Basel, 347-356.

[31] Haber, R., Bars R., Lengyel, O. (1999c). Three extended horizon adaptive nonlinear predictive control schemes based on the parametric Volterra model. European Control Conference, ECC'99, Karlsruhe. Paper No. F625 on the CD ROM.

[32] Honeywell Process Solution. Profit Suit Solutions. Product Brochure, May 2007.

[33] Ioannis Tziortzis (2007). Real time control of the ball and beam system. B.Sc. Diploma project, BME

[34] Johnson, M.A., Moradi, M.H. (2005). PID Control, Chapter 13. Springer-Verlag, 543.

[35] Katebi, M.R., Moradi, M.H. (2001). Predictive PID controllers, in IEE Proc.-Control Theory Appl. Vol. 148, No. 6, Nov. 478-487.

[36] Kuhn, U. (1995). A practical tuning rule for PID controllers: the T-sum rule (in German). Automatisierungstechnische Praxis, Vol. 37, No. 5, 10-16.

[37] Lengyel, O. (1998). Predictive control algorithms for nonlinear model. M.Sc. Diploma project, BME (in Hungarian)

[38] Maner, B. R., Doyle, F. J., Ogunnaike, B. A., Pearson, R. K. (1996). Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models, Automatica, Vol. 32, No 9, 1285-1301.

[39] Maurath, P. R., Seborg, D. E., Mellichamp, D. A. (1986). Achieving decoupling with predictive controllers. Proc. Amer. Control Conf., Seattle, 1372-1377.

[40] Morari, M., Zafiriou, E. (1989). Robust process control. *Prentice-Hall*

[41] Normey-Rico, J.E., Camacho, E.F. (2007). Control of dead-time processes, Springer-Verlag, 462.

[42] Qin S.J., Badgwell TA. (2003). A survey of industrial model predictive control technology. Control Engineering Practice. 11(7), 733-764.

[43] Maciejowski, J.M. Predictive control with constraints. Prentice-Hall, Pearson Education Limited, Harlow, UK, 2002, ISBN 0-201-39823-0