LIBRARY FOR DESIGN AND SIMULATION VERIFICATION OF SELF-TUNING CONTROLLERS

Vladimír Bobál, Petr Chalupa and Petr Dostál
Department of Process Control, Faculty of Applied Informatics
Tomas Bata University in Zlín
Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
E-mail:bobal@fai.utb.cz

KEYWORDS

Self-tuning control, ARX model, recursive least squares, PID control, polynomial approach, MATLAB–Simulink, simulation verification

ABSTRACT

This contribution present structure and usage of Self-tuning Controllers Simulink Library (STCSL). The STCSL was created for design, simulation verification and especially real-time implementation of single input - single output (SISO) digital self-tuning controllers. The proposed adaptive controllers which are included in library can be divided into three groups. The first group covers PID adaptive algorithms using traditional Ziegler-Nichols method for setting controller parameters, the second group of described controllers is based on the polynomial approach and the third group contains the controllers derived on the other approaches (minimum variance etc.). The controllers are implemented as an encapsulated Simulink blocks and thus allows users simple integration into existing Simulink schemas. The process of developing real-time applications using MATLAB Real-Time Workshop and several control courses is also presented. This Library is very successfully used in Adaptive Control Course in education practice for design and verification of self-tuning control systems in simulation and real-time conditions. It is suitable also for design and verification of the industrial digital controllers. The STCSL is available free of charge at Internet site - **http://www.utb.cz/stctool/**.

INTRODUCTION

Self-tuning controllers belong by their character to the class of adaptive control systems. The main aim of the adaptive control approach is to solve the control problem in cases where the characteristics of a controlled system are unknown or time variable. The basic principle of the adaptive control system is to change the controller characteristics based on the characteristics of control process. Self-tuning controllers use the combination of the recursive process identification on base of a selected model process and the controller synthesis based on knowledge of parameter estimates of controlled process.

The general task of optimal adaptive control with on-line process identification is very complicated and thus the method of **the forced separation of the identification and the control** is often used for the design of self-tuning controllers. The principle of this simplification is in the cyclic repeating of the following steps:

1. The process parameters are assumed to be known for the current control step and equal to their current estimations.
2. The control strategy for selected criterion of control quality is designed based on the previous assumption and controller output is calculated.
3. The next identification step is performed after the obtaining the new sample of the controlled variable (eventually the external measured disturbance) – the parameters of controlled process model are recomputed using a recursive identification algorithm.

The aim of this contribution is also to inform potential users about Self-tuning Controllers Simulink Library - STCSL (see Bobál and Chalupa, 2002) and to help for practical usage in the simulation and real time conditions. The individual self-tuning algorithms are introduced in the brief form in User's Guide that is attached into the STCSL. This library can be also the suitable bridge between theory and practice by presenting some digital controller algorithms in a form acceptable for industrial users. The theoretical background of self-tuning controllers which are contained in the STCSL is given in Bobál, *et al.* (1999, 2005).

All the explicit self-tuning controllers that are included into STCSL 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.

RECURSIVE IDENTIFICATION

The regression (ARX) model of the following form

$$y(k) = \boldsymbol{\Theta}^T(k)\boldsymbol{\Phi}(k-1) + n(k) \qquad (1)$$

Proceedings 22nd European Conference on Modelling and Simulation ©ECMS Loucas S. Louca, Yiorgos Chrysanthou, Zuzana Oplatková, Khalid Al-Begain (Editors)
ISBN: 978-0-9553018-5-8 / ISBN: 978-0-9553018-6-5 (CD)

is used in the identification part of the designed controller algorithms, where

$$\boldsymbol{\Theta}^T(k) = [a_1, a_2, \ldots, a_{na}, b_1, b_2, \ldots, b_{nb}] \qquad (2)$$

is the vector of the parameters and

$$\boldsymbol{\Phi}^T(k-1) = [-y(k-1), -y(k-2), \ldots, -y(k-na), \\ u(k-1), u(k-2), \ldots, u(k-nb)] \qquad (3)$$

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) $R = E[n^2(k)]$.

The recursive least squares method for calculating of parameter estimates $\hat{\boldsymbol{\Theta}}(k)$ is utilized. Using the pure least squares method, the influence of all pairs of identified system inputs and outputs to the parameters estimates is the same. This property can be inconvenient for example when identifying the system with time-varying parameters. In this case, it is better to use least squares method with exponential forgetting where the influence of latter data to the calculation of the parameter estimates is greater then the influence of older data.

The exponential forgetting method can be further improved by adaptive directional forgetting (Kulhavý, 1987) which changes forgetting coefficient with respect to changes of input and output signal.

CONTROLLER ALGORITHMS

The proposed self-tuning controllers which are included in the library are divided into three groups:

- classical digital PI and PID controllers whose tuning is based on Ziegler-Nichols method, pole assignment method and its modifications,
- controllers based on polynomial approach (dead-beat strong and weak version, pole assignment, controllers based on minimization of quadratic criterion),
- controllers based on other approaches (minimum variance controllers, Dahlin's controller, Bányász-Keviczky's controller etc.).

Algorithms of digital PID Ziegler-Nichols controllers

The PID controllers are still widely used in industry. These types of controllers are more convenient for users due to their simple 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). 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 for practical use

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

where q_0, q_1, q_2 are the controller parameters. As a matter of fact, the advantage of algorithm (4) is that it is not necessary to storage last input and output data in the computer storage.

It is subsequently possible to derive further variants of digital 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.* (1999, 2005).

Algorithms of PID pole placement 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. It is possible to express the digital controllers in the form of a discrete transfer function

$$G_R(z) = \frac{U(z)}{E(z)} = \frac{Q(z^{-1})}{P(z^{-1})} \qquad (5)$$

Let the controlled process be given by the transfer function

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

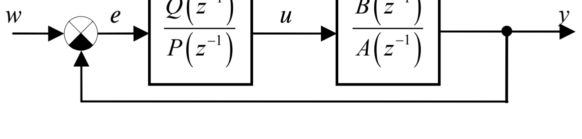


Figure 1: Control loop with PID-A controller

PID - A controller structure. This controller structure is shown in Fig. 1, then the transfer function of closed loop circuit is

$$G_W(z)=\frac{Y(z^{-1})}{W(z^{-1})}=\frac{B(z^{-1})Q(z^{-1})}{A(z^{-1})P(z^{-1})+B(z^{-1})Q(z^{-1})} \qquad (7)$$

and the characteristic polynomial of the closed-loop system with a PID-A controller is in the form

$$A(z^{-1})P(z^{-1})+B(z^{-1})Q(z^{-1})=D(z^{-1}) \qquad (8)$$

The pole placement of characteristic polynomial (8) determines the dynamic behaviour of the closed - loop system. The characteristic polynomial $D(z^{-1})$ can be specified by different methods.

PID - B controller structure. The structure of the control loop with controller PID-B is shown in Fig. 2). The characteristic polynomial has in this case form

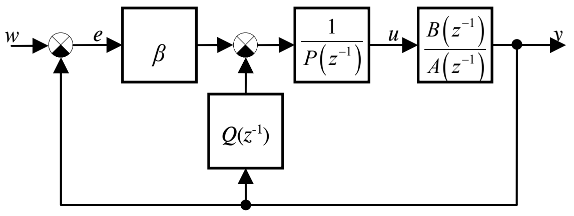


Figure 2: Control loop with PID-B controller

$$A(z^{-1})P(z^{-1})+B(z^{-1})\left[Q(z^{-1})+\beta\right]=D(z^{-1}) \qquad (9)$$

where β is feed forward controller parameter. PID – B controller structure is in principle 2DOF (two-degree freedom) configuration.

Self-tuning controllers based on polynomial approach

Different forms of self-tuning controllers could be suggested according to the chosen type of a plant model (and, consequently according to the chosen identification method), according to a criterion of quality, according to a mathematical procedure during deriving of controller equations, etc. Next group of described controllers is based on the polynomial approach. These algorithms succeed various criterions for a control process course – dead-beat method, pole assignment method and LQ (linear-quadratic) control method. The design of these controllers results from the basic block diagram of 2DOF configuration (see Fig. 3). From Fig. 3 it is obvious that the controller equation has the form

$$P(z^{-1})K(z^{-1})u(k)=R(z^{-1})w(k)-Q(z^{-1})y(k) \qquad (10)$$

where $K(z^{-1})=1-z^{-1}$ is integrator.

Then the transfer function of closed loop 2DOF circuit is

$$G_W(z)=\frac{Y(z^{-1})}{W(z^{-1})}=\frac{R(z^{-1})B(z^{-1})}{A(z^{-1})K(z^{-1})P(z^{-1})+B(z^{-1})Q(z^{-1})} \qquad (11)$$

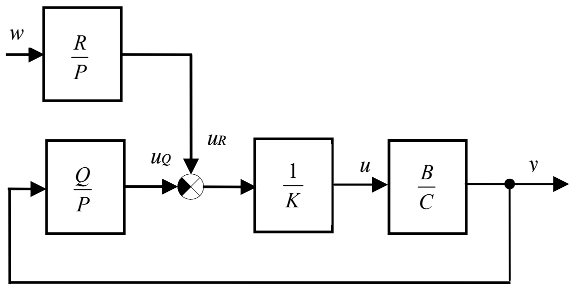


Figure 3: Closed loop 2DOF control system

The special case is a controller 1DOF (one degree of freedom - see Fig. 1)), which is valid for $R(z^{-1}) = Q(z^{-1})$ and which works with a tracking error $e(k) = w(k) - y(k)$. It could be proved that such a controller is only suboptimal for tasks of a reference value tracking.

The process of computing coefficients of polynomials $Q(z^{-1})$ and $P(z^{-1})$ is similar to 1DOF case. The presence of feedforward part brings the possibility of simpler processing of reference signal. If the 1DOF and 2DOF controllers have to fulfil the same requirements when controlling the same system, the degrees of polynomials in 1DOF control are usually higher then in 2DOF control.

SELF-TUNING CONTROLLERS SIMULINK LIBRARY

The Simulink is nowadays a word-wide standard in simulation, testing, and verification of behaviour of various dynamic systems. Simulink is a part of MATLAB system and supports linear or nonlinear systems modelled in continuous time, sampled time or a hybrid of the two. Systems can also be multirate, i.e. have different parts that are sampled or updated at different rates.

Based on monograph Bobál, et al. (2005) a library of self-tuning controllers in MATLAB/Simulink environment was created. The purpose was to create a framework suitable for creating and testing of self-tuning controllers. The library is available free of charge at internet site of Tomas Bata University in Zlín – www.utb.cz/stctool/ (see Bobál and Chalupa, 2002). The library was created using MATLAB version 6.5 (Release 13), but it can be ported with some changes to lower MATLAB versions. Controllers are implemented in the library as standalone Simulink blocks, which allow an easy incorporation into existing simulation schemes and an easy creation of new simulation circuits. Only standard techniques of Simulink environment were used when creating the controller blocks and thus just basic knowledge of this environment is required for the start of work with the library. Controllers can be implemented to simulation schemes just by the copy or drag & drop operation and their parameters are set using dialog windows. Another advantage of used approach is a relatively easy implementation of user-defined controllers by modifying some suitable controller in the library.

Nowadays the library contains over 30 simple single input single output discrete self-tuning controllers, which use discrete ARX models of second and third order for the on-line system identification. All of these controllers use discrete control laws, where controller parameters are computed by various methods. Many methods calculate controller parameters on base of the ultimate parameters of controlled system (Ziegler-Nichols approach) or on base of pole placement approach. The library package contains not only the controllers but also reference manual with simple description of the algorithm and the internal structure of each controller.

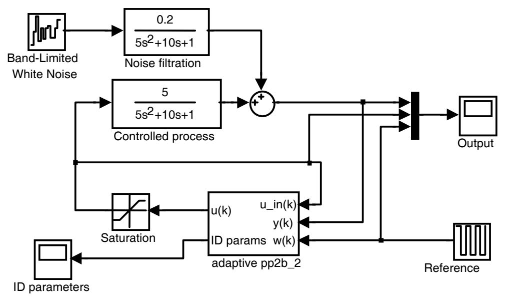


Figure 4: Control circuit in Simulink environment

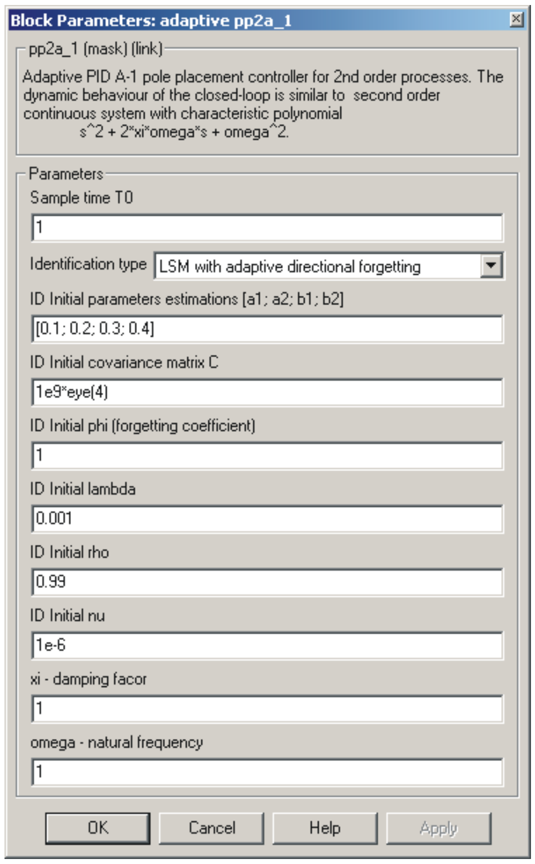
The typical wiring of any library controller is shown in Fig. 4. Each self-tuning controller from the library uses 3 input signals and provides 2 outputs. The inputs are the reference signal (w) and the actual output of controlled process (y). The last controller input is the current input of controlled process – the control signal (u_in). The value of this signal does not have to be the same as controller output e.g. due to the saturation of controller output. The main controller output is, of course, control signal – the input signal of the controlled process. The second controller output consists of the current parameter estimates of the controlled process model. The number of parameters this output consists of depends on the model used by on-line identification.

A scheme analogous to the scheme in Fig. 4 can also be used to simulate both discrete and a continuous-time controlled process with much more complicated structure. It is possible to implement processes with time-varying parameters, processes described by non-linear differential equations etc.

The dynamic behaviour of a controller can be influenced by changing the parameters available for a given controller. Controller parameters can be divided into two groups:

- parameters common to all controllers,
- controller specific parameters.

All controller parameters are set or changed by standard approaches of the Simulink environment, that is, by changing items in the dialogue window - invoked for example, by double-clicking with the mouse on the controller object. This dialogue window also contains a short description of the corresponding controller; and the “Help” button is used to invoke the corresponding part of the hypertext reference guide. The parameter setting dialog window of the *pp2a_1* controller is shown in Fig. 5.


Block Parameters: adaptive pp2a_1

pp2a_1 (mask) (link)

Adaptive PID A-1 pole placement controller for 2nd order processes. The dynamic behaviour of the closed-loop is similar to second order continuous system with characteristic polynomial
s^2 + 2*xi*omega*s + omega^2.

Parameters

Sample time T0: 1

Identification type: LSM with adaptive directional forgetting

ID Initial parameters estimations [a1; a2; b1; b2]: [0.1; 0.2; 0.3; 0.4]

ID Initial covariance matrix C: 1e9*eye(4)

ID Initial phi (forgetting coefficient): 1

ID Initial lambda: 0.001

ID Initial rho: 0.99

ID Initial nu: 1e-6

xi - damping facor: 1

omega - natural frequency: 1

Figure 5: Dialogue for *pp2a_1* Controller Setting Parameters

Each library controller is constructed as a mask of a subsystem, which consists of Simulink blocks and has inputs, outputs and parameters. Internal controller structure consists of Simulink blocks which provide, among others, the possibility of easy creation of a new controller by a modification of some suitable library controller. The structure of controller *pp2a_1* is presented in Fig. 6 as an example. Each library controller consists of three basic parts:

- on-line identification block,
- block computing controller parameters,
- block computing controller output.

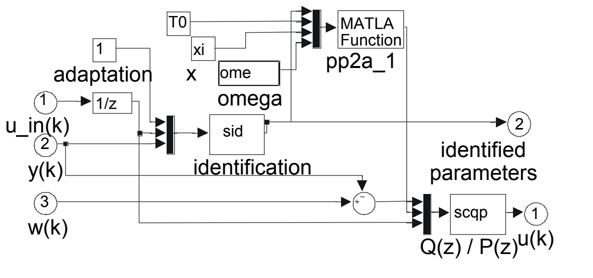


Figure 6: Block scheme of PP2A_1 controller

SIMULATION EXAMPLES IN THE SIMULINK ENVROMENT

Simulation is a useful tool for the synthesis of control systems, allowing one not only to create mathematical models of a process but also to design virtual controllers in a computer. The mathematical models provided are sufficiently close to a real object that simulation can be used to verify the dynamic characteristics of control loops when the structure or parameters of the controller change. The models of the processes may also be excited by various random noise generators which can simulate the stochastic characteristics of the processes noise signals with similar properties to disturbance signals measured in the machinery. Simulation results are valuable for implementation of a chosen controller (control algorithm) under laboratory and industrial conditions. It must be borne in mind, however, that the practical application of a controller verified by simulation cannot be taken as a routine event. Obviously simulation and laboratory conditions can be quite different from those in real plants, and therefore one must verify its practicability with regard to process dynamics and the required standard of control quality (for example maximum allowable overshoot, accuracy, settling time, etc.).

Simulation experiments of the control of some digital process models will be shown. Only higher order models which have been approximated by second- or third-order models in the identification procedure have been chosen for simulation verification. Simulation verification has been limited to digital PID controllers based on the Ziegler-Nichols and the pole assignment methods, which have proved to be the best in practical applications. A scheme of the control circuit used for simulation in MATLAB-Simulink environment is shown in Fig. 7.

As an example of verification by computer simulation a fourth-order system with the transfer function

$$G(s) = \frac{1}{(s+1)^4} \tag{12}$$

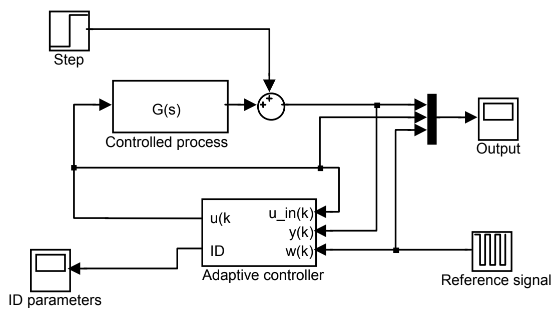


Figure 7: Control circuit used for simulation

was used. The simulation verification was realized using four types of the controllers included in the Self-tuning Controllers Simulink Library (Bobál and Chalupa 2002). In time interval $t = 150 - 200$ s a constant disturbance $v(k) = 1$ was injected on the system output. The controller output value $u(k)$ was limited within the range $\langle 0; 2 \rangle$. The initial value of the directional forgetting factor was chosen as $\varphi(0) = 1$. The initial values of the model parameter estimates are $\hat{\boldsymbol{\Theta}}^T(0) = [0.1, 0.2, 0.3, 0.4]$ for a second-order model, and $\hat{\boldsymbol{\Theta}}^T(0) = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$ for a third-order model.

The control loop was simulated using a PID Takahashi controller (*zn2tak*) – see (Bobál and Chalupa, 2002, Bobál *et al.*, 2005) with equation

$$\begin{aligned} u(k) = K_P\{&-y(k) + y(k-1) + \frac{T_0}{T_I}\left[w(k) - y(k)\right] + \\ &+\left[2y(k-1) - y(k) - (k-2)\right]\} + u(k-1) \end{aligned} \tag{13}$$

where

$$K_P = 0.6K_{Pu}\left(1 - \frac{T_0}{T_u}\right); \quad T_I = \frac{K_P T_u}{1.2K_{Pu}}; \quad T_D = \frac{3K_{Pu}T_u}{40K_P} \tag{14}$$

The sampling period was chosen T_0 = 1 s, critical parameters (proportional gain K_{Pu} and period of oscillations T_u) were computed for the identification of a second-order model. Fig. 8 illustrates the simulation control performance using controller (13), (14).

Fig. 9 illustrates the simulation control performance using controller (13), (14) (*zn3tak*) – see (Bobál and Chalupa, 2002, Bobál *et al.*, 2005), the sampling period

was chosen T_0 = 1.5 s. Critical parameters were computed for the identification of a third-order model.
Fig. 10 illustrates the simulation control performance using pole placement controller *pp2a_1* – see (Bobál and Chalupa, 2003, Bobál *et al*., 2005). The idea of this controller is to make the dynamic behaviour of the closed loop similar to that of the second order continuous system with characteristic polynomial (8) as stated by equation

$$D(s) = s^2 + 2\xi\omega_n s + \omega_n^2 \qquad (15)$$

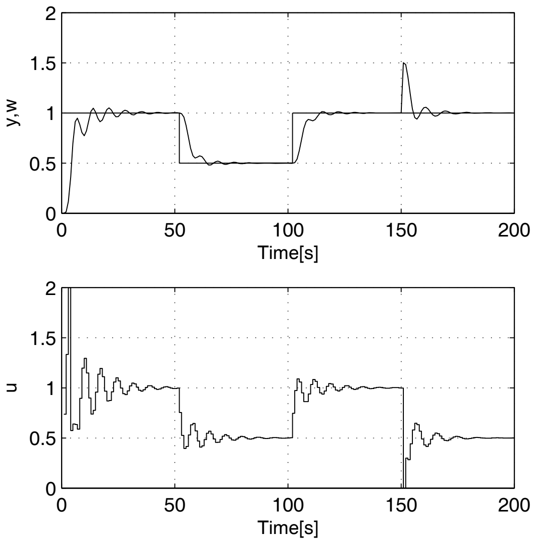


Figure 8: Simulation control performance using a PID Takahashi controller (*zn2tak*) for the identification of a second-order model

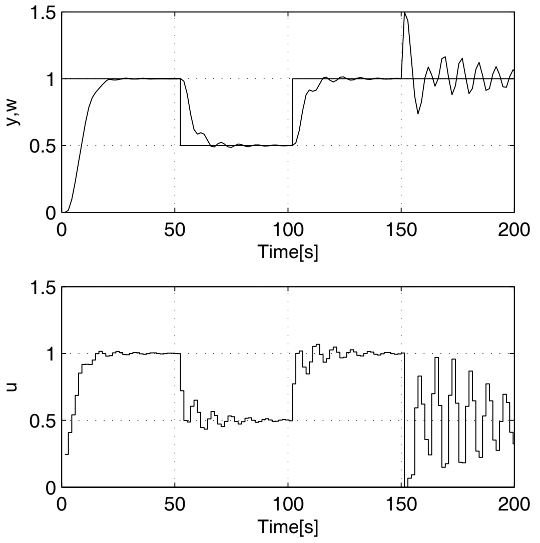


Figure 9: Simulation control performance using a PID Takahashi controller (*zn3tak*) for the identification of a third-order model

Structure of control circuit with this controller is shown in Fig 1and the control law is given by equation

$$u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) + (1-\gamma)u(k-1) + \gamma u(k-2) \qquad (16)$$

where q_0, q_1, q_2 and γ are controller parameters. Additional parameters were set to T_0=2.5 s, ξ=1 and ω_n=0.3, which leads to asymptotic step responses.

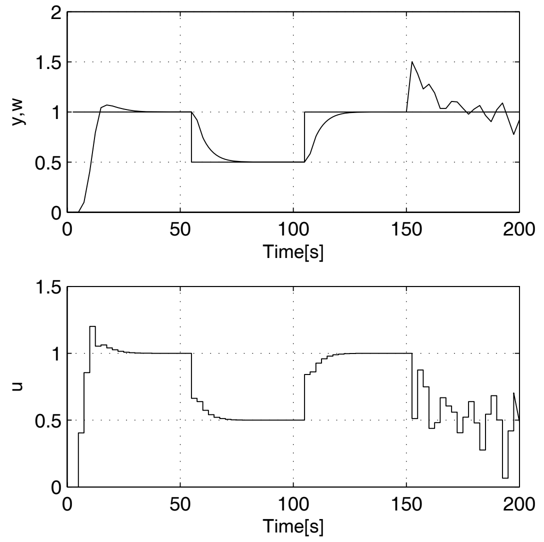


Figure: 10 Simulation control performance using a pole placement controller PID-1 (*pp2a_1*)

From Fig. 8 – 10 it is evident that the response of the controlled variable to step change of the reference signal in the control loop with the *pp2a_1* controller is aperiodical. In the control loop with the controller *zn3tak* occur negligible oscillations of particular variables. The controller *zn2tak* causes light stable oscillations in the closed loop. On the contrary, the best ability to eliminate step disturbances proved the controller *zn2tak.* In control loops with the controllers *zn3tak* and *pp2a_1* step disturbances cause oscillations of the variables.

CREATING APPLICATIONS WITH REAL - TIME WORKSHOP

The MATLAB/Simulink environment can also be used to generate code to be used in controllers in industrial practice. *Real-Time Workshop*, one of the toolboxes shipped with MATLAB, allows generating of source code and programs to be used outside the MATLAB environment. The process of generating the source code is controlled by special compiler files that are interpreted by *Target Language Compiler*. These files are identified by the *.tlc* (target language compiler) extension and describe how to convert Simulink schemes to target language. Thereby, source code is generated and after compiling and linking, the resulting application is created. Applications for various microprocessors and operating systems can be created by selecting corresponding *.tlc* files.

The *Target Language Compiler* can create applications to be used under the Windows environment, which perform control algorithm and save the results in a binary file with a structure acceptable to the MATLAB. An analysis of the control process can then be performed using the advantages of MATLAB functions and commands. Selecting another *.tlc* file leads to creation of an MS-DOS application, or an application to be used on PC - based industrial computers without the requirement for an operating system.

Many manufactures of industrial computers and controllers have created their own target language compiler files used to create applications for the equipment they produce. *Real-Time Workshop* provides a relatively open environment for the conversion of block schemes to various platforms, where any users can create their own *.tlc* files for converting the block scheme to a source code and hence, achieve compatibility with any hardware.

Application creation only requires selection of the appropriate *.tlc* file, or eventually, setting the compiler parameters and then initiation of the compilation process.

CONCLUSIONS

The Self-tuning Controllers Simulink Library is used in university course of Adaptive Control Systems. Its architecture enables an easy user orientation in Simulink block schemes and generates source code for controllers' functions. The controllers provided are suitable for modification and thereby implementation of user-defined controllers. The compatibility with *Real-Time Workshop* ensures not only the possibility of laboratory testing using real time models but also the possibility of creating applications for industrial controllers.

ACKNOWLEDGEMENTS

This work was supported in part by the Ministry of Education of the Czech Republic under grants MSM 7088352101, 1M0567 and by the Grant Agency of the Czech Republic under grant 102/06/1132.

REFERENCES

Bobál, V., J. Böhm, J. Fessl and J. Macháček. 2005. *Practical Aspects of Self-tuning Controllers: Algorithms,Implementation and Applications*. Springer-Verlag, London.

Bobál, V., J. Böhm, and R. Prokop. 1999. "Practical Aspects of Self-tuning Controllers". *International Journal of Adaptive Control and Signal Processing*, 13, 671-690.

Bobál, V. and P. Chalupa. 2002. "Self-tuning Controllers Simulink Library". http: //www.utb.cz/stctool/.

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

Kučera, V. 1991. *Analysis and Design of Discrete Linear Control Systems*. Prentice Hall, London.

Kulhavý, R. 1987. "Restricted Exponential Forgetting in Real Time Identification". *Automatica*, 23**,** 586-600.

Ziegler, J. G. and N. B. Nichols. 1942. "Optimum Settings for Automatic Controllers". *Trans. ASME*, 64, 759 – 768.

AUTHOR BIOGRAPHIES

VLADIMÍR BOBÁL was born in Slavičín, Czech Republic in 1942. He graduated in 1966 from the Brno University of Technology. He received his Ph.D. degree in Technical Cybernetics at Institute of Technical Cybernetics, Slovak Academy of Sciences, Bratislava, Slovak Republic. He is now Professor in the Department of Process Control, Faculty of Applied Informatics of the Tomas Bata University in Zlín. His research interests are adaptive control systems, system identification and CAD for self-tuning controllers. You can contact him on email address `bobal@fai.utb.cz`

PETR CHALUPA was born in Zlin, Czech Republic in 1976. He graduated from Brno University of Technology in 1999. He obtained his Ph.D. in Technical Cybernetics at Tomas Bata University in Zlin in 2003. Nowadays he works as a researcher at Centre of Applied Cybernetics at Tomas Bata University in Zlin. His research interests are adaptive and predictive control of real-time systems. You can contact him on email address `chalupa@fai.utb.cz`

PETR DOSTÁL was born in Kněždub, Czech Republic in 1945. He studied at the Technical University of Pardubice, where he obtained his master degree in 1968 and PhD. degree in Technical Cybernetics in 1979. In the year 2000 he became professor in Process Control. He is now head of the Department of Process Control, Faculty of Applied Informatics of the Tomas Bata University in Zlín. His research interests are modelling and simulation of continuous-time chemical processes, polynomial methods, optimal and adaptive control. You can contact him on email address `dostalp@fai.utb.cz`