Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Robust Predictive PI Controller Based on First-Order Dead Time Model

F. Arousi*, U. Schmitz, R. Bars*, R. Haber*****

*Dept. of Automation and Applied Informatics, Budapest University o f Technology and Economics, MTA-BME Control Research group (e-mail: fakhre_arousi@hotmail.com, bars@aut.bme.hu)
**Dept. of Advanced Process Control, Shell Rheinland Refinery, Köln/Wesseling, Germany (e-mail: ulrich.schmitz@shell.com)
***Inst. of Chemical Process Engineering and Plant Design, University of Applied Science Cologne, Germany (e-mail: robert.haber@fh-koeln.de)

Abstract: Predictive control algorithms compute the manipulated variable minimizing a cost function considering expected future errors. PI control algorithms can be equipped with predictive properties. Simple predictive control algorithms are derived using approximation of an aperiodic process by a first-order model with dead time. Applying a noise model the robustness properties of the algorithm are enhanced considering plant-model mismatch. The noise filter is considered as a design parameter. Simulation examples demonstrate the behavior of the predictive PI algorithm and the robustifying effect of the noise filter.

1. INTRODUCTION

The most widely used algorithms in practice are the PI(D) control algorithms. The algorithms are simple, and with three effects (proportional, integrating and differentiating) generally the quality specifications prescribed for the control system can be met.

Nevertheless in case of big dead time in the process the performance of the control system will be slow, the PI(D) controller can not accelerate the control system significantly. There are some discrete control algorithms as Smith predictor or dead-beat control, which provide faster performance than PI(D) control for dead time systems, but these algorithms did not get really a wide industrial acceptance because of their sensitivity against plant/model mismatch (Normey-Rico, Camacho, 2007).

Predictive control algorithms where predicted error values are used to calculate the actual manipulated variable are also widely applied. Predictive algorithms provide good performance especially in case of big dead time and if the future reference trajectory is known. Applications of predictive control algorithms are supported by different industrial software packages. Nowadays besides PI(D) control predictive control is getting an increased use.

As operators of industrial process control systems are familiar with PI(D) controllers and have expertise in PI(D) controller tuning, it would be advantageous to enhance the performance of the PI(D) controllers with predictive properties, while applying the well accepted PI(D) tuning rules. In this way the operator will see a PI(D) controller with hidden predictive properties.

The properties of the two algorithms – predictive and PI(D) - can be combined. The idea of predictive PI(D) controllers was initiated by Katebi and Moradi (2001) and Johnson and Moradi (2005).

2. PREDICTIVE PI(D) CONTROL STRUCTURE

A predictive PI(D) controller can consider not only one predicted output signal, but a series of predicted output values. Katebi and Moradi suggested *m* number of parallel connected PI(D) 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. Here $\hat{e}(k+d+i\,|\,k)$ denote the predicted values of the error signal *i* step ahead over the dead time. $y_r(k+d+i)$ is the reference signal and $\hat{y}(k+d+i\,|\,k)$ is the predicted output signal. n_{e1} is the first point of the prediction horizon over the dead time, while n_{e2} is the last point of the prediction horizon, and the number of the parallel paths is $m = n_{e2} - n_{e1} + 1$.

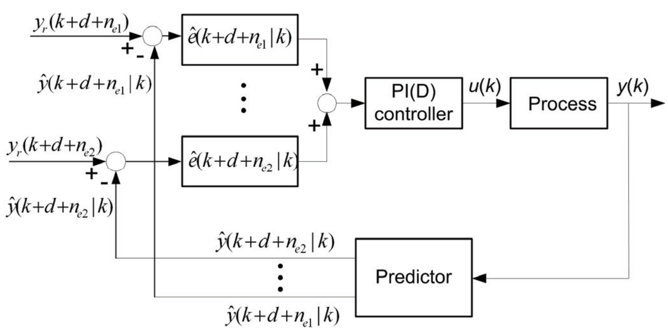

Fig.1. Predictive PID controller with parallel paths

3. THE PROCESS MODEL

The process model is given by the following equation:

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})} z^{-d} u(k) + \frac{T(z^{-1})}{A(z^{-1})(1-z^{-1})} v_u(k) \qquad (1)$$

This is the so called CARIMA model, where

$$B(z^{-1}) = b_0 + b_1 z^{-1} + ... + b_{n-1} z^{-(n-1)}$$

$$A(z^{-1}) = 1 + a_1 z^{-1} + ... + a_n z^{-n}$$

$$T(z^{-1}) = 1 + t_1 z^{-1} + ... + t_{nt} z^{-nt}$$

The output is affected by the input signal $u(k)$ and the disturbance $v_u(k)$. d is the discrete dead time. In most cases $T(z^{-1}) = 1$ 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 the model mismatch, which is particularly important at high frequency. The high frequency disturbances are mainly due to the presence of high frequency components in unmodeled dynamics and unmeasurable load disturbances. If there is no unmodeled dynamics, the effect of polynomial T is rejection of disturbances, with no influence on reference tracking. In this case polynomial T can be used to detune the response to unmeasurable high-frequency load disturbances, preventing excessive control action. On the other hand, T is used as a design parameter that can influence robust stability. In this case the predictions will not be optimal, but robustness in the face of uncertainties can be achieved. Then this polynomial can be considered as prefilter or as an observer (Camacho, Bordons, 2004). It can play an essential role in the robust realization of predictive PI(D) controllers as well.

4. PREDICTIVE PI CONTROL ALGORITHM

The form of a non-predictive discrete PI controller is

$$u(k) = K_P e(k) + K_I \sum_{i=1}^{k} e(i) \tag{2}$$

where e denotes the error signal and K_P, K_I are the coefficients of the proportional and the integral components, respectively. Taking the difference on both sides of (2) at step k and (k-1) leads to

$$\begin{aligned} \Delta u(k) = u(k) - u(k-1) = K_P\left[e(k) - e(k-1)\right] + \\ + K_I e(k) = (K_P + K_I) e(k) - K_P e(k-1) \end{aligned} \tag{3}$$

In predictive PI control the manipulated variable is the sum of the controller outputs based on the predicted control errors. Applying the algorithm on a future error signal d+i step ahead of the actual time point the corresponding control increment $\Delta u_i(k)$ is obtained as

$$\begin{aligned} \Delta u_i(k) = (K_P + K_I)\hat{e}(k+d+i \mid k) - \\ - K_P \hat{e}(k+d+i-1 \mid k) \end{aligned} \tag{4}$$

The future error signals are predicted on the basis of the information available till the actual time point k.

Let us introduce the following vector notations:

$$\mathbf{K} = [-K_P \quad K_P + K_I] \tag{5}$$

$$\hat{\mathbf{E}}(k+d+i) = \begin{bmatrix} \hat{e}(k+d+i-1) \\ \hat{e}(k+d+i) \end{bmatrix} =$$
$$= \begin{bmatrix} y_r(k+d+i-1) - \hat{y}(k+d+i-1) \\ y_r(k+d+i) - \hat{y}(k+d+i) \end{bmatrix} =$$
$$= \mathbf{Y}_r(k+d+i) - \hat{\mathbf{Y}}(k+d+i) \tag{6}$$

where

$$\mathbf{Y}_r(k+d+i) = \begin{bmatrix} y_r(k+d+i-1) \\ y_r(k+d+i) \end{bmatrix} \tag{7}$$

is the vector composed of the future reference signal values and

$$\hat{\mathbf{Y}}(k+d+i) = \hat{\mathbf{Y}}_{forced}(k+d+i) + \hat{\mathbf{Y}}_{free}(k+d+i) =$$
$$= \begin{bmatrix} \hat{y}(k+d+i-1) \\ \hat{y}(k+d+i) \end{bmatrix} \tag{8}$$
$$= \begin{bmatrix} \hat{y}_{forced}(k+d+i-1) + \hat{y}_{free}(k+d+i-1) \\ \hat{y}_{forced}(k+d+i) + \hat{y}_{free}(k+d+i) \end{bmatrix}$$

is the output vector built from the consecutive points of the predicted output signal, which is composed of the forced and the free responses.
With these notations

$$\Delta u_i(k) = [-K_P \quad K_P + K_I] \begin{bmatrix} \hat{e}(k+d+i-1) \\ \hat{e}(k+d+i) \end{bmatrix} \tag{9}$$
$$= \mathbf{K}\hat{\mathbf{E}}(k+d+i)$$

The control increment $\Delta u(k)$ is the sum of the increments in the individual controller paths. Taking into consideration (9) the control increment can be expressed as

$$\Delta u(k) = \mathbf{K}\begin{bmatrix} \hat{\mathbf{E}}(k+d+n_{e1}+1) + \\ + \hat{\mathbf{E}}(k+d+n_{e1}+2) + ... + \hat{\mathbf{E}}(k+d+n_{e2}) \end{bmatrix}$$
$$= \mathbf{K} \sum_{i=n_{e1}+1}^{n_{e2}} \hat{\mathbf{E}}(k+d+i) \tag{10}$$

It has to be mentioned, that the first point of the prediction horizon was chosen for $n_{e1}+1$ in order to start the calculations from the error signal predicted in sampling point $k+d+n_{e1}$.

In expression (10) the predicted error values are calculated as the difference between the predicted reference and the predicted output signals. The predicted output values contain effects of the forced response and the free response. The forced response contains the effect of the actual and the subsequent input increments. Thus the future control increments appear also on the right side of the expression. A closed form to calculate $\Delta u(k)$ can be obtained simply only if some assumptions are considered for the future control increments, e.g.

$$\Delta u(k+i)=0 \text{ for } i\rangle 0 \tag{11}$$

is supposed. (Another assumption can be that a given number of equal subsequent input increments is taken into account.)

With assumption (11) expression (10) can be given in detailed form as

$$\Delta u(k)=\mathbf{K}\begin{bmatrix}\sum_{i=n_{e1}}^{n_{e2}-1} y_r(k+d+i)\\ \sum_{i=n_{e1}+1}^{n_{e2}} y_r(k+d+i)\end{bmatrix}-\mathbf{K}\begin{bmatrix}\sum_{i=n_{e1}}^{n_{e2}-1} h_i\\ \sum_{i=n_{e1}+1}^{n_{e2}} h_i\end{bmatrix}\Delta u(k)-$$
$$-\mathbf{K}\begin{bmatrix}\sum_{i=n_{e1}}^{n_{e2}-1} \hat{y}_{free}(k+d+i)\\ \sum_{i=n_{e1}+1}^{n_{e2}} \hat{y}_{free}(k+d+i)\end{bmatrix} \tag{12}$$

where h_i are the points of the step response. The second term at the right side of (12) is the effect of the forced response, while the third term is the free response, the effect of the past inputs on the future output signal, where the input signal is frozen at point k-1.

Let us introduce the following notations:

$$\mathbf{h}_{sum}=\begin{bmatrix}\sum_{i=n_{e1}}^{n_{e2}-1} h_i\\ \sum_{i=n_{e1}+1}^{n_{e2}} h_i\end{bmatrix};\mathbf{Y}_{rsum}=\begin{bmatrix}\sum_{i=n_{e1}}^{n_{e2}-1} y_r(k+d+i)\\ \sum_{i=n_{e1}+1}^{n_{e2}} y_r(k+d+i)\end{bmatrix};$$

$$\hat{\mathbf{Y}}_{freesum}=\begin{bmatrix}\sum_{i=n_{e1}}^{n_{e2}-1} \hat{y}_{free}(k+d+i)\\ \sum_{i=n_{e1}+1}^{n_{e2}} \hat{y}_{free}(k+d+i)\end{bmatrix} \tag{13}$$

From (12) the control increment can be expressed as

$$\Delta u(k)=(1+\mathbf{K}\mathbf{h}_{sum})^{-1}\mathbf{K}(\mathbf{Y}_{rsum}-\hat{\mathbf{Y}}_{freesum}) \tag{14}$$

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.

4.1 Predictive PI control of a first-order process with dead time

Aperiodic processes can be approximated well by a first-order process with dead time. In the process industries a lot of processes can be described by this model. In most cases the step response of the system can be measured easily even within industrial circumstances. A good, but slow control of this process can be achieved by a PI controller. Different practical tuning rules are given considering the parameters of the approximating first order model of the process. Applying predictive PI controller can improve the performance of the control system. For this process control algorithm (14) can be expressed in analytical form.

The first-order system is described by the following CARIMA model:

$$y(k)=\frac{b_1 z^{-1}}{1+a_1 z^{-1}}z^{-d}u(k)+\frac{T(z^{-1})}{(1-z^{-1})(1+z_1 q^{-1})}v_u(k) \tag{15}$$

First let us consider $T(z^{-1})=1$. The predictive equations are given in (7) and (8).

Vector $\hat{\mathbf{Y}}$ can be expressed as

$$\hat{\mathbf{Y}}(k+d+i)=\begin{bmatrix}\hat{y}(k+d+i-1)\\ \hat{y}(k+d+i)\end{bmatrix}=\begin{bmatrix}h_{i-1} & h_{i-2} & \cdots & h_1 & 0\\ h_i & h_{i-1} & \cdots & h_2 & h_1\end{bmatrix}\cdot$$
$$\cdot\begin{bmatrix}\Delta u(k)\\ \Delta u(k+1)\\ \vdots\\ \Delta u(k+i-2)\\ \Delta u(k+i-1)\end{bmatrix}+\begin{bmatrix}f_1[d+i-1] & f_2[d+i-1]\\ f_1[d+i] & f_2[d+i]\end{bmatrix}\begin{bmatrix}y(k)\\ y(k-1)\end{bmatrix}$$
$$+\begin{bmatrix}h_i & h_{i+1} & \cdots & h_{i+d-1}\\ h_{i+1} & h_{i+2} & \cdots & h_{i+d}\end{bmatrix}\begin{bmatrix}\Delta u(k-1)\\ \Delta u(k-2)\\ \vdots\\ \Delta u(k-d)\end{bmatrix} \tag{16}$$

where h_i are the points of the step response, and $f_1[d+i]$, $f_2[d+i]$ are the coefficients in row $d+i$ of the following $\mathbf{f}_1$ and $\mathbf{f}_2$ vectors (Arousi et al., 2006):

$$\mathbf{f}_1=\begin{bmatrix}1-a_1\\ 1-a_1+a_1^2\\ 1-a_1+a_1^2-a_1^3\\ \vdots\end{bmatrix};\qquad \mathbf{f}_2=\begin{bmatrix}a_1\\ (1-a_1)a_1\\ (1-a_1+a_1^2)a_1\\ \vdots\end{bmatrix} \tag{17}$$

Let us write equation (16) in the following form:

$$\hat{\mathbf{Y}}(k+d+i)=\mathbf{H}_i^f\Delta\mathbf{u}^f+\mathbf{F}_{yi}^p\mathbf{y}^p+\mathbf{H}_i^p\Delta\mathbf{u}^p \tag{18}$$

where

$$\mathbf{H}_i^f = \begin{bmatrix} h_{i-1} & h_{i-2} & \cdots & h_1 & 0 \\ h_i & h_{i-1} & \cdots & h_2 & h_1 \end{bmatrix};$$

$$\mathbf{F}_{yi}^p = \begin{bmatrix} f_1[d+i-1] & f_2[d+i-1] \\ f_1[d+i] & f_2[d+i] \end{bmatrix};$$

$$\mathbf{H}_i^p = \begin{bmatrix} h_i & h_{i+1} & \cdots & h_{i+d-1} \\ h_{i+1} & h_{i+2} & \cdots & h_{i+d} \end{bmatrix}$$

and

$$\Delta\mathbf{u}^f = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+i-2) \\ \Delta u(k+i-1) \end{bmatrix}; \quad \mathbf{y}^p = \begin{bmatrix} y(k) \\ y(k-1) \end{bmatrix}; \tag{19}$$

$$\Delta\mathbf{u}^p = \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-d) \end{bmatrix}$$

If only one change is supposed in the control signal, $n_u = 1$, then $\mathbf{H}_i^f = \begin{bmatrix} h_{i-1} \\ h_i \end{bmatrix}$.

The control algorithm can be written according to (14), where the free response is obtained by the second and the third terms of the right side of (16).

If n_{e1} is at least 1, in $\hat{\mathbf{Y}}$ only predicted components (starting from $y(k+d)$, $y(k+d+1)$) are considered.

The control increment is obtained as

$$\Delta u(k) = (1 + \mathbf{K}\mathbf{h}_{sum})^{-1}\mathbf{K}(\mathbf{Y}_{rsum} - \hat{\mathbf{Y}}_{freesum})$$
$$= (1 + \mathbf{K}\sum_{i=n_{e1}+1}^{n_{e2}} \mathbf{H}_i^f)^{-1}\mathbf{K}(\mathbf{Y}_{rsum} - \sum_{i=n_{e1}+1}^{n_{e2}} \mathbf{F}_{yi}^p \mathbf{y}^p - \sum_{i=n_{e1}+1}^{n_{e2}} \mathbf{H}_i^p \Delta\mathbf{u}^p) \tag{20}$$

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 (16) the free response, the two last terms on the right side are substituted by the filtered values. The control algorithm (20) 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. This filtering procedure has a robustifying effect in case of plant-model mismatch.

5. TUNING OF PREDICTIVE PI ALGORITHMS

PI(D) controller tuning rules can be applied for predictive PI(D) algorithms. Different tuning rules are available mainly for continuous PI(D) controllers which can be considered as rules of thumb. These rules can be used for discrete controllers as well after discretization.

For predictive PID control $n_{e2} - n_{e1} + 1$ parallel controller paths are considered. If tuning is done considering continuous control, the continuous gain has to be divided by the number of the paths.

5.1 Tuning rules for aperiodic processes

There are different tuning rules for aperiodic processes. An aperiodic process can be described by the following transfer function:

$$P(s) = \frac{K_S}{(1+sT_1)(1+sT_2)...(1+sT_P)} e^{-sT_d} \tag{21}$$

where K_S 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 and the dead time:

$$T_{SUM} = T_d + \sum_{i=1}^{P} T_i \tag{22}$$

Kuhn (1995) suggests the following rules of thumb for the coefficients of the continuous PID controller (Table1):

Table 1. PI(D) controller tuning rules according to Kuhn

	PI	*PID*
K_C	$0.5/K_S$	$1/K_S$
T_I	$0.5T_{SUM}$	$0.66T_{SUM}$
T_D	0	$0.167T_{SUM}$

These coefficients have to be then discretised. For a PI continuous controller the continuous controller algorithm is

$$u(t) = K_C\left[e(t) + \frac{1}{T_I}\int_0^t e(t)dt\right] \tag{23}$$

The discrete control increment is expressed as

$$\Delta u(k) = p_0 e(k) + p_1 e(k-1) \tag{24}$$

Applying the trapezoid rule for approximating the integration the coefficients of the discrete PI algorithm are obtained as

$$p_0 = \frac{K_C}{m}(1 + \frac{\Delta T}{2T_I}); \quad p_1 = -\frac{K_C}{m}(1 - \frac{\Delta T}{2T_I}) \tag{25}$$

where ΔT denotes the sampling time.

It has to be emphasized that the continuous gain factor is divided by the number of the predictive paths.

Considering (3) the discrete controller parameters are calculated as

$$K_P = -p_1; \quad K_I = p_0 + p_1 \tag{26}$$

For predictive PI controller tuning $T_{SUM} = \sum_{i=1}^{P} T_i$, so the physical dead time is not taken into account, consideration of m number of prediction paths will take the effect of the dead time into account.

6. SIMULATION RESULTS

Matlab programs have been written to realize the control algorithms. The simulation results are demonstrated through a simple example.

The linear process has a static gain of K_S=1 and three equal time constants T_l=1/3, and 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:

$$G(s) = \frac{1}{(1+0.333s)^3} \approx \frac{1}{1+1.25s} e^{-0.26s}$$

The first-order approximation is calculated considering the initial tangent of the step response.

The T polynomial is chosen as $(1-0.7z^{-1})/0.3$.

The tuning parameters for $n_{e1}=1, n_{e2}=5$ taking into account Table 1 and (25) are

$$p_0 = (0.5/5)(1+0.1/(2\cdot 0.5)) = 0.11$$

$$p_1 = (-0.5/5)(1-0.1/(2\cdot 0.5)) = -0.09$$

Applying (26) the discrete tuning parameters are

$$K_P = 0.09; K_I = 0.02.$$

Including physical dead time the tuning parameters are the same. Different dead times (0, 0.5, 1, 2 and 5) are considered in the process.

In the simulation a positive unit step reference signal acts at time point 1, and a negative unit step disturbance is applied at time point 15. No prediction of the reference signal is taken into account. Fig. 2. shows the output and the control signals when the system is of first-order, and its model is accurate, also of first-order with the same parameters. T polynomial is not applied. It is seen, that the quality of the control with a stepwise reference signal change is the same for all dead time cases, the outputs are shifted appropriately, while the control signal is the same. Disturbance rejection depends on the dead time. Fig. 3. gives the output and the control signals when the system is of third-order with the dead times above, and the controller is designed according to the first-order approximation. T polynomial is not applied. The performance is worse than before, and also with bigger dead time the dynamics is also affected.

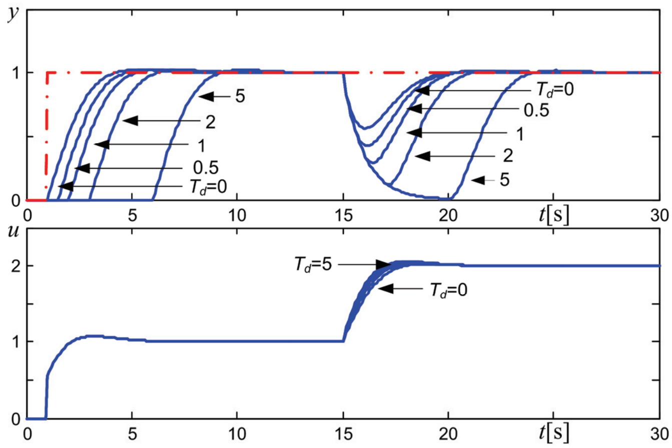

Fig.2. Output and control signals without mismatch, with first-order system and model.

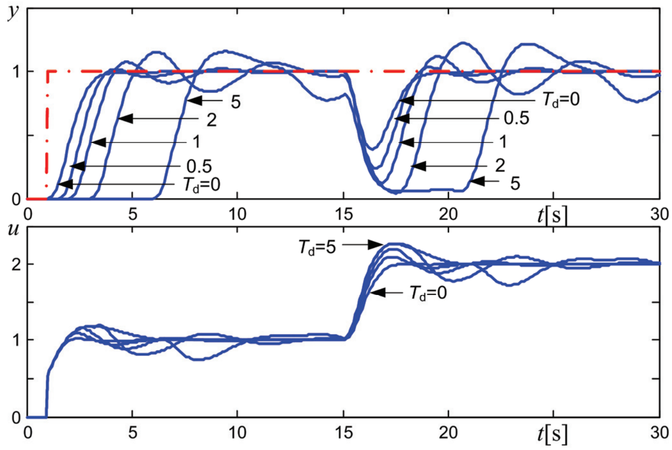

Fig.3. Output and control signals with mismatch, the system is of third-order, the controller is designed based on first-order approximation.

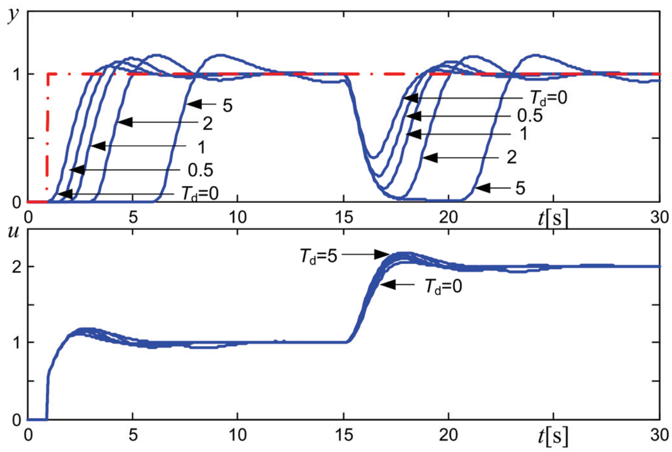

Fig.4. Output and control signals, the system is of third-order, the controller is based on first-order model,with T polynomial

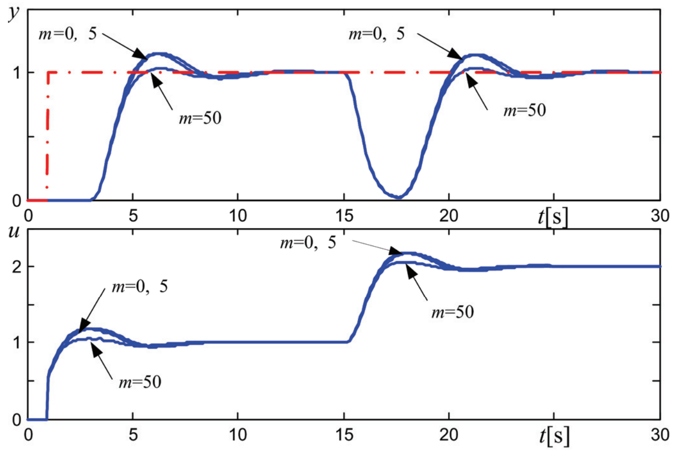

Fig.5. Output and control signals, the system is of third-order with dead time 2, the controller is based on first-order model, different prediction ranges (*m*=0, 5,50), with *T* polynomial

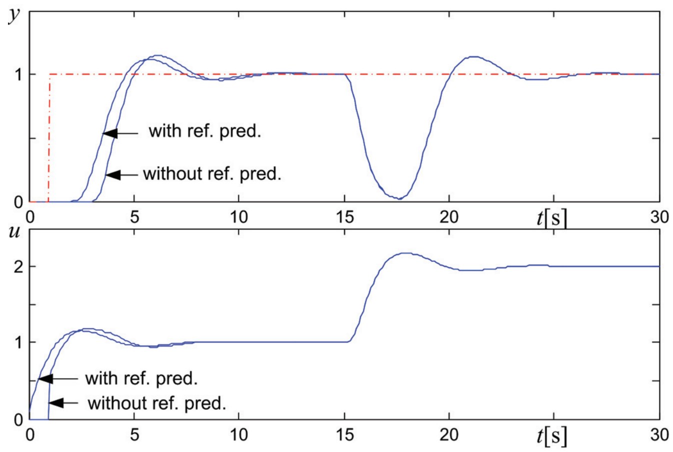

Fig.6. Output and control signals of third-order with dead time 2, the controller is based on first-order model, without and with prediction of the reference signal.

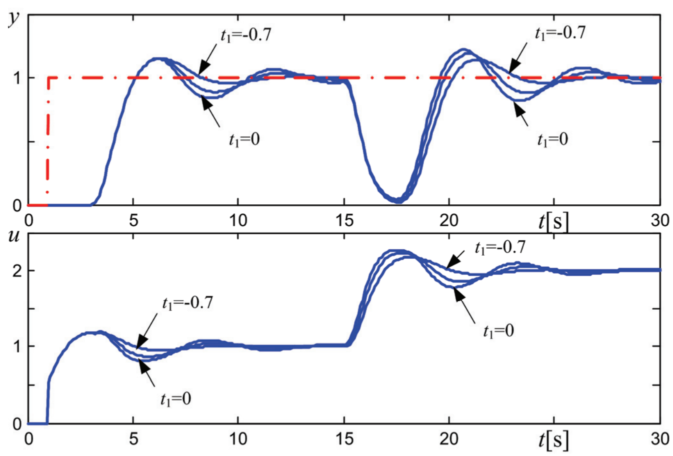

Fig.7. Output and control signals, the system is of third-order, the controller is based on first-order approximation, with different *T* polynomials (with $t_1 = 0$, $t_1 = -0.5$, $t_1 = -0.7$).

Fig. 4. shows the effect of including the *T* polynomial for the former case. This modification improves the performance in case of plant-model mismatch (here $t_1 = -0.7$).

Fig. 5. demonstrates, that increasing the prediction horizon works also against mismatch.

Fig. 6. shows that with predicted reference signal the control signal acts before the change of the reference signal, accelerating the output transient.

Fig. 7. gives the performance with different *T* polynomials. It is expected, that with further tuning of the *T* polynomial the effect of the mismatch could be decreased further.

7. CONCLUSION

PI control algorithms with predictive property have been derived based on a first-order model with dead time. For the disturbance model a *T* polynomial is taken also into account. Practically real processes frequently can be approximated by these models. Parallel connected PI controllers are applied which calculate the manipulated signal based on the predicted values of the error signal. Simple tuning rules are used. Predictive property of the algorithms compensates the dead time. Simulation results show the effectiveness of the predictive PI algorithm with *T* polynomial, which improves the robust performance in case of plant/model mismatch. This approach can be effective also if there is a mismatch in the dead time as well.

ACKNOWLEDGEMENT

The authors' work from BME was supported by the fund of the Hungarian Academy of Science for control research and partly by the OTKA fund T68370. The collaboration of the two Universities has been supported by the program of EU-Socrates. All supports are kindly acknowledged.

REFERENCES

Arousi, F., R. Bars, R. Haber (2006). Predictive PI(D) controllers based on first- and second-order models with dead time models, in *Proc. of the Workshop on System Identification and Control Systems, in honor of László Keviczky on his 60th birthday,* Budapest University of Technology and Economics, eds. J. Bokor and K.M. Hangos, pp. 165-180.

Camacho, E. F., C.Bordons (1999, 2004). Model Predictive Control, *Springer-Verlag*, 280 p.

Johnson, M.A., M.H. Moradi (2005). PID Control, Chapter 13, *Springer-Verlag*, 543 p.

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

Kuhn, U. (1995). A practical tuning rule for PID controllers: the T-sum rule (in German). *Automatisierungs-technische Praxis*, Vol. 37, No. 5, pp. 10-16.

Normey-Rico, J.E., E.F. Camacho (2007). Control of Dead-time Processes, *Springer-Verlag*, 462 p.