# Various Approaches to Solving an Industrially Motivated Control Problem: Software Implementation and Simulation

Radek Matušů, and Roman Prokop

***Abstract*—**The main aim of this paper is to present various approaches to solving an industrially motivated control problem, especially from the viewpoint of implementation of control algorithms into the Matlab and Pascal environment. The motivation and basic conditions of the application have been based on real technical assignment of a manufacturer of aluminium-based rolled products and packaging materials. The primary part of the work deals with selected digital self-tuning controllers where the applied methods comprise a polynomial approach to discrete-time control design and recursive least-squares identification algorithm LDDIF. Subsequently, two alternative approaches were analyzed, namely control using continuous-time regulator with fixed parameters and usage of delta approach in self-tuning control.

***Keywords*—**Self-tuning controllers, digital control, polynomial approach, software implementation, continuous-time control, delta models.

## I. Introduction

Real control of industrial processes is almost always burden with various perturbations, disturbances and changes in process parameters or dynamics due to varying operational conditions, plant properties themselves, etc. Furthermore, an acceptable a priori mathematical model does not have to be known. In spite of it, such processes have to be controlled.

A possible solution to this task represents an area of control theory known as adaptive control or more specifically usage of self-tuning controllers [1]-[9]. Some specific issues related to self-tuning control can be found e.g. in [10], [11]. The main idea consists in modification of control law according to the changing plant parameters obtained via recursive identification. Its advantage is some kind of "intelligent" behaviour, but on the other hand these regulators are quite complex and not easily applicable. A possible different approach is represented e.g. by determination of stabilizing controllers [12], [13] and additional verification of control system robustness [14], [15].

This paper deals mainly with software implementation of selected digital self-tuning control algorithms into the Matlab and Pascal environment for the purpose of possible industrial utilization. The work was motivated by co-operation with a manufacturer of aluminium-based rolled products and packaging materials. His project has supposed primarily the application of discrete-time adaptive compensator to control of a metal smelting furnace. Other requirements were the plant model with "a2b3" structure and final implementation in Borland Pascal (because of integration into the existing system). However, the paper presents not only derived relations applicable to Pascal environment but also program for simulative purposes and testing created under Matlab and some preliminary simulation results. The primary digital self-tuning control approach has included a polynomial method to discrete-time control design and recursive least-squares identification algorithm LDDIF. Furthermore, two alternative techniques, namely control using continuous-time regulator with fixed parameters and use of delta approach in self-tuning control, have been studied. Even though all the tasks were motivated by the specific problem, the paper tries to present it in more or less generally applicable way.

The previous versions of this work have been presented at conferences [16]-[18] and as the chapter in the book [19].

The work is organized as follows. In Section II, the main principles of digital self-tuning control as well as the basics of polynomial synthesis are described and the "Pascal-friendly" rules for computation of controller parameters are derived. The Section III then focuses on the recursive least-squares algorithm with exponential and directional forgetting LDDIF. The following Section IV presents implementation of the control and identification algorithms into the Matlab environment and demonstrates its capabilities by means of preliminary simulation example. Further, two alternative approaches including design of fixed continuous-time controller and control synthesis using delta models can be found in the extensive Section V. And finally, Section VI offers some conclusion remarks.

Manuscript received December 3, 2011. This research was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the Research Plan No. MSM 7088352102 and by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089. This assistance is very gratefully acknowledged. Moreover, the authors would like to thank Jaromír Kovář for his intense and enthusiastic co-operation on the project connected with content of this paper.

Radek Matušů is with the Department of Automation and Control Engineering, Faculty of Applied Informatics, Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic. The email contact is: rmatusu@fai.utb.cz.

Roman Prokop is with the Department of Automation and Control Engineering, Faculty of Applied Informatics, Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic. The email contact is: prokop@fai.utb.cz.

## II. Polynomial Synthesis in Discrete-Time Domain

The basic principle of applied self-tuning control scenario consists in consecutive identification of the controlled process using a recursive algorithm (see the following Section III) and application of obtained plant parameters in computing the control law. The control design itself has been based on algebraic approach and pole placement [20]-[23].

In spite of the existence of more complex control configurations, only the very basic single-input single-output (SISO) control loop with one degree of freedom has been assumed. This classical feedback connection in a discrete-time sense is shown in Fig. 1.

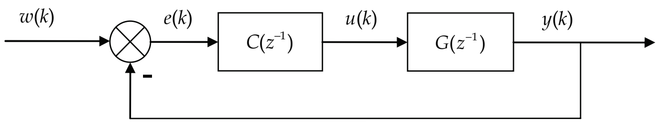


Fig. 1 discrete-time feedback control loop

The signals $w(k)$, $e(k)$, $u(k)$ and $y(k)$ from Fig. 1 represent reference value, tracking (control) error, actuating (manipulated) signal and controlled (output) variable, respectively, and blocks $C(z^{-1})$ and $G(z^{-1})$ mean discrete-time transfer functions of a controller and controlled system.

According to project requirements a controlled plant is supposed to has an "a2b3" structure, i.e. its transfer function is:

$$G(z^{-1}) = \frac{b(z^{-1})}{a(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{1}$$

A suitable controller which ensures stabilization of the whole control loop (Fig. 1) and asymptotic tracking of stepwise reference variable can be obtained by solution of Diophantine equation [20], [21]:

$$a(z^{-1}) f(z^{-1}) p(z^{-1}) + b(z^{-1}) q(z^{-1}) = m(z^{-1}) \tag{2}$$

where $a(z^{-1})$, $b(z^{-1})$ are from the controlled system (1), and $p(z^{-1})$, $q(z^{-1})$ from discrete-time controller:

$$C(z^{-1}) = \frac{q(z^{-1})}{f(z^{-1}) p(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{f(z^{-1})\left(p_0 + p_1 z^{-1} + p_2 z^{-2}\right)} \tag{3}$$

and where $f(z^{-1})$ is the denominator of image of stepwise reference signal:

$$W(z^{-1}) = \frac{h(z^{-1})}{f(z^{-1})} = \frac{h(z^{-1})}{1 - z^{-1}} \tag{4}$$

Moreover, right-hand polynomial $m(z^{-1})$ from (2) is a stable polynomial of appropriate order. Thus the equation (2) takes here the specific form:

$$\begin{aligned}
&\left(1 + a_1 z^{-1} + a_2 z^{-2}\right)\left(1 - z^{-1}\right)\left(p_0 + p_1 z^{-1} + p_2 z^{-2}\right) + \cdots \\
&\cdots \left(b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}\right)\left(q_0 + q_1 z^{-1} + q_2 z^{-2}\right) = \\
&= m_0 + m_1 z^{-1} + m_2 z^{-2} + m_3 z^{-3} + m_4 z^{-4} + m_5 z^{-5}
\end{aligned} \tag{5}$$

The aim is to calculate coefficients of $p(z^{-1})$, $q(z^{-1})$ to get the controller (3). A simple method for finding the particular solution of Diophantine equation (5) grounds in the comparison of coefficients with the same power and consequent transformation of (5) into the set of six equations with six unknowns. This set can be written in a matrix form as follows:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
a_1 - 1 & 1 & 0 & b_1 & 0 & 0 \\
a_2 - a_1 & a_1 - 1 & 1 & b_2 & b_1 & 0 \\
-a_2 & a_2 - a_1 & a_1 - 1 & b_3 & b_2 & b_1 \\
0 & -a_2 & a_2 - a_1 & 0 & b_3 & b_2 \\
0 & 0 & -a_2 & 0 & 0 & b_3
\end{pmatrix}
\begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ q_0 \\ q_1 \\ q_2 \end{pmatrix}
=
\begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \\ m_5 \end{pmatrix} \tag{6}$$

Solving the equation system (6) would be an easy task in many software packages. However, the final implementation of control algorithm in Borland Pascal environment was required by assignment and so the analytical solution of (6) had to be derived in order to be easily programmable. Thus, the utilizable controller parameters are computed according to:

$$\begin{aligned}
q_0 &= x_{13}/x_{12} \\
q_1 &= x_{14}/x_{12} \\
q_2 &= x_{15}/x_{12} \\
p_0 &= m_0 \\
p_1 &= x_2 - b_1 q_0 \\
p_2 &= \left(q_2 b_3 - m_5\right)/a_2
\end{aligned} \tag{7}$$

where auxiliary variables are:

$$\begin{aligned}
x_1 &= \left(a_2 - a_1\right) b_3 + b_2 a_2 \\
x_2 &= m_1 + \left(1 - a_1\right) m_0 \\
x_3 &= \left[a_2 m_4 + a_2^2 x_2 + \left(a_2 - a_1\right) m_5\right]/x_1 \\
x_4 &= \left(a_2^2 b_1\right)/x_1 \\
x_5 &= \left(a_2 b_3\right)/x_1 \\
x_6 &= m_0 \left(a_2 - a_1\right) + x_2 a_1 - x_2 \\
x_7 &= -b_1 a_1 + b_1 + b_2 \\
x_8 &= m_2 - x_6 + m_5/a_2 \\
x_9 &= -b_1 a_2 + b_1 a_1 + b_3 \\
x_{10} &= \left(b_3 a_1\right)/a_2 - b_3/a_2 + b_1 \\
x_{11} &= m_3 + m_0 a_2 - h_2 a_2 + h_2 a_1 + \left(m_5 a_1\right)/a_2 - m_5/a_2 \\
x_{12} &= x_4 b_1 x_{10} + x_7 b_2 + x_9 x_5 \left(b_3/a_2\right) - b_1 x_9 - \left(b_3/a_2\right) b_2 x_4 - \cdots \\
&\quad \cdots x_{10} x_5 x_7
\end{aligned} \tag{8}$$

$$x_{13} = x_3 b_1 x_{10} + x_8 b_2 + x_{11} x_5 \left(b_3/a_2\right) - b_1 x_{11} - \left(b_3/a_2\right) b_2 x_3 - \cdots$$
$$\cdots x_{10} x_5 x_8$$
$$x_{14} = x_4 x_8 x_{10} + x_7 x_{11} + x_9 x_3 \left(b_3/a_2\right) - x_8 x_9 - \left(b_3/a_2\right) x_{11} x_4 - \cdots$$
$$\cdots x_{10} x_3 x_7$$
$$x_{15} = x_4 b_1 x_{11} + x_7 b_2 x_3 + x_9 x_5 x_8 - x_3 b_1 x_9 - x_8 b_2 x_4 - x_{11} x_5 x_7$$

The coefficients of $m(z^{-1})$ can be used for controller tuning and thus for influencing the closed-loop control behaviour. The suitable choice of the roots of the closed-loop characteristic polynomial $m(z^{-1})$ is known as pole placement problem. Anyway, this case of fifth order $m(z^{-1})$ can be easily "degraded" to the lower order ones by equalling the appropriate coefficients to zero. The special events are represented by dead-beat control for $m(z^{-1})=1$ or by linear quadratic (LQ) control for $m(z^{-1})$ given by means of minimizing the LQ criterion [8], [9], [24].

Finally, the calculated parameters (7) are applied to programmable control law which corresponds to the controller (3) and which generates the control signal $u(k)$. It can be formulated as:

$$u(k) = \left[\begin{array}{l}(p_0 - p_1)u(k-1) + (p_1 - p_2)u(k-2) + \cdots \\ \cdots p_2 u(k-3) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)\end{array}\right] \Big/ p_0 \qquad (9)$$

Interested reader can find more information on algebraic methods and their application in analysis and synthesis of control systems e.g. in [20], [21], [25]-[27].

## III. Recursive Identification Algorithm

A LDDIF routine has been used as plant parameters identification technique for combination with algebraic synthesis from the previous Section in order to obtain self-tuning controller. It is recursive least-squares algorithm with exponential and directional forgetting [28]. Moreover, the corrections influencing the covariance matrix $\boldsymbol{P}(k)$ of the estimated parameters by adding some multiple of identity matrix, which have been suggested in [29], are implemented to improve the tracking performance. The algorithm can be described by equations [30]:

$$\begin{aligned}
&\varepsilon(k) = \boldsymbol{y}(k) - \boldsymbol{\Phi}(k)^T \boldsymbol{\theta}(k-1) \\
&r(k) = \boldsymbol{\Phi}(k)^T \boldsymbol{P}(k-1)\boldsymbol{\Phi}(k) \\
&\boldsymbol{\kappa}(k) = \frac{\boldsymbol{P}(k-1)\boldsymbol{\Phi}(k)}{1 + r(k)} \\
&\beta(k) = \begin{cases} \varphi - \dfrac{1-\varphi}{r(k)} & \text{if } r(k) > 0 \\ 1 & \text{if } r(k) > 0 \end{cases} \\
&\boldsymbol{P}(k) = \boldsymbol{P}(k-1) - \frac{\boldsymbol{P}(k-1)\boldsymbol{\Phi}(k)\boldsymbol{\Phi}(k)^T \boldsymbol{P}(k-1)}{\beta(k)^{-1} + r(k)} + \delta \boldsymbol{I} \\
&\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k-1) + \boldsymbol{\kappa}(k)\varepsilon(k)
\end{aligned} \qquad (10)$$

where:

$$\begin{aligned}\boldsymbol{\Phi}(k) = [-y(k-1) \quad -y(k-2) \quad \cdots \\ \cdots \quad u(k-1) \quad u(k-2) \quad u(k-3)]\end{aligned} \qquad (11)$$

is observation vector and:

$$\theta(k) = [a_1(k) \quad a_2(k) \quad b_1(k) \quad b_2(k) \quad b_3(k)] \qquad (12)$$

is vector of parameters. The term $\varphi$ then represents exponential forgetting factor. The initial values for the algorithm are usually preset to $\varphi = 0.985$, $\boldsymbol{P}(0) = 10^6 \boldsymbol{I}$ and $\delta = 0.01$.

The main complication from the implementation viewpoint has been the arduousness in working with matrices.

## IV. Software Implementation

As it was outlined before, Borland Pascal had to be supposed for final application under real industrial conditions because of easy implementation into the existing system. However, several preliminary tests, algorithm verifications and simulations were done in Matlab environment due to better convenience for these testing purposes. As a result, a simple program has been created. Its main window is shown in Fig. 2. The Matlab represents very popular and effective programming and simulation environment for many various theoretical and application disciplines as demonstrated e.g. in [31].

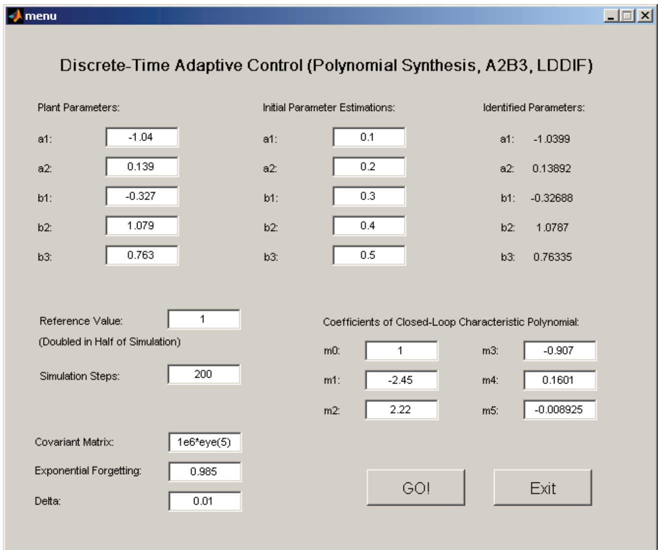

Fig. 2 main window of the preliminary simulation program in Matlab

Initial supportive control and identification experiments for sampling time $T = 45\,\text{s}$ have led to parameters of controlled system (1):

$$\begin{aligned} a_1 &= -1.04 \\ a_2 &= 0.139 \\ b_1 &= -0.327 \\ b_2 &= 1.079 \\ b_3 &= 0.763 \end{aligned} \tag{13}$$

The closed-loop characteristic polynomial has been supposed as:

$$\begin{aligned} m(z^{-1}) &= 1 - 2.45z^{-1} + 2.22z^{-2} - 0.907z^{-3} + 0.1601z^{-4} - \cdots \\ &\cdots 0.008925z^{-5} \end{aligned} \tag{14}$$

which means that the poles of the closed loop transfer function (Fig. 1) have been placed to:

$$\begin{aligned} r_1 &= 0.85 \\ r_2 &= 0.7 \\ r_3 &= 0.5 \\ r_4 &= 0.3 \\ r_5 &= 0.1 \end{aligned} \tag{15}$$

Simulation result of control behaviour is depicted in Fig. 3. The huge overshoot in the beginning of the control process is caused by incomplete identification stage. The parameters of the controlled system were assumed to be unknown and preset to random starting values (as demonstrated in window from Fig. 2). The progress in identification of these parameters during control is shown in Fig. 4 with zoomed x-axis. As can be clearly seen, the plant parameters were properly identified after several initial steps and thanks to this the control response from Fig. 3 is much better at the middle step change of reference signal.

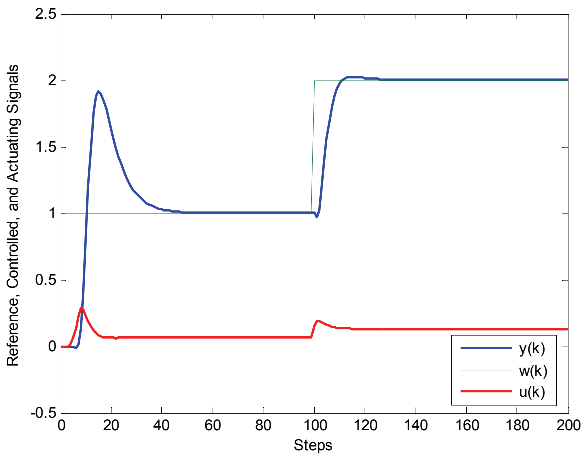


Fig. 3 control of plant using discrete-time self-tuning controller – simulation

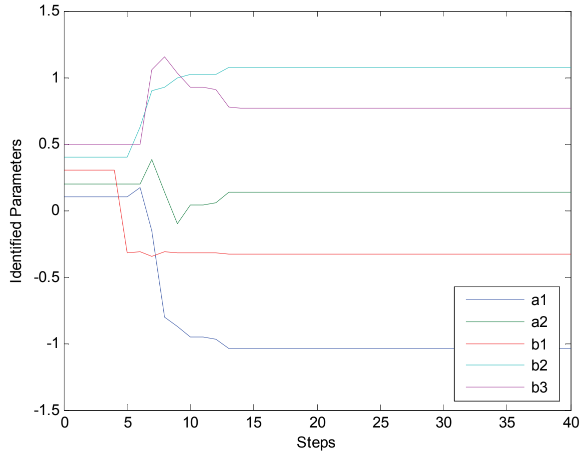


Fig. 4 development of the identified parameters

## V. Alternative Approaches

In spite of the fact that preliminary control simulations from the previous Section IV have brought satisfactory results, two technically different approaches have been studied, i.e. control using continuous-time regulator with fixed parameters and application of delta approach in self-tuning control scenario. The main motivation for these investigations consisted in use of shorter sampling time.

### A. *Fixed Continuous-Time Controller*

First alternative technique to control synthesis has been based on the similar algebraic tools as described in Section II [21], but now in continuous-time representation. Moreover, one off-line controller with fixed parameters has been tuned and due to this fact no recursive identification for adaptation reasons was needed anymore.

Primarily, the discrete-time model (1) with identified parameters (13) was transformed into continuous-time model suitable for linear Diophantine equations (which means without time-delay term). This has been performed very simply by using the first order Taylor approximation of time-delay term in denominator:

$$\begin{aligned} G(z^{-1}) &= \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-1} \Rightarrow \\ &\Rightarrow \frac{-0.327s^2 - 0.0289s + 0.001746}{s^2 + 0.04385s + 0.0001141} e^{-45s} \approx \\ &\approx \frac{-0.007267s^2 - 0.0006423s + 0.0000388}{s^3 + 0.06607s^2 + 0.001089s + 0.000002535} = G(s) \end{aligned} \tag{16}$$

Correspondence of $G(z^{-1})$ and $G(s)$ is demonstrated in Fig. 5 where step responses of both models are compared.

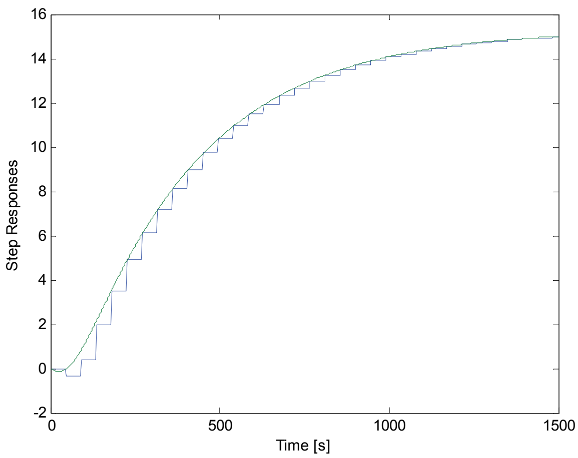


Fig. 5 comparison of step responses of discrete-time and continuous-time models

The control design itself starts from continuous-time version of Diophantine equation (2):

$$a(s)f(s)p(s)+b(s)q(s)=m(s) \tag{17}$$

where analogically to Section II:

$$\begin{aligned} a(s) &= s^3+0.06607s^2+0.001089s+0.000002535 \\ b(s) &= -0.007267s^2-0.0006423s+0.0000388 \\ f(s) &= s \end{aligned} \tag{18}$$

and where closed-loop characteristic polynomial has been assumed:

$$\begin{aligned} m(s) &= s^6+0.133s^5+0.006765s^4+0.0001689s^3+\cdots \\ &\cdots 2.14925\cdot10^{-6}s^2+1.26\cdot10^{-8}s+2.25\cdot10^{-11} \end{aligned} \tag{19}$$

i.e. its roots are:

$$\begin{aligned} r_1 &= -0.003 \\ r_2 &= -0.05 \\ r_3 &= -0.02 \\ r_4 &= -0.02 \\ r_5 &= -0.025 \\ r_6 &= -0.015 \end{aligned} \tag{20}$$

The final continuous-time controller has been calculated as:

$$\begin{aligned} C(s) &= \frac{q(s)}{f(s)p(s)} = \\ &= \frac{0.2591s^3+0.01549s^2+0.0002423s+5.799\cdot10^{-7}}{s^3+0.06881s^2+0.001409s} \end{aligned} \tag{21}$$

Supposing the derivative approximation (e.g. here for tracking error $e$):

$$\frac{de(t)}{dt} \approx \frac{e(k)-e(k-1)}{T} \tag{22}$$

leads to "emulation" of continuous-time (21) suitable for Borland Pascal environment. Thus, the control law can be accomplished by relation:

$$u(k)=\frac{\left[\begin{array}{l} q_3\left(e(k)-3e(k-1)+3e(k-2)-e(k-3)\right)+\cdots \\ \cdots Tq_2\left(e(k)-2e(k-1)+e(k-2)\right)+\cdots \\ \cdots T^2q_1\left(e(k)-e(k-1)\right)+T^3q_0e(k)+\cdots \\ \cdots 3u(k-1)-3u(k-2)+u(k-3)-\cdots \\ \cdots Tp_2\left(-2u(k-1)+u(k-2)\right)+T^2p_1u(k-1) \end{array}\right]}{\left(1+Tp_2+T^2p_1\right)} \tag{23}$$

Symbol $T$ in (22) and (23) represents sampling time, usually very short one, because the shorter sampling period means the closer approximation of continuous-time controller (21) by the equation (23). From the practical point of view, the sampling time must be adjusted according to available hardware possibilities.

Results of control simulation are visualized in Fig. 6.

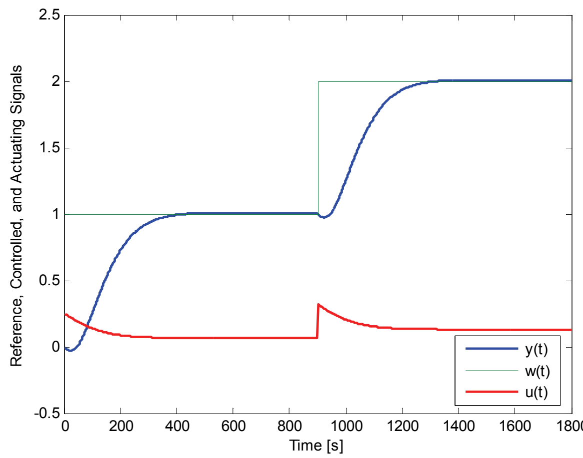


Fig. 6 control of plant using continuous-time controller – simulation

### *B. Control Design Using Delta Models*

Another alternative to avoid potential problems with long sampling periods consists in usage of delta models. They act as a bridge between discrete-time and continuous-time representations and eliminate objectionable numerical properties of discrete-time models under short sampling times. Originally, the delta operator has been defined in [32]:

$$\delta = \frac{z-1}{T} \tag{24}$$

Consequent generalization of such models with complex variable $\gamma$ has been published in [33]. It has proved that all operators:

$$\gamma = \frac{z-1}{\lambda T z + (1-\lambda)T}; \quad 0 \le \lambda \le 1 \tag{25}$$

converge to derivation. Three most common cases are for $\lambda = 0$ (forward model):

$$\gamma = \frac{z-1}{T} \tag{26}$$

$\lambda = 1$ (backward model):

$$\gamma = \frac{1-z^{-1}}{T} \tag{27}$$

and $\lambda = 0.5$ (Tustin approximation):

$$\gamma = \frac{2}{T}\frac{z-1}{z+1} \tag{28}$$

The PID-B2 controller [8], [9], [34] has been utilized in this method. It is based on structure developed in [35] which is shown in Fig. 7.

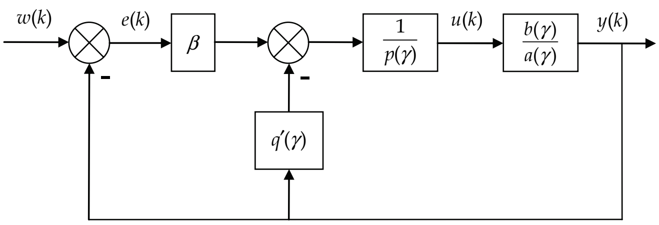


Fig. 7 closed loop with PID-B controller

The controlled plant from Fig. 7 is supposed as:

$$\frac{b(\gamma)}{a(\gamma)} = \frac{b_1\gamma + b_0}{\gamma^2 + a_1\gamma + a_0} \tag{29}$$

and controller polynomials are:

$$\begin{aligned} p(\gamma) &= \gamma(\gamma + \lambda) \\ q'(\gamma) &= \gamma(q_2'\gamma + q_1') \end{aligned} \tag{30}$$

Generally, closed-loop characteristic polynomial of the connection in Fig. 7 is:

$$a(\gamma)p(\gamma) + b(\gamma)\left[q'(\gamma) + \beta\right] = m(\gamma) \tag{31}$$

And more specifically, it is assumed to have the form:

$$m(\gamma) = (\gamma - \alpha)^2\left[\gamma - (\alpha + j\omega)\right]\left[\gamma - (\alpha - j\omega)\right] \tag{32}$$

The parameter $\alpha$ can serve for changing speed of control process and "aggressiveness" of actuating signal, while $\omega$ is useful for selecting size of overshoot.

However, for the sake of control loop stability, roots of the polynomial (32) must always lie inside the circle with centre in $-1/T$ and the same radius (the circle goes through the origin of the complex plane).

Thus, adjusted characteristic polynomial can be written as [34]:

$$\begin{aligned} &\gamma^4 + \gamma^3\left[a_1 + \lambda + b_1(q_2' + \beta)\right] + \cdots \\ &\cdots \gamma^2\left[a_0 + a_1\lambda + b_1\left(q_1' + \frac{2\beta}{T}\right) + b_0(q_2' + \beta)\right] + \cdots \\ &\cdots \gamma\left[a_0\lambda + \frac{\beta}{T^2}b_1 + b_0\left(q_1' + \frac{2\beta}{T}\right)\right] + \frac{\beta}{T^2}b_0 = \\ &= \gamma^4 - 4\gamma^3\alpha + \gamma^2\left(6\alpha^2 + \omega^2\right) + \cdots \\ &\cdots \gamma\left[-2\alpha\left(2\alpha^2 + \omega^2\right)\right] + \alpha^2\left(\alpha^2 + \omega^2\right) \end{aligned} \tag{33}$$

More convenient matrix form is:

$$\begin{pmatrix} b_1 & 0 & b_1 & 1 \\ b_0 & b_1 & b_0 + \frac{2b_1}{T} & a_1 \\ 0 & b_0 & \frac{2b_0}{T} + \frac{b_1}{T^2} & a_0 \\ 0 & 0 & \frac{b_0}{T^2} & 0 \end{pmatrix} \begin{pmatrix} q_2' \\ q_1' \\ \beta \\ \lambda \end{pmatrix} = \begin{pmatrix} -4\alpha - a_1 \\ 6\alpha^2 + \omega^2 - a_0 \\ -2\alpha\left(2\alpha^2 + \omega^2\right) \\ \alpha^2\left(\alpha^2 + \omega^2\right) \end{pmatrix} \tag{34}$$

Analytical solution suitable for Pascal implementation can look like:

$$\begin{aligned} \beta &= \frac{x_{r4}}{x_{l3}} \\ q_1' &= \frac{a_0 x_{r5} - (b_1a_1 - b_0)x_{r6}}{a_0 b_1^2 - (b_1 a_1 - b_0)b_0} \\ \lambda &= \frac{x_{r6} - b_0 q_1}{a_0} \\ q_2' &= \frac{x_{r1} - b_1\beta - \lambda}{b_1} \end{aligned} \tag{35}$$

where auxiliary variables are:

$$\begin{aligned} x_{l1} &= b_0 + \frac{2b_1}{T} \\ x_{l2} &= \frac{2b_0}{T} + \frac{b_1}{T^2} \\ x_{l3} &= \frac{b_0}{T^2} \\ x_{r1} &= -4\alpha - a_1 \\ x_{r2} &= 6\alpha^2 + \omega^2 - a_0 \\ x_{r3} &= -2\alpha\left(2\alpha^2 + \omega^2\right) \\ x_{r4} &= \alpha^2\left(\alpha^2 + \omega^2\right) \end{aligned} \tag{36}$$

and

$$\begin{aligned} x_{r5} &= -b_0 x_{r1} + b_1 x_{r2} + b_0 b_1 \beta - b_1 x_{l1} \beta \\ x_{r6} &= x_{r3} - x_{l2} \beta \end{aligned} \tag{37}$$

The final control law is then generated by:

$$\begin{aligned} &u(k) = \beta e(k) - q_2' \left[ y(k) - 2y(k-1) + y(k-2) \right] - \cdots \\ &\cdots q_1' T \left[ y(k-1) - y(k-2) \right] - \lambda T \left[ u(k-1) - u(k-2) \right] + \cdots \\ &\cdots 2u(k-1) - u(k-2) \end{aligned} \tag{38}$$

while the vector of parameters:

$$\theta(k) = \left[ a_1(k) \quad a_0(k) \quad b_1(k) \quad b_0(k) \right] \tag{39}$$

is identified using the same recursive algorithm as described in Section III. The only modifications necessary because of delta representation are that measured output $y(k)$ is replaced by the ratio:

$$\left[ y(k) - 2y(k-1) + y(k-2) \right] / T^2 \tag{40}$$

and that the observation vector has the form:

$$\begin{aligned} &\boldsymbol{\Phi}(k) = \left[ -\left( y(k-1) - y(k-2) \right) / T \quad -y(k-2) \quad \cdots \right. \\ &\left. \cdots \quad \left( u(k-1) - u(k-2) \right) / T \quad u(k-2) \right] \end{aligned} \tag{41}$$

## VI. Conclusion

This paper has been focused mainly on preliminary software implementation of digital self-tuning controllers into the Matlab (for simulative and testing purposes) and Pascal (for real application) environment. The motivation to this task as well as basic conditions and restrictions have been based on technical assignment of a manufacturer of aluminium-based products related to control of a metal smelting furnace. In the first instance, the applied techniques have comprised a polynomial approach to discrete-time control design and recursive least-squares identification algorithm LDDIF. On top of that, continuous-time controller with fixed parameters and delta approach in self-tuning control scenario have been studied.

## References


[1] K. J. Åström, and B. Wittenmark, "On self-tuning regulators" *Automatica*, vol. 9, no. 2, pp. 185-199, 1973.

[2] K. J. Åström, U. Borisson, L. Ljung, and B. Wittenmark, "Theory and applications of self-tuning regulators" *Automatica*, vol. 13, no. 5, pp. 457-476, 1977.

[3] D. W. Clarke, and P. J. Gawthrop, "Self-tuning control" *IEE Proceedings*, vol. 126, no. 6, pp. 633-640, 1979.

[4] K. J. Åström, and B. Wittenmark, *Adaptive Control*. Reading, Massachusetts, USA: Addison-Wesley, 1989.

[5] P. E. Wellstead, and M. B. Zarrop, *Self-Tuning Systems*. New York, USA: Wiley, 1991.

[6] R. Isermann, K.-H. Lachmann, and D. Matko, *Adaptive Control Systems*. Englewood Cliffs, New Jersey, USA: Prentice Hall, 1992.

[7] C. C. Hang, T. H. Lee, and W. K. Ho, *Adaptive Control*. North Carolina, USA: Instrument Society of America, 1993.

[8] V. Bobál, J. Böhm, R. Prokop, and J. Fessl, *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace* (*Practical Aspects of Self-Tuning Controllers: Algorithms and Implementation*). Brno, Czech Republic: Brno University of Technology, VUTIUM, 1999. (In Czech).

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

[10] V. Bobál, P. Chalupa, P. Dostál, and J. Novák, "Disturbance Rejection of Nonlinear Servo System by Self-tuning Control" *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 5, no. 8, pp. 1431-1438, 2011.

[11] V. Bobál, P. Chalupa, P. Dostál, and M. Kubalčík, "Design and Simulation Verification of Self-tuning Smith Predictors" *International Journal of Mathematics and Computers in Simulation*, vol. 5, no. 4, pp. 342-351, 2011.

[12] R. Matušů, "Calculation of all stabilizing PI and PID controllers" *International Journal of Mathematics and Computers in Simulation*, vol. 5, no. 3, pp. 224-231, 2011.

[13] L. Pekař, R. Prokop, "Stabilization of a Delayed System by a Proportional Controller" *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 4, no. 4, pp. 282-290, 2010.

[14] T. Emami, and J. M. Watkins, "Robust Performance Characterization of PID Controllers in the Frequency Domain" *WSEAS Transactions on Systems and Control*, vol. 4, no. 5, pp. 232-242, 2009.

[15] R. Matušů, R. Prokop, and L. Pekař, "Parametric and unstructured approach to uncertainty modelling and robust stability analysis" *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 5, no. 6, pp. 1011-1018, 2011.

[16] R. Matušů, and R. Prokop, "Software Implementation of Discrete-Time Adaptive Control" presented at the 13th International Research/Expert Conference "Trends in the Development of Machinery and Associated Technology", Hammamet, Tunisia, 2009.

[17] R. Matušů, R. Prokop, and L. Pekař, "Polynomial Approach to Discrete-Time Adaptive Control: Software Implementation for Industrial Application" presented at the 20th International DAAAM Symposium, Vienna, Austria, 2009.

[18] R. Matušů, and R. Prokop, "Polynomial Approach to Digital Self-Tuning Control: Industrially Motivated Application" presented at the 13th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering, Catania, Italy, 2011.

[19] R. Matušů, and R. Prokop, "Software Implementation of Self-Tuning Controllers" in *New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems*, Meng Joo Er, Ed. Rijeka, Croatia: Sciyo, 2010, pp. 39-52.

[20] V. Kučera, *Discrete Linear Control: The Polynomial Equation Approach*. Chichester, UK: John Wiley, 1979.

[21] V. Kučera, "Diophantine Equations in Control – A Survey" *Automatica*, vol. 29, no. 6, pp. 1361-1375, 1993.

[22] P. E. Wellstead, D. I. Prager, and P. M. Zanker, "A pole assignment self-tuning regulator" *IEE Proceedings*, vol. 126, no. 8, pp. 781-787, 1979.

[23] K. J. Åström, and B. Wittenmark, "Self-tuning controller based on pole-zero placement" *IEE Proceedings*, vol. 127, no. 3, pp. 120-130, 1980.

[24] K. J. Hunt, M. Šebek, and V. Kučera, "$H_2$ Control Problems" in *Polynomial Methods in Optimal Control and Filtering*, K. J. Hunt, Ed. Stevenage, United Kingdom: Peter Peregrinus Ltd., 1993, pp. 29-55.

[25] M. Vidyasagar, *Control System Synthesis: A Factorization Approach*. Cambridge, Massachusetts, USA: MIT Press, 1985.

[26] V. Kučera, *Analysis and Design of Discrete Linear System*. Englewood Cliffs, New Jersey, USA: Prentice Hall, 1991.

[27] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. New York, USA: Macmillan, 1992.

[28] R. Kulhavý, and M. Kárný, "Tracking of Slowly Varying Parameters by Directional Forgetting" presented at the 9th IFAC World Congress, Budapest, Hungary, 1984.

[29] S. Bittanti, P. Bolzern, and M. Campi, "Exponential Convergence of a Modified Directional Forgetting Identification Algorithm" *Systems & Control Letters*, vol. 14, no. 2, pp. 131-137, 1990.

[30] Ľ. Čirka, M. Fikar, and P. Petruš, "IDTOOL 4.0 – A Dynamical System Identification Toolbox for Matlab/Simulink" presented at the Technical Computing Prague 2006, Prague, Czech Republic, 2006.

[31] J. Vojtěšek, and P. Dostál, "Use of MATLAB Environment for Simulation and Control of CSTR" *International Journal of Mathematics and Computers in Simulation*, vol. 5, no. 6, pp. 528-535, 2011.
[32] R. H. Middleton, and G. C. Goodwin, *Digital Control and Estimation: A Unified Approach*. Englewood Cliffs, New Jersey, USA: Prentice Hall, 1989
[33] S. Mukhopadhyay, A. Patra, and G. P. Rao, "New class of discrete-time models for continuous-time systems" *International Journal of Control*, vol. 55, no. 5, pp. 1161-1187, 1992.
[34] M. Sysel, *Application of delta models to process control* (*Využití delta modelů pro řízení procesů*). Ph.D. Thesis, Zlín, Czech Republic: Faculty of Technology, Tomas Bata University in Zlín, 2001. (In Czech).
[35] R. Ortega, and R. Kelly, "PID self-tuners: Some theoretical and practical aspects" *IEEE Transactions on Industrial Electronics*, vol. 31, no. 4, pp. 332-338, 1984.

**Radek Matušů** was born in Zlín, Czech Republic in 1978. He is a Researcher at Faculty of Applied Informatics of Tomas Bata University in Zlín, Czech Republic. He graduated from Faculty of Technology of the same university with an MSc in Automation and Control Engineering in 2002 and he received a PhD in Technical Cybernetics from Faculty of Applied Informatics in 2007. He worked as a Lecturer from 2004 to 2006. The main fields of his professional interest include robust systems and application of algebraic methods to control design.

**Roman Prokop**, born in 1952, is a Vice-Dean and a Full Professor at Faculty of Applied Informatics of Tomas Bata University in Zlín, Czech Republic. He graduated from Czech Technical University in Prague in 1976 and received a PhD from Slovak University of Technology in Bratislava in 1983. He was an Associate Professor from 1996 and a Full Professor from 2004. He aims his pedagogical and research work to automatic control theory, algebraic methods in control design and optimization. The main interests of the latest period are uncertain and robust systems, autotuning of controllers and time-delay systems.