NEW MODIFICATION OF MATLAB-TOOLBOX FOR CAD OF SIMPLE ADAPTIVE CONTROLLERS

Vladimír Bobál, Petr Chalupa

Department of Control Theory, Institute of Information Technologies,

Tomas Bata University in Zlin, Nám. TGM 275, 762 72 Zlín, Czech Republic,

Tel.:+420 67 754 3217, E-mail: bobal@ft.utb.cz

Abstract: The contribution presents a MATLAB-Toolbox for design, simulation and verification of single input single output (SISO) discrete self-tuning controllers. The proposed adaptive controllers who are included into a Toolbox can be divided into two parts. The first part covers PID adaptive algorithms using traditional Ziegler-Nichols method for the setting of the controller parameters, the second part of described controllers is based on the pole placement design. It is considered as a SIMULINK block library of individual adaptive controllers and it is a completely open system based on SIMULINK features. The MATLAB-Toolbox is very successfully used in Adaptive Control Course in education practice for design, simulation and verification of self - tuning control systems in real - time conditions. The Toolbox is supplemented by very friendly User's Manual. This Toolbox is suitable for design and verification of the industrial adaptive controllers, too.

Key words: Self-tuning control; ARX model; recursive least squares; PID control; pole assignment; MATLAB–Toolbox.

1. INTRODUCTION

Adaptive control methods have been developing significantly over the past tree decades. The aim of this research was to solve the problem of designing a controller for systems where the characteristics are not completely known or vary. Different approaches were proposed and utilized. One successful approach is represented by self-tuning controllers (STC). The main idea of an STC is based on the combination of a recursive identification procedure and a selected controller synthesis. Different branches of the STC differ mainly in the method of the controller design while the identification recursive least squares method (RLSM) applied to a regression (ARX) model forms a standard.

Despite intensive research activity, there is a lack of controllers, which are able to cope with practical requirements. Industrial applications on a large scale are still missing, however, many realized cases were successful. It is evident that until to now application has required a control engineer skilled in specific technology as well as versed in modern control methods.

Also the aim of this contribution is to bridge the gap between theory and practice and to present some simple controller algorithms in a form acceptable for industrial users. The theoretical background not only these simple algorithms, also algorithms based on polynomial solutions of controller synthesis and controllers what are derived from the use of minimization of linear quadratic criterion, are given in Bobál, *et al.* (1999a).

All the explicit self-tuning controllers who are included into MATLAB-Toolbox have been algorithmically modified in the form of mathematical relations or as flow diagrams so as to make them easy to program and apply. Some are original algorithms based on a modified Ziegler-Nichols criterion, others have been culled from publications and adapted to make them more accessible to the user.

2. THEORETICAL BACKGROUND

2.1 Recursive identification

In the identification part of the designed controller algorithms the regression (ARX) model of the following form

$$y(k) = \Theta^T(k)\phi(k-1) + n(k) \tag{1}$$

is used, where

$$\Theta^T(k) = [a_1\ a_2\ \dots\ a_{na}\ b_1\ b_2\ \dots\ b_{nb}] \tag{2}$$

is the vector of the parameter estimates and

$$\phi^T(k-1) = [-y(k-1) - y(k-2) \dots - y(k-na)\ u(k-1)\ u(k-2) \dots u(k-nb)] \tag{3}$$

is the regression vector ($y(k)$ is the process output variable, $u(k)$ is the controller output variable). The non-measurable random component $n(k)$ is assumed to have zero mean value $E[n(k)] = 0$ and constant covariance (dispersion) $R = E[n^2(k)]$. For calculating of the parameter estimates (2) is utilized the recursive least squares method, where adaptation is supported by directional forgetting (Kulhavý, 1987).

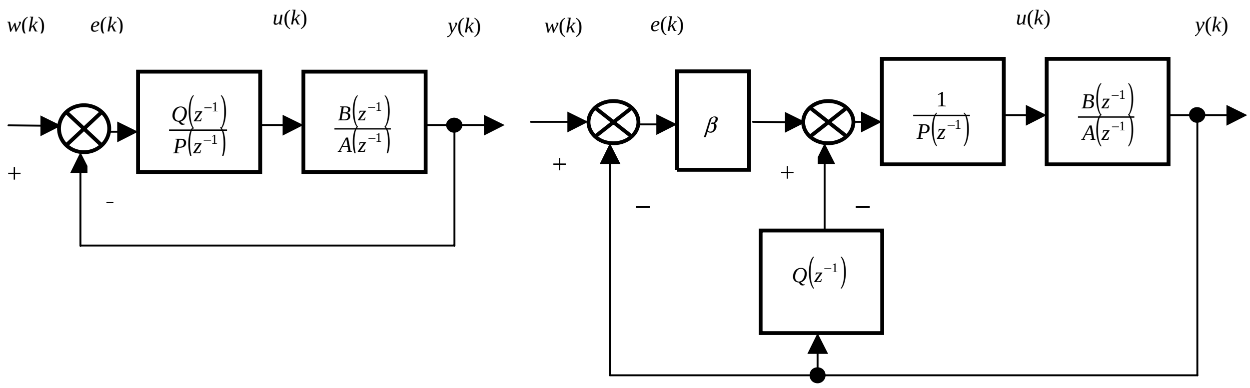
2.2 Algorithms of digital Ziegler-Nichols PID controllers

The PID controllers are still widely used in industry. These types of controllers are more convenient for users owing to their simplicity of implementation, which is generally well known. Provided the controller parameters are well chosen they can control a considerable part of continuous technological processes. To get a digital version of the PID controller, it is necessary to discretize the integral and derivative component of the continuous-time controller. For discretizing the integral component we usually employ the forward rectangular method (FRM), backward rectangular method (BRM) or trapezoidal method (TRM). The derivative component is mostly replaced by the 1st order difference (two-point difference). For practical use the recurrent control algorithms which compute the actual value of the controller output $u(k)$ from the previous value $u(k\text{-}1)$ and from compensation increment seem to be suitable

$$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1) \tag{4}$$

where q_0, q_1, q_2 are the controller parameters and T_0 is the sampling period.

It is subsequently possible to derive further variants of digital PID controllers. For example the PID controller with a filter constant in the D-part (Isermann, 1989), Takahashi's PID controller (Takahashi, *et al.*, 1971), the PID controller based on the ideas of Åström, *et al.* (1992), Dahlin's PID controller (Dahlin, 1968) or the PID controller designed by Bányász and Keviczky (1993).



a) PID - A controller b) PID - B controller

Fig. 1 Block diagrams of control loops with pole placement PID controllers

First group of proposed self-tuning PID controllers is based on the classical Ziegler and Nichols (1942) method. In this well-known approach the parameters of the controller are calculated from the ultimate (critical) gain K_{pu} and the ultimate period of oscillations T_u of the closed loop system. The analytical expressions for computing of these critical parameters are derived in Bobál, *et al.* (1999a) and Bobál, *et al.* (1999b). The flow diagrams for the computing of the critical parameters for the second- and for the third-order model are introduced in the same publications.

2.3 Pole placement PID controllers

A controller based on the pole placement method in a closed feedback control loop is designed to stabilise the closed control loop whilst the characteristic polynomial should have a previously determined pole. Digital PID controllers is possible can be expressed in the form of a discrete transfer function

$$G_R(z) = \frac{U(z)}{E(z)} = \frac{Q(z^{-1})}{P(z^{-1})} = \frac{q_0 + q_1 z^{-1} + q_2 z^{-2}}{(1 - z^{-1})(1 + \gamma z^{-1}]} \tag{5}$$

If the process is given by the transfer function

$$G_P(z) = \frac{Y(z)}{U(z)} = \frac{B(z^{-1})}{A(z^{-1})} \tag{6}$$

with the polynomials $A(z^{-1})$, $B(z^{-1})$ for degree n = 2 then the characteristic polynomial of the closed-loop system with a PID-A controller (see Fig. 1a)) is in the form

$$A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1}) = D(z^{-1}) \tag{7}$$

The pole placement of the characteristic polynomial (7) determines the dynamic behaviour of the closed - loop system. The characteristic polynomial $D(z^{-1})$ can be specified by different methods. It is usually described by the dominant poles for a second order continuous model (controller PID A-1)

$$D(s) = s^2 + 2\xi\omega_n s + \omega_n^2 \tag{8}$$

The dominant poles are given by the desired damping factor ξ and the natural frequency ω_n of the closed-loop.

A PID controller with the characteristic polynomial (controller PID A-2)

$$D(z) = (z - \alpha)^2 [z - (\alpha + j\omega)][z - (\alpha - j\omega)] \tag{9}$$

is proposed in Bobál, *et al*. (1999a). Characteristic polynomial (9) has a double real pole $z_{1,2} = \alpha$ within interval $0 \le \alpha < 1$ and a pair of complex conjugated poles $z_{3,4} = \alpha \pm j\omega$, where $\alpha^2 + \omega^2 < 1$. The parameter α influences the speed of the control-loop transient response and influences controller output changes, too. By changing parameter ω it is possible to influence the desired overshoot.

The structure of the control loop with controller PID B is shown in Fig. 1b). The characteristic polynomial has in this case form

$$A(z^{-1})P(z^{-1}) + B(z^{-1})[Q'(z^{-1}) + \beta] = D(z^{-1}) \tag{10}$$

where polynomial $P(z^{-1})$ has the same form as in the transfer function (5) and second polynomial in equation (10) is given

$$Q'(z^{-1}) = (1 - z^{-1})(q_0' - q_2' z^{-1}) \tag{11}$$

Using equation (7) it is possible to derived controller PID A-2, using equation (10) controller PID B-2.

3. TOOLBOX DESCRIPTION

First modification of this Toolbox has been programmed in the version MATLAB 4.2. This modification made use of a GUI. Here all actions are guided by prepared graphical windows that checks any action and leads the user step by step to correct settings. This version is typically efficient in education.

Because the individual GUI are depended on the individual MATLAB versions, the second modification is based on the use a SIMULINK block library of individual adaptive controllers. In this case it is a completely open system based on SIMULINK features (Bobál and Böhm, 2000).

These two versions differ in the upper level appearance, however, use the same or slightly different lower level functions. The SIMULINK oriented part uses only SIMULINK features. The choice of a specific controller, setting its parameters connection to the system, use of additional blocks etc. all must be done by user. Compared with a GUI form this offers, on one side, large flexibility, on the other side, it requires some knowledge and responsibility to avoid misuse of the controller.

The survey of the individual controllers with name of the m-files and requisite input parameters is contained in Table 1. The detailed description of all controller algorithms with the recursive identification procedure is introduced in Bobál, *et al.* (1999a). This Toolbox is suitable for real-time control using the acquisition card ADVANTECH PCL-812 PG with Extended Real Time Toolbox (see User's Manual, 1996), too.

Table 1 Survey of individual controllers

Contr. No. **m-file**	**Controller algorithm**	**Input parameters** **Controller type**
1 zn2fr.m zn3fr.m	$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1)$ $q_0 = K_P\left(1 + \frac{T_D}{T_0}\right)$; $q_1 = -K_P\left(1 - \frac{T_0}{T_I} + 2\frac{T_D}{T_0}\right)$; $q_2 = K_P \frac{T_D}{T_0}$	K_{Pu}; T_u; T_0 PID – forward rectangular method
2 zn2br.m zn3br.m	$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1)$ $q_0 = K_P\left(1 + \frac{T_0}{T_I} + \frac{T_D}{T_0}\right)$; $q_1 = -K_P\left(1 + 2\frac{T_D}{T_0}\right)$; $q_2 = K_P \frac{T_D}{T_0}$	K_{Pu}; T_u; T_0 PID – backward rectangular method
3 zn2tr.m zn3tr.m	$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1)$ $q_0 = K_P\left(1 + \frac{T_0}{2T_I} + \frac{T_D}{T_0}\right)$; $q_1 = -K_P\left(1 - \frac{T_0}{2T_I} + \frac{2T_D}{T_0}\right)$; $q_2 = K_P \frac{T_D}{T_0}$	K_{Pu}; T_u; T_0 PID - trapezoidal rectangular method
4 zn2fpd.m zn3fpd.m	$u(k) = K_P\left\{e(k) - e(k-1) + \frac{T_0}{T_I}e(k) + \frac{T_D}{6T_0}\left[e(k) + 2e(k-1) - 6e(k-2) + 2e(k-3) + e(k-4)\right]\right\} +$ $+ u(k-1)$	K_{Pu}; T_u; T_0 PID - backward rectangular method, replacing derivation by a four- point difference
5 zn2fd.m zn3fd.m	$u(k) = p_1 u(k-1) + p_2 u(k-2) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$ $p_1 = \frac{-4\frac{T_f}{T_0}}{\frac{2T_f}{T_0} + 1}$; $p_2 = \frac{\frac{2T_f}{T_0} - 1}{\frac{2T_f}{T_0} + 1}$; $q_0 = \frac{K_P + 2K_P\frac{T_f + T_D}{T_0} + \frac{K_P T_0}{2T_I}(\frac{2T_f}{T_0} + 1)}{\frac{2T_f}{T_0} + 1}$ $q_1 = \frac{\frac{K_P T_0}{2T_I} - 4K_P\frac{T_f + T_D}{T_0}}{\frac{2T_f}{T_0} + 1}$; $q_2 = \frac{\frac{T_f}{T_0}(2K_P - \frac{K_P T_0}{T_I}) + 2\frac{K_P T_D}{T_0} + \frac{K_P T_0}{2T_I} - K_P}{\frac{2T_f}{T_0} + 1}$; $T_f = \frac{T_D}{\alpha}$; $\alpha \in \langle 3;20\rangle$	K_{Pu}; T_u; T_0; T_f PID - filtration of D-component, Tustin approximation
6 zn2tak.m zn3tak.m	$u(k) = K_R[y(k-1) - y(k)] + K_I[w(k) - y(k)] + K_D[2y(k-1) - y(k-2) - y(k)] +$ $+ u(k-1)$ $K_R = 0.6K_{Pu} - \frac{K_I}{2}$; $K_I = \frac{1.2K_{Pu}T_0}{T_u}$; $K_D = \frac{3K_{Pu}T_u}{40T_0}$	K_{Pu}; T_u; T_0 PID - Takahashi's controller
7 zn2sic.m zn3sic.m	$u(k) = u_P(k) + u_I(k) + u_D(k)$ $u_P(k) = K_P e(k)$; $u_I(k) = K_P\frac{T_0}{T_I}e(k) + u_I(k-1)$; $u_D(k) = K_P\frac{T_D}{T_0}[e(k) - e(k-1)]$	K_{Pu}; T_u; T_0 PID - component form
8 zn2ast.m zn3ast.m	$u(k) = u_{PI}(k) + u_D(k)$ $u_{PI}(k) = K_P[y(k-1) - y(k)] + \frac{K_P T_0}{2T_I}[e(k) + e(k-1)] + \beta K_P[w(k) - w(k-1)] + u_{PI}(k-1)$ $u_D(k) = K_P\frac{T_D\alpha}{T_D + T_0\alpha}[y(k-1) - y(k)] + \frac{T_D}{T_D + T_0\alpha}u_D(k-1)$	K_{Pu}; T_u; T_0 PID - Åström's controller
9 ba2.m	$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + u(k-1)$ $q_0 = \frac{k_I}{b_0}$; $q_1 = q_0 a_1 = \frac{k_I}{b_0}a_1$; $q_2 = q_0 a_2 = \frac{k_I}{b_0}a_2$ $k_I = \frac{1}{2d - 1}$ $(\gamma = 0)$; $k_I = \frac{1}{2d(1+\gamma)(1-\gamma)}$ $(\gamma > 0)$; $\gamma = \frac{b_1}{b_0}$	d - time delay PID - Bányász - Keviczky's controller
10 da2.m	$u(k) = K_P\left\{e(k) - e(k-1) + \frac{T_0}{T_I}e(k) + \frac{T_D}{T_0}[e(k) - 2e(k-1) + e(k-2)]\right\} + u(k-1)$ $K_P = -\frac{(a_1 + 2a_2)Q}{b_1}$; $T_I = -\frac{T_0}{\frac{1}{a_1 + 2a_2} + 1 + \frac{T_D}{T_0}}$ $T_D = \frac{T_0 a_2 Q}{K_P b_1}$; $Q = 1 - e^{-\frac{T_0}{B}}$	B - adjustment factor T_0 PID - Dahlin's controller
11 pp2a_1.m	$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + (1-\gamma)u(k-1) + \gamma u(k-2)$ $q_0 = \frac{1}{b_1}(d_1 + 1 - a_1 - \gamma)$; $q_1 = \frac{a_2}{b_2} - q_2(\frac{b_1}{b_2} - \frac{a_1}{a_2} + 1)$; $q_2 = -\frac{s_1}{r_1}$; $\gamma = q_2\frac{b_2}{a_2}$ $s_1 = a_2\left[(b_1 + b_2)(a_1 b_2 - a_2 b_1) + b_2(b_1 d_2 - b_2 d_1 - b_2)\right]$; $r_1 = (b_1 + b_2)(a_1 b_1 b_2 + a_2 b_1^2 + b_2^2)$	ω_n - natural frequency ξ - damping factor PID A-1 pole placement controller

12 pp2a_2.m	$u(k)=q_0e(k)+q_1e(k-1)+q_2e(k-2)+(1-\gamma)u(k-1)+\gamma u(k-2)$ $q_0=\frac{r_2-r_3}{r_1}$; $q_1=-\frac{r_4+r_5}{r_1}$; $q_2=\frac{x_4+\gamma a_2}{b_2}$; $\gamma=\frac{r_6}{r_1}$ $r_1=(b_1+b_2)(a_1b_1b_2+a_2b_1^2+b_2^2)$; $r_2=x_1(b_1+b_2)(a_1b_2-a_2b_1)$ $r_3=b_1^2x_4-b_2\left[b_1x_3-b_2(x_1+x_2)\right]$; $r_4=a_1\left[b_1^2x_4+b_2^2x_1-b_1b_2(x_2+x_3)\right]$ $r_5=(b_1+b_2)\left[a_2(b_1x_2-b_2x_1)-b_1x_4+b_2x_3\right]$; $r_6=b_1(b_1^2x_4-b_1b_2x_3+b_2^2x_2)-b_2^3x_1$	α, ω - real and imaginary component of the pole PID A-2 pole placement controller
13 pp2b_1.m	$u(k)=-\left[(q_0'+\beta)y(k)-(q_0'+q_2')y(k-1)+q_2'y(k-2)\right]-$ $-(\gamma-1)u(k-1)+\gamma u(k-2)+\beta w(k)$ $q_0'=q_2'(\frac{b_1}{b_2}-\frac{a_1}{a_2})-\frac{a_2}{b_2}$; $q_2'=\frac{s_1}{r_1}$; $\gamma=q_2'\frac{b_2}{a_2}$; $\beta=\frac{1}{b_1}(d_1+1-a_1-\gamma-b_1q_0')$ $s_1=a_2\left\{b_2\left[a_1(b_1+b_2)+b_1(d_2-a_2)-b_2(d_1+1)\right]-a_2b_1^2\right\}$ $r_1=(b_1+b_2)(a_1b_1b_2-a_2b_1^2-b_2^2)$	ω_a - natural frequency ξ - damping factor PID B-1 pole placement controller
14 pp2b_2.m	$u(k)=-\left[(q_0'+\beta)y(k)-(q_0'+q_2')y(k-1)+q_2'y(k-2)\right]-$ $-(\gamma-1)u(k-1)+\gamma u(k-2)+\beta w(k)$ $q_0'=-\frac{r_2-r_3+r_4}{r_1}$; $q_2'=\frac{r_6+r_7}{r_1}$; $\gamma=\frac{r_5}{r_1}$; $\beta=\frac{x_1+x_2-x_3+x_4}{b_1+b_2}$ $r_1=(b_1+b_2)(a_1b_1b_2-a_2b_1^2-b_2^2)$; $r_2=a_1b_2\left[b_1(x_2+x_3+x_4)-b_2x_1\right]$ $r_3=a_2b_1\left[b_2x_1-b_1(x_2-x_3+x_4)\right]$; $r_4=(b_1+b_2)\left[b_1x_4+b_2(-x_3-x_4)\right]$ $r_5=b_1(b_1^2x_4-b_1b_2x_3+b_2^2x_2)-b_2^3x_1$; $r_6=b_1^2(-a_2x_3+a_1x_4-a_2x_4)$ $r_7=b_2\left[b_1(a_1x_4+a_2x_2-x_4)-b_2(a_2x_1+x_4)\right]$	α, ω - real and imaginary component of the pole PID B-2 pole placement controller
15 zn2pd.m	$u(k)=K_P\left\{w(k)-y(k)+\frac{T_D}{T_0}\left[y(k-1)-y(k)\right]\right\}$ $K_P=0.4K_{Pu}$; $T_D=\frac{T_U}{20}$	K_{Pu}; T_u; T_0 PD controller
16 zn2pi.m	$u(k)=q_0e(k)+q_1e(k-1)+u(k-1)$ $q_0=K_P\left(1+\frac{T_0}{2T_I}\right)$; $q_1=-K_P\left(1-\frac{T_0}{2T_I}\right)$	K_{Pu}; T_u; T_0 PI controller
17 mv2.m	$u(k)=\frac{1}{q}\left[a_1y(k-1)+a_2y(k-2)-b_1u(k-1)-b_2u(k-2)+w(k)\right]+u(k-1)$	q - penalisation factor minimum variance controller

Notes:

m-file name: **xx(x)a(yyyy).m**

xx(x) - controller type, **n** = 2 or 3 - process model order, **(yyyy)** - controller further details

Controllers number 1,2,3,4,6,7,8,16 and 17:

$K_P=0.6K_{Pu}$; $T_I=0.5T_u$; $T_D=0.125T_u$

Controllers number 11 and 13:

$d_1=-2\exp(-\xi\omega_nT_0)\cos(\omega_nT_0\sqrt{1-\xi^2})$ for $\xi \le 1$, $d_1=-2\exp(-\xi\omega_nT_0)\cosh(\omega_nT_0\sqrt{1-\xi^2})$ for $\xi > 1$,
$d_2=\exp(-2\xi\omega_nT_0)$

Controllers number 12 and 14:

$x_1=c+1-a_1$; $x_2=d+a_1-a_2$; $x_3=f+a_2$; $x_4=g$

$c=-4\alpha$; $d=6\alpha^2+\omega^2$; $f=-2\alpha(2\alpha^2+\omega^2)$; $g=\alpha^2(\alpha^2+\omega^2)$

4. CONCLUSIONS

The Toolbox is widely used in the education, laboratory training and some practical applications. Its structure can address various users. The example of a typical SIMULINK scheme of adaptive control – loop using the digital PID controller based on the Zigler-Nichols method with discretization using backward rectangular method, replacing derivation by a four- point difference is shown in Fig. 2.

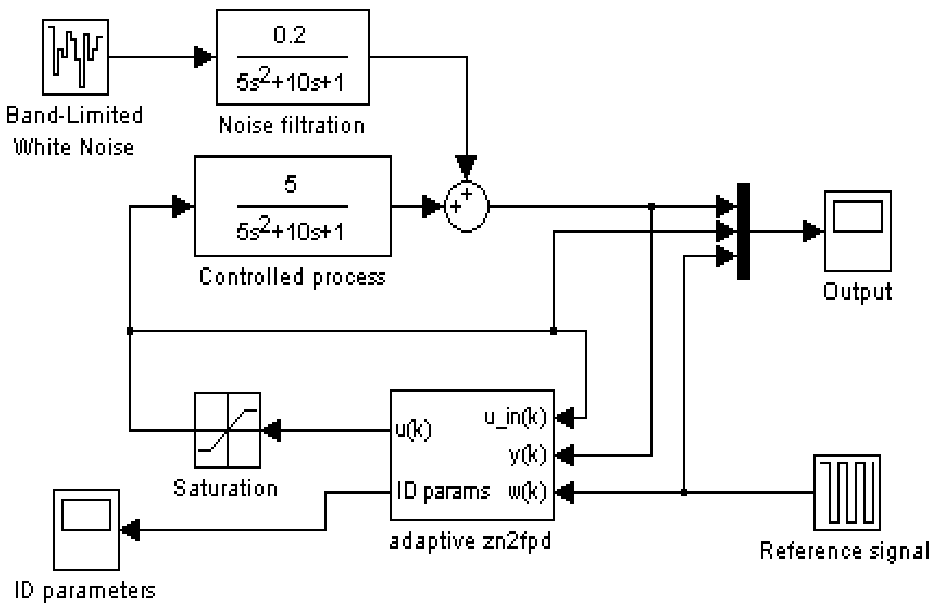


Fig. 2 Typical SIMULINK scheme of self-tuning controller

The MATLAB - Toolbox is used not only for simulation of designed self-tuning controllers but it is used for the control a number of laboratory equipments in practical courses. PID controllers have been implemented e. g. for control of the coupled tanks apparatus, ball and plate model, electronic analog models, laboratory thermo – analyser, laboratory through – flow heater etc.

The purpose of this contribution was to arrange the brief informative User's Manual about MATLAB-Toolbox. This Toolbox is free and will be installed from 1 November 2001 on the www pages:

www.utb.cz/stctool

ACKNOWLEDGEMENTS

This work was supported in part by the Grant Agency of the Czech Republic under grants No.102/99/1292, 102/00/0526 and in part by the Ministry of Education of the Czech Republic under grant MSM 281100001.

REFERENCES

Åström, K. J., C. Hang, P. Pearson and W. K. Ho (1992). Towards intelligent PID control, *Automatica*, **28**, 1-9.

Bányász, Cs. and L. Keviczky (1993). Design of adaptive PID regulators based on recursive estimation of the process parameters. *Journal of Process Control*, **3**, 1993, 53-59.

Bobál, V., J. Böhm, J. Prokop and J. Fessl (1999a). *Practical Aspects of Self-Tuning Controllers: Algorithms and Implementation*. VUTIUM Press, Brno University of Technology, Brno, (in Czech).

Bobál, V., J. Böhm, and R. Prokop (1999b). Practical aspects of self-tuning controllers. *International Journal of Adaptive Control and Signal Processing*, **13**, 1999, 671-690.

Bobál, V. and J. Böhm (2000). Matlab - Toolbox for design and verification of self - tuning controllers. Preprintd of IFAC Symposium on Advanced in Control Education, Gold Coast, Australia, Paper 10S-1.

Dahlin, D. B. (1968). Designing and tuning digital controllers. Inst. Cont. Syst., 42, 1968, 77-83.

Isermann, R., *Digital Control Systems*, Springer Verlag, Berlin, 1989.

Kulhavý, R. (1987). Restricted exponential forgetting in real time identification. *Automatica*, **23,** 586-600.

Takahashi, Y., C. Chan and D. Auslander (1971). Parametereinstellung bei linearen DDC-Algorithmen, *Regelungstech. Prozessdatenver.*, **19**, 237-284.

User's Manual (1996). *Real Time Toolbox for Use with MATLAB.* Humusoft s. r. o. Praha.

Ziegler, J. G. and N. B. Nichols (1942). Optimum settings for automatic controllers. *Trans. ASME*, **64**, 759-768.