3

Software Implementation of Self-Tuning Controllers

Radek Matušů and Roman Prokop
Department of Automation and Control Engineering
Faculty of Applied Informatics
Tomas Bata University in Zlín
Nad Stráněmi 4511, Zlín 76005
Czech Republic

1. 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 (Åström & Wittenmark, 1973); (Åström et al., 1977); (Clarke & Gawthrop, 1979); (Åström & Wittenmark, 1989); (Wellstead & Zarrop, 1991); (Isermann et al. 1992); (Hang et al., 1993); (Bobál et al., 1999); (Bobál et al., 2005). Main idea consists in modification of control law according to the changing plant parameters obtained via recursive identification. Their advantage is some kind of "intelligent" behaviour, but on the other hand these regulators are quite complex and not easily applicable.

This chapter 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. In the first stage, the applied methods have included a polynomial approach to discrete-time control design and recursive least-squares identification algorithm LDDIF, but subsequently also two alternative techniques, namely control using continuous-time regulator with fixed parameters and use of delta approach in self-tuning control, have been verified. Although all the tasks were motivated by our

specific problem, the whole chapter tries to present them in more or less generally applicable way.
The chapter is organized as follows. In Section 2, basic theoretical background of digital self-tuning controllers using polynomial synthesis and applicable relations are provided. The Section 3 then contains the fundamentals of related recursive identification algorithm LDDIF. Next, the Section 4 describes the preliminary software implementation and demonstrates its facilities on an application-oriented example. Furthermore, design and utilization of fixed continuous-time controller along with self-tuning control using delta models as two various alternative approaches are presented in an extensive Section 5. And finally, Section 6 offers some conclusion remarks.
The preliminary version of this work has been presented at conferences (Matušů & Prokop, 2009), (Matušů et al., 2009).

2. Discrete-time polynomial synthesis

In the first instance the digital self tuning controllers were intended to be implemented. Their very basic principle consists in consecutive identification of the controlled process using a recursive algorithm (see the following Section) and application of obtained plant parameters in computing the control law. The control design itself has been based on algebraic approach and pole placement (Kučera, 1979); (Kučera, 1993); (Wellstead et al., 1979); (Åström & Wittenmark, 1980).
Despite the existence of more complex control configurations, just 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.

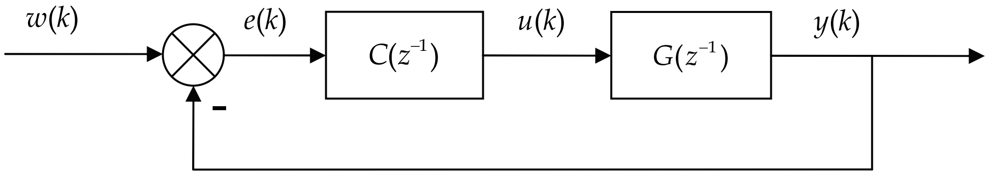

Fig. 1. Basic closed control loop

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 circuit (Fig. 1) and asymptotic tracking of stepwise reference variable can be obtained by solution of Diophantine equation (Kučera, 1979); (Kučera, 1993):

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

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

$$C(z^{-1})=\frac{q(z^{-1})}{f(z^{-1})p(z^{-1})}=\frac{q_0+q_1z^{-1}+q_2z^{-2}}{f(z^{-1})\left(p_0+p_1z^{-1}+p_2z^{-2}\right)} \tag{3}$$

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:

$$\begin{aligned}&\left(1+a_1z^{-1}+a_2z^{-2}\right)\left(1-z^{-1}\right)\left(p_0+p_1z^{-1}+p_2z^{-2}\right)+\left(b_1z^{-1}+b_2z^{-2}+b_3z^{-3}\right)\left(q_0+q_1z^{-1}+q_2z^{-2}\right)=\\&=m_0+m_1z^{-1}+m_2z^{-2}+m_3z^{-3}+m_4z^{-4}+m_5z^{-5}\end{aligned} \tag{5}$$

Obviously, 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:

$$\begin{aligned}q_0&=x_{13}/x_{12}\\q_1&=x_{14}/x_{12}\\q_2&=x_{15}/x_{12}\\p_0&=m_0\\p_1&=x_2-b_1q_0\\p_2&=\left(q_2b_3-m_5\right)/a_2\end{aligned} \tag{7}$$

where auxiliary variables are:

$$
\begin{aligned}
x_1 &= (a_2 - a_1)b_3 + b_2 a_2 \\
x_2 &= m_1 + (1 - a_1)m_0 \\
x_3 &= \left[a_2 m_4 + a_2^2 x_2 + (a_2 - a_1)m_5\right] / x_1 \\
x_4 &= \left(a_2^2 b_1\right) / x_1 \\
x_5 &= (a_2 b_3) / x_1 \\
x_6 &= m_0(a_2 - a_1) + x_2 a_1 - x_2 \\
x_7 &= -b_1 a_1 + b_1 + b_2 \\
x_8 &= m_2 - x_6 + m_5 / a_2 \\
x_9 &= -b_1 a_2 + b_1 a_1 + b_3 \\
x_{10} &= (b_3 a_1) / a_2 - b_3 / a_2 + b_1 \\
x_{11} &= m_3 + m_0 a_2 - h_2 a_2 + h_2 a_1 + (m_5 a_1) / a_2 - m_5 / a_2 \\
x_{12} &= x_4 b_1 x_{10} + x_7 b_2 + x_9 x_5 (b_3 / a_2) - b_1 x_9 - (b_3 / a_2) b_2 x_4 - x_{10} x_5 x_7 \\
x_{13} &= x_3 b_1 x_{10} + x_8 b_2 + x_{11} x_5 (b_3 / a_2) - b_1 x_{11} - (b_3 / a_2) b_2 x_3 - x_{10} x_5 x_8 \\
x_{14} &= x_4 x_8 x_{10} + x_7 x_{11} + x_9 x_3 (b_3 / a_2) - x_8 x_9 - (b_3 / a_2) x_{11} x_4 - x_{10} x_3 x_7 \\
x_{15} &= x_4 b_1 x_{11} + x_7 b_2 x_3 + x_9 x_5 x_8 - x_3 b_1 x_9 - x_8 b_2 x_4 - x_{11} x_5 x_7
\end{aligned}
\tag{8}
$$

The coefficients of the polynomial $m(z^{-1})$ 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. Moreover, the special events are represented by so-called 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 (Hunt et al., 1993); (Bobál et al., 1999).

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:

$$u(k) = \left[(p_0 - p_1)u(k-1) + (p_1 - p_2)u(k-2) + p_2 u(k-3) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)\right] / p_0 \tag{9}$$

Interested reader can find more information on algebraic methods and their application in analysis and synthesis of control systems e.g. in (Kučera, 1979); (Kučera, 1993); (Vidyasagar, 1985); (Doyle et al., 1992).

3. 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 (Kulhavý & Kárný, 1984). 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 (Bittanti et al., 1990), are implemented to improve the tracking performance.
The algorithm can be described by equations (Čirka et al., 2006):

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

where $\boldsymbol{\Phi}(k) = \begin{bmatrix} -y(k-1) & -y(k-2) & u(k-1) & u(k-2) & u(k-3) \end{bmatrix}$ is observation vector, $\theta(k) = \begin{bmatrix} a_1(k) & a_2(k) & b_1(k) & b_2(k) & b_3(k) \end{bmatrix}$ is vector of parameters and φ is exponential forgetting. 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 point of view has been the arduousness in working with matrices.

4. Software implementation of self-tuning controllers

As it was adumbrated before, Borland Pascal had to be assumed for final application in 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.
Initial real control and identification experiments for sampling time T = 45s have led to parameters of controlled system (1):

$$\begin{aligned}
a_1 &= -1.04 \\
a_2 &= 0.139 \\
b_1 &= -0.327 \\
b_2 &= 1.079 \\
b_3 &= 0.763
\end{aligned} \tag{11}$$

The closed-loop characteristic polynomial has been supposed as:

$$m(z^{-1}) = 1 - 2.45z^{-1} + 2.22z^{-2} - 0.907z^{-3} + 0.1601z^{-4} - 0.008925z^{-5} \tag{12}$$

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

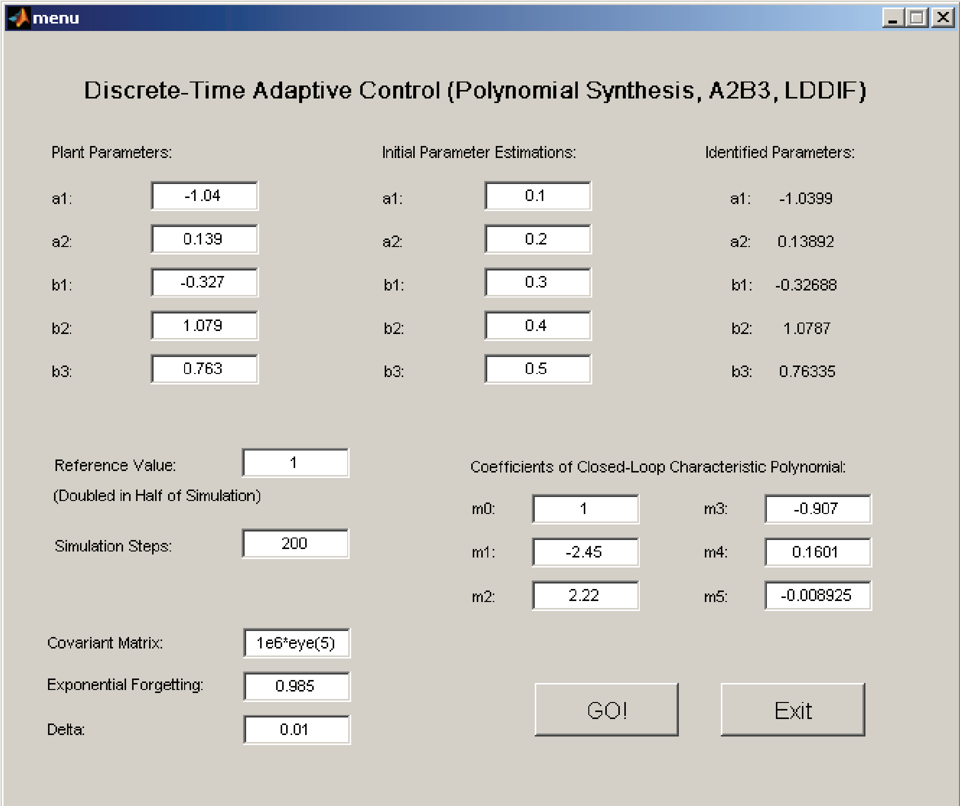

Fig. 2. Main window of the preliminary simulation program in Matlab

$$
\begin{aligned}
r_1 &= 0.85 \\
r_2 &= 0.7 \\
r_3 &= 0.5 \\
r_4 &= 0.3 \\
r_5 &= 0.1
\end{aligned}
\qquad (13)
$$

Simulation result of control behaviour is depicted in Fig. 3. The huge overshoot in the beginning of the process is caused by not completed identification stage. The parameters of the controlled system were assumed to be unknown and preset to random starting values (as demonstrated in Fig. 2). The progress in identification of these parameters during control is shown in Fig. 4 with zoomed x-axis. As can be seen the plant parameters was properly identified after several initial steps and thanks to this the control response from Fig. 3 is much better at middle step change of reference signal.

However, one must emphasize that horizontal axis from Fig. 3 represents steps while each step takes 45 seconds. Unfortunately, such sampling time turned out not to be admissible to the submitter of the task because of psychological aspects for staff during operational changes. Thus, also other approaches without long-term sampling periods have been verified.

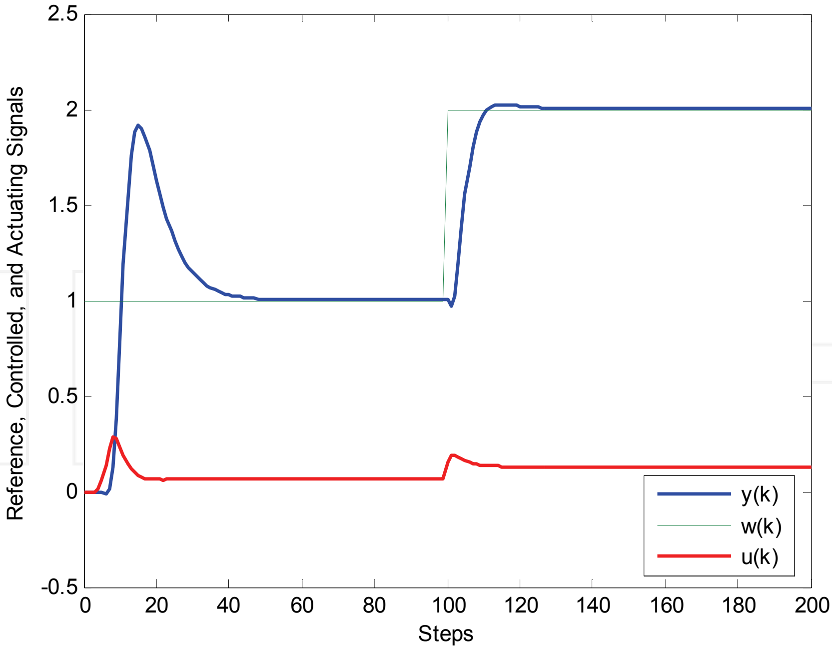

Fig. 3. Control of plant using discrete-time self-tuning controller – simulation

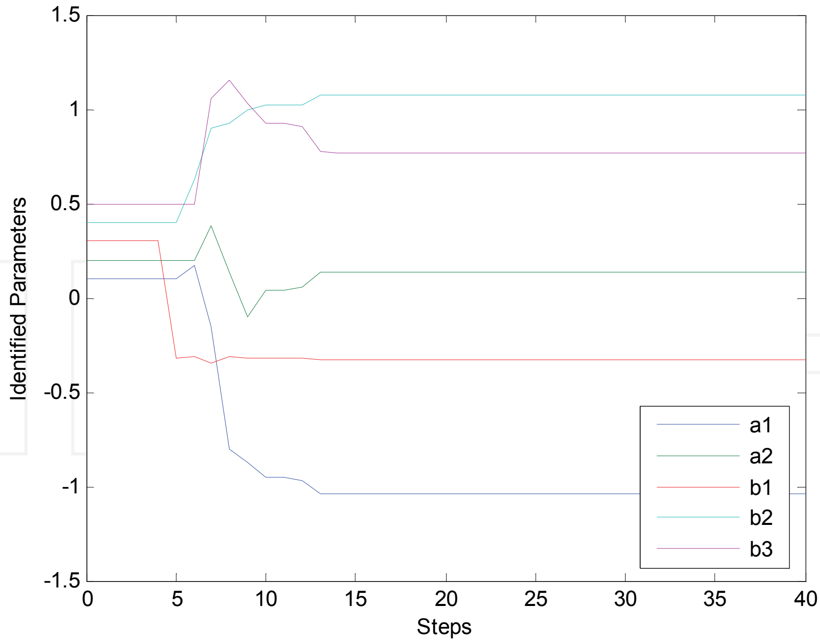

Fig. 4. Development of the identified parameters

5. Alternative approaches

5.1 Fixed continuous-time controller

First alternative technique to control synthesis has been based on the similar algebraic tools as described in Section 2 (Kučera, 1993), 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 (11) had to be transformed into continuous-time model suitable for linear Diophantine equations using first order Taylor approximation of time-delay term in denominator:

$$\begin{aligned} G(z^{-1}) &= \frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-1} \Rightarrow \frac{-0.327s^2 - 0.0289s + 0.001746}{s^2 + 0.04385s + 0.0001141} e^{-45s} \approx \\ &\approx \frac{-0.007267s^2 - 0.0006423s + 0.0000388}{s^3 + 0.06607s^2 + 0.001089s + 0.000002535} = G(s) \end{aligned} \tag{14}$$

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

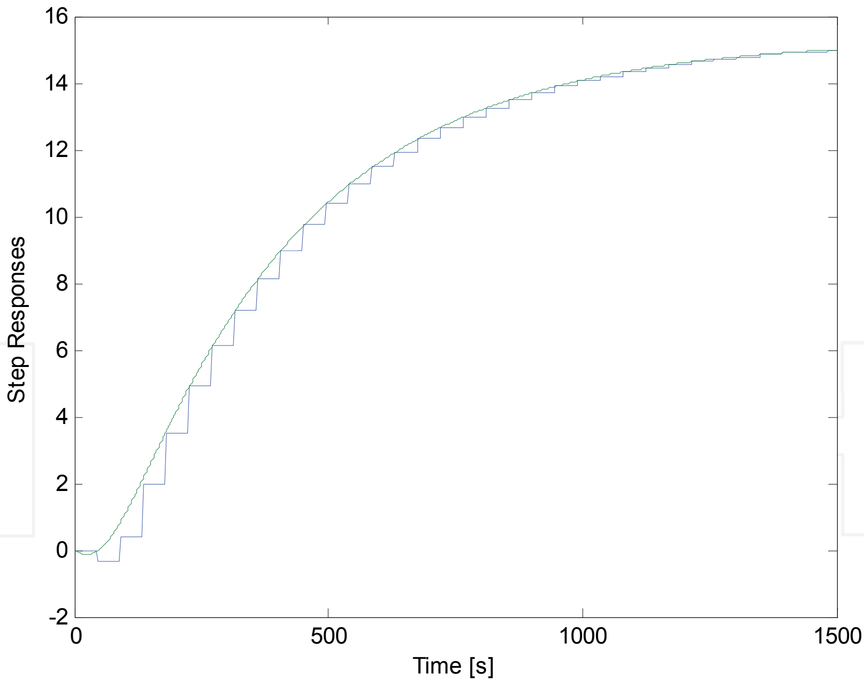

Fig. 5. Comparison of step responses of discrete-time and continuous-time models

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

$$a(s)f(s)p(s)+b(s)q(s)=m(s) \tag{15}$$

where analogically to Section 2:

$$\begin{aligned} a(s) &= s^3+0.06607s^2+0.001089s+0.000002535 \\ b(s) &= -0.007267s^2-0.0006423s+0.0000388 \\ f(s) &= s \end{aligned} \tag{16}$$

and where closed-loop characteristic polynomial has been assumed:

$$m(s)=s^6+0.133s^5+0.006765s^4+0.0001689s^3+2.14925\cdot10^{-6}s^2+1.26\cdot10^{-8}s+2.25\cdot10^{-11} \tag{17}$$

i.e. its roots are:

$$\begin{aligned} r_1 &= -0.003 \\ r_2 &= -0.05 \\ r_3 &= -0.02 \\ r_4 &= -0.02 \\ r_5 &= -0.025 \\ r_6 &= -0.015 \end{aligned} \tag{18}$$

The final continuous-time controller has been calculated as:

$$C(s)=\frac{q(s)}{f(s)p(s)}=\frac{0.2591s^3+0.01549s^2+0.0002423s+5.799\cdot10^{-7}}{s^3+0.06881s^2+0.001409s} \tag{19}$$

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

$$\frac{de(t)}{dt}\approx\frac{e(k)-e(k-1)}{T} \tag{20}$$

leads to “emulation” of continuous-time (19) suitable for Borland Pascal environment. Thus, the control law can be accomplished by relation:

$$u(k)=\left.\begin{bmatrix} q_3\left(e(k)-3e(k-1)+3e(k-2)-e(k-3)\right)+\cdots \\ \cdots+Tq_2\left(e(k)-2e(k-1)+e(k-2)\right)+T^2q_1\left(e(k)-e(k-1)\right)+\cdots \\ \cdots+T^3q_0e(k)+3u(k-1)-3u(k-2)+u(k-3)-\cdots \\ \cdots Tp_2\left(-2u(k-1)+u(k-2)\right)+T^2p_1u(k-1) \end{bmatrix}\middle/\left(1+Tp_2+T^2p_1\right)\right. \tag{21}$$

Symbol T in (20) and (21) represents sampling time, usually very short one, because the shorter sampling period means the closer approximation of continuous-time controller (19) by the equation (21). 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.

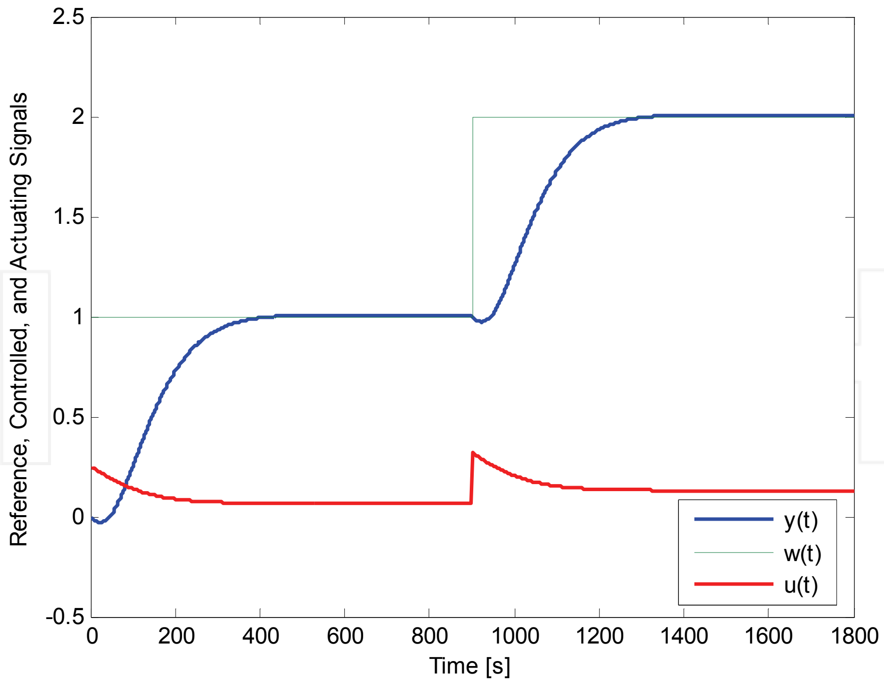

Fig. 6. Control of plant using continuous-time controller – simulation

5.2 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 (Middleton & Goodwin, 1989):

$$\delta = \frac{z-1}{T} \tag{22}$$

Consequent generalization of such models with complex variable γ has been published in (Mukhopadhyay et al., 1992). It has proved that all operators:

$$\gamma = \frac{z-1}{\lambda T z + (1-\lambda) T}; \quad 0 \le \lambda \le 1 \tag{23}$$

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

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

$\lambda = 1$ (backward model):

$$\gamma = \frac{1 - z^{-1}}{T} \tag{25}$$

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

$$\gamma = \frac{2}{T}\frac{z-1}{z+1} \tag{26}$$

The PID-B2 controller (Bobál et al., 1999); (Sysel, 2001) has been utilized in this method. It is based on structure developed in (Ortega & Kelly, 1984) which is shown in Fig. 7.

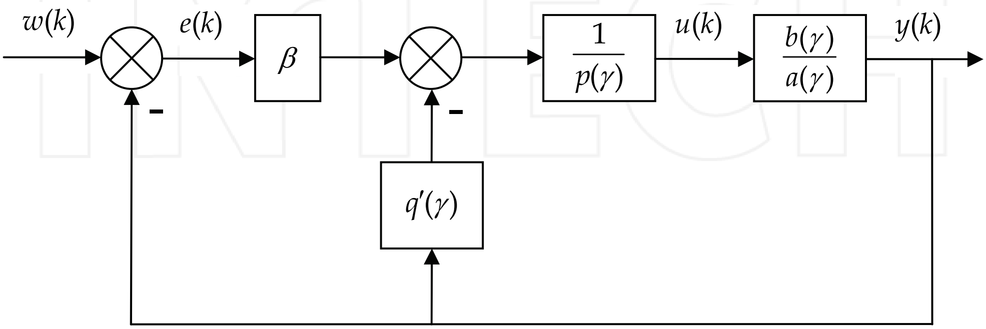

Fig. 7. Closed loop with PID-B controller

The controlled plant from Fig. 7 is supposed as:

$$\frac{b(\gamma)}{a(\gamma)} = \frac{b_1\gamma + b_0}{\gamma^2 + a_1\gamma + a_0} \tag{27}$$

and controller polynomials are:

$$\begin{aligned} p(\gamma) &= \gamma(\gamma + \lambda) \\ q'(\gamma) &= \gamma(q_2'\gamma + q_1') \end{aligned} \tag{28}$$

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

$$a(\gamma)p(\gamma) + b(\gamma)\left[q'(\gamma) + \beta\right] = m(\gamma) \tag{29}$$

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

$$m(\gamma) = (\gamma - \alpha)^2\left[\gamma - (\alpha + j\omega)\right]\left[\gamma - (\alpha - j\omega)\right] \tag{30}$$

The parameter α can serve for changing speed of control process and "aggressiveness" of actuating signal, while ω is useful for selecting size of overshoot.

However, for the sake of control loop stability, roots of the polynomial (30) 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 (Sysel, 2001):

$$\begin{aligned}&\gamma^4+\gamma^3\left[a_1+\lambda+b_1\left(q_2'+\beta\right)\right]+\gamma^2\left[a_0+a_1\lambda+b_1\left(q_1'+\frac{2\beta}{T}\right)+b_0\left(q_2'+\beta\right)\right]+\\&+\gamma\left[a_0\lambda+\frac{\beta}{T^2}b_1+b_0\left(q_1'+\frac{2\beta}{T}\right)\right]+\frac{\beta}{T^2}b_0=\\&=\gamma^4-4\gamma^3\alpha+\gamma^2\left(6\alpha^2+\omega^2\right)+\gamma\left[-2\alpha\left(2\alpha^2+\omega^2\right)\right]+\alpha^2\left(\alpha^2+\omega^2\right)\end{aligned} \tag{31}$$

More convenient matrix form is:

$$\begin{pmatrix} b_1 & 0 & b_1 & 1 \\ b_0 & b_1 & b_0+\frac{2b_1}{T} & a_1 \\ 0 & b_0 & \frac{2b_0}{T}+\frac{b_1}{T^2} & a_0 \\ 0 & 0 & \frac{b_0}{T^2} & 0 \end{pmatrix}\begin{pmatrix} q_2' \\ q_1' \\ \beta \\ \lambda \end{pmatrix}=\begin{pmatrix} -4\alpha-a_1 \\ 6\alpha^2+\omega^2-a_0 \\ -2\alpha\left(2\alpha^2+\omega^2\right) \\ \alpha^2\left(\alpha^2+\omega^2\right) \end{pmatrix} \tag{32}$$

Analytical solution suitable for Pascal implementation can look like:

$$\begin{aligned}&\beta=\frac{x_{r4}}{x_{l3}}\\&q_1'=\frac{a_0x_{r5}-\left(b_1a_1-b_0\right)x_{r6}}{a_0b_1^2-\left(b_1a_1-b_0\right)b_0}\\&\lambda=\frac{x_{r6}-b_0q_1}{a_0}\\&q_2'=\frac{x_{r1}-b_1\beta-\lambda}{b_1}\end{aligned} \tag{33}$$

where auxiliary variables are:

$$\begin{aligned}&x_{l1}=b_0+\frac{2b_1}{T}\\&x_{l2}=\frac{2b_0}{T}+\frac{b_1}{T^2}\\&x_{l3}=\frac{b_0}{T^2}\\&x_{r1}=-4\alpha-a_1\\&x_{r2}=6\alpha^2+\omega^2-a_0\\&x_{r3}=-2\alpha\left(2\alpha^2+\omega^2\right)\\&x_{r4}=\alpha^2\left(\alpha^2+\omega^2\right)\end{aligned} \tag{34}$$

and

$$\begin{aligned}&x_{r5}=-b_0x_{r1}+b_1x_{r2}+b_0b_1\beta-b_1x_{l1}\beta\\&x_{r6}=x_{r3}-x_{l2}\beta\end{aligned} \tag{35}$$

The final control law is then generated by:

$$\begin{aligned} u(k) &= \beta e(k) - q_2'\left[y(k) - 2y(k-1) + y(k-2)\right] - q_1'T\left[y(k-1) - y(k-2)\right] - \cdots \\ &\cdots \lambda T\left[u(k-1) - u(k-2)\right] + 2u(k-1) - u(k-2) \end{aligned} \tag{36}$$

while the vector of parameters $\theta(k) = \left[a_1(k) \quad a_0(k) \quad b_1(k) \quad b_0(k)\right]$ is identified using the same recursive algorithm as described in Section 3. The only modifications necessary because of delta representation are that measured output $y(k)$ is replaced by the ratio $\left[y(k) - 2y(k-1) + y(k-2)\right]/T^2$; and that the observation vector has the form $\boldsymbol{\Phi}(k) = \left[-\left(y(k-1) - y(k-2)\right)/T \quad -y(k-2) \quad \left(u(k-1) - u(k-2)\right)/T \quad u(k-2)\right]$.

6. Conclusion

This chapter 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 have been utilized. The future work should eventually lead to complete real-time industrial application.

7. Acknowledgements

The work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under Research Plan No. MSM 7088352102. 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 chapter.

8. References

Åström, K. J. & Wittenmark, B. (1973). On self-tuning regulators. *Automatica*, Vol. 9, No. 2, pp. 185-199, ISSN 0005-1098.

Åström, K. J. & Wittenmark, B. (1980). Self-tuning controller based on pole-zero placement. *IEE Proceedings*, Vol. 127, No. 3, pp. 120-130.

Åström, K. J. & Wittenmark, B. (1989). *Adaptive Control*, Addison-Wesley, Reading, Massachusetts, USA.

Åström, K. J.; Borisson, U.; Ljung, L. & Wittenmark, B. (1977). Theory and applications of self-tuning regulators. *Automatica*, Vol. 13, No. 5, pp. 457-476, ISSN 0005-1098.

Bittanti, S.; Bolzern, P. & Campi, M. (1990). Exponential Convergence of a Modified Directional Forgetting Identification Algorithm. *Systems & Control Letters*, Vol. 14, No. 2, pp. 131-137, ISSN 0167-6911.

Bobál, V.; Böhm, J.; Fessl, J. & Macháček, J. (2005). *Digital Self-tuning Controllers: Algorithms, Implementation and Applications*, Springer-Verlag London, ISBN 1-85233-980-2.

Bobál, V.; Böhm, J.; Prokop, R. & Fessl, J. (1999). *Practical Aspects of Self-Tuning Controllers: Algorithms and Implementation*, (*Praktické aspekty samočinně se nastavujících regulátorů:*

algoritmy a implementace), Brno University of Technology, VUTIUM, ISBN 80-214-1299-2, Brno, Czech Republic. (In Czech).
Čirka, Ľ.; Fikar, M. & Petruš, P. (2006). IDTOOL 4.0 – A Dynamical System Identification Toolbox for Matlab/Simulink, *Proceedings of Technical Computing Prague 2006*, Prague, Czech Republic.
Clarke, D. W. & Gawthrop, P. J. (1979). Self-tuning control. *IEE Proceedings*, Vol. 126, No. 6, pp. 633-640.
Doyle, J. C.; Francis, B. A. & Tannenbaum, A. R. (1992). *Feedback Control Theory*, Macmillan, New York, USA.
Hang, C. C.; Lee, T. H. & Ho, W. K. (1993). *Adaptive Control*, Instrument Society of America, North Carolina, USA.
Hunt, K. J.; Šebek, M. & Kučera, V. (1993). H_2 Control Problems, In: *Polynomial Methods in Optimal Control and Filtering*, Hunt, K. J. (Ed.), pp. 29-55, Peter Peregrinus Ltd., ISBN 0-86341-295-5, Stevenage, United Kingdom.
Isermann, R.; Lachmann, K.-H. & Matko, D. (1992). *Adaptive Control Systems*, Prentice Hall, Englewood Cliffs, New Jersey, USA.
Kučera, V. (1979). *Discrete Linear Control: The Polynomial Equation Approach*, John Wiley, Chichester, UK.
Kučera, V. (1991). *Analysis and Design of Discrete Linear System*, Prentice Hall, Englewood Cliffs, New Jersey, USA.
Kučera, V. (1993). Diophantine Equations in Control – A Survey. *Automatica*, Vol. 29, No. 6, pp. 1361-1375, ISSN 0005-1098.
Kulhavý, R. & Kárný, M. (1984). Tracking of Slowly Varying Parameters by Directional Forgetting, *Proceedings of the 9th IFAC World Congress*, pp. 79-83, Budapest, Hungary, 1984.
Matušů, R. & Prokop, R. (2009). Software Implementation of Discrete-Time Adaptive Control, *Proceedings of 13th International Research/Expert Conference "Trends in the Development of Machinery and Associated Technology"*, Hammamet, Tunisia, 2009, pp. 785-788, ISSN 1840-4944.
Matušů, R., Prokop, R. & Pekař, L. (2009). Polynomial Approach to Discrete-Time Adaptive Control: Software Implementation for Industrial Application, *Annals of DAAAM for 2009 & Proceedings of the 20th International DAAAM Symposium*, Vienna, Austria, pp. 523-524, ISBN 978-3-901509-70-4, ISSN 1726-9679.
Middleton, R. H. & Goodwin, G. C. (1989). *Digital Control and Estimation: A Unified Approach*, Prentice Hall, Englewood Cliffs, New Jersey, USA.
Mukhopadhyay, S.; Patra, A. & Rao, G. P. (1992). New class of discrete-time models for continuous-time systems. *International Journal of Control*, Vol. 55, No. 5, pp. 1161-1187.
Ortega, R. & Kelly, R. (1984). PID self-tuners: Some theoretical and practical aspects. *IEEE Transactions on Industrial Electronics*, Vol. 31, No. 4, pp. 332-338.
Sysel, M. (2001). *Application of delta models to process control*, (*Využití delta modelů pro řízení procesů*), Ph.D. Thesis, Faculty of Technology, Tomas Bata University in Zlín, Czech Republic. (In Czech).
Vidyasagar, M. (1985). *Control System Synthesis: A Factorization Approach*, MIT Press, Cambridge, Massachusetts, USA.
Wellstead, P. E. & Zarrop, M. B. (1991). *Self-Tuning Systems*, Wiley, New York, USA.
Wellstead, P. E.; Prager, D. I. & Zanker, P. M. (1979). A pole assignment self-tuning regulator. *IEE Proceedings*, Vol. 126, No. 8, pp. 781-787.

New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems

Edited by Meng Joo Er

ISBN 978-953-307-213-5
Hard cover, 438 pages
Publisher Sciyo
Published online 02, November, 2010
Published in print edition November, 2010

The grandest accomplishments of engineering took place in the twentieth century. The widespread development and distribution of electricity and clean water, automobiles and airplanes, radio and television, spacecraft and lasers, antibiotics and medical imaging, computers and the Internet are just some of the highlights from a century in which engineering revolutionized and improved virtually every aspect of human life. In this book, the authors provide a glimpse of the new trends of technologies pertaining to control, management, computational intelligence and network systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Radek Matusu and Roman Prokop (2010). Software Implementation of Self-Tuning Controllers, New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems, Meng Joo Er (Ed.), ISBN: 978-953-307-213-5, InTech, Available from: http://www.intechopen.com/books/new-trends-in-technologies--control--management--computational-intelligence-and-network-systems/software-implementation-of-self-tuning-controllers

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821