ENVIRONMENT FOR CAD AND VERIFICATION OF SELF – TUNING CONTROLLERS

Marek Kubalcik, Vladimír Bobal, Miroslav Maca

Department of Process Control, Faculty of Applied Informatics,
Tomas Bata University in Zlin, Nad Stranemi 4511, 760 05, Zlín, Czech Republic,
Tel.: +420 57 603 5198, Fax: +420 57 603 5279, E-mail: kubalcik@fai.utb.cz

Abstract: The contribution deals with an adaptive controllers' library designed under MATLAB – Simulink environment. The library enables design and verification of two input – two output (TITO) discrete self – tuning controllers. It is a completely open system based on Simulink features. Design of the TITO controllers that are included into the library is based on polynomial methods. The controllers are based on various configurations of the closed loop system. Several decoupling compensators are also applied. The main idea of self – tuning controllers is discussed. In the paper are presented also several examples. The possibility of developing real – time applications using MATLAB Real – Time Workshop is discussed. *Copyright © 2007 IFAC*

Keywords: Simulation tools, multivariable systems, polynomial methods, discrete self – tuning controllers

1. INTRODUCTION

A tool for computer aided design and verification of TITO discrete self – tuning controllers was created under MATLAB –Simulink environment. It is a library which represents a completely open system based on Simulink features. It has two different forms. The first one was created using the programming language of the MATLAB environment. This version is typically efficient in education. The second version which is compatible with Real Time Workshop, one of the toolboxes shipped with MATLAB, was written with use of C programming language. The main advantage of this version is possibility of generating of source code and programs to be used outside the MATLAB environment. On the contrary, disadvantageous are extensive source codes and consequently worse legibility for a user.

Although the library can be simply extended for systems with various numbers of inputs and outputs, it is primarily focused on TITO systems. It is because there are many examples of industrial processes which have character of TITO systems. Lot of TITO laboratory models which allow simulation of industrial processes and verification of control algorithms are available. For example the coupled drives apparatus (it allows to investigate controlling the tension and speed of material in a continuous process, which is a very common industrial process), the air – heating model, the coupled tanks apparatus, electrical analogous models etc. All these systems require a simultaneous control of two variables related to one system. Each input may influence both system outputs. The design of a controller for such a system must be quite sophisticated to be able to control the system adequately.

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 now application has required a control engineer skilled in specific technology as well as versed in modern control methods. The original aim of the contribution was using of the library for education of multivariable adaptive systems. Compatibility with Real Time Workshop also enables both real – time laboratory verification of designed two – variable controllers and creating industrial applications. The library also enables to present control algorithms in a form acceptable for industrial users.

Design of controllers that are included into the library is based on polynomial methods (Kučera, 1980, Kučera, 1991). Various configurations of the closed loop system were considered. Several decoupling methods using decoupling compensators (Peng, 1990, Krishnawamy *et al.*, 1991) were also applied. All control algorithms were realized as self – tuning controllers based on a recursive identification of a suitable model of the controlled process. The process is identified by the regression model using the recursive least squares method with the directional forgetting (Kulhavý, 1987; Bittanti *et al.*, 1990).

This paper is organized as follows: Section 2 contains description of the controllers implemented into the library; Section 3 describes the system identification method; Section 4 contains description of the library; Section 5 gives an experimental example; Section 6 concludes the paper.

2. IMPLEMENTED CONTROLLERS

The library contains controllers based on 1DOF configuration with a feedback part and traditional 2DOF configuration with both feedback and feedforward parts. Further it contains also controller based on the 2DOF configuration presented in (Ortega and Kelly, 1984) for SISO systems. Several decoupling compensators were also applied. For design of particular controllers were used polynomial methods. This section contains a brief description of models which are the controllers in the library based on and two examples of controller design.

2.1 Model of the Controlled Process

It is possible to assume that the dynamic behaviour of a two input – two output system can be described by a discrete linear model in the form of the matrix fraction

$$\boldsymbol{G}(z)=\frac{\boldsymbol{Y}(z)}{\boldsymbol{U}(z)}=\boldsymbol{A}^{-1}\left(z^{-1}\right)\boldsymbol{B}\left(z^{-1}\right)=\boldsymbol{B}_1\left(z^{-1}\right)\boldsymbol{A}_1^{-1}\left(z^{-1}\right) \quad (1)$$

Where polynomial matrices $\boldsymbol{A}\in R_{22}[z^{-1}]$, $\boldsymbol{B}\in R_{22}[z^{-1}]$ are the left coprime factorization of the transfer matrix $\boldsymbol{G}(z)$ and matrices $\boldsymbol{A_1}\in R_{22}[z^{-1}]$, $\boldsymbol{B_1}\in R_{22}[z^{-1}]$ are the right coprime factorization of $\boldsymbol{G}(z)$. $\boldsymbol{U}(z)$ is a vector of manipulated variables and $\boldsymbol{Y}(z)$ is an output vector. Design of all controllers included into the library is based on a model with polynomials of second order. This model proved to be effective for control of several laboratory processes (Kubalčík and Bobál, 2002; Kubalčík and Bobál, 2002) where controllers based on a model with polynomials of the first order failed. A model with polynomials of higher order would result to sophisticated derivation of controllers' parameters and control law. But the library is an open system and it is possible to add controllers based on models with polynomials of various orders. In case of controllers with decoupling compensators the model was simplified by considering matrix $\boldsymbol{A}(z^{-1})$ as diagonal. When matrix $\boldsymbol{A}(z^{-1})$ is supposed as nondiagonal, it would increase the order of the controller and sophistication of the closed loop system (Kubalčík and Bobál, 2003). The model with nondiagonal matrix has sixteen parameters

$$\boldsymbol{A}\left(z^{-1}\right)=\begin{bmatrix}1+a_1z^{-1}+a_2z^{-2} & a_3z^{-1}+a_4z^{-2}\\ a_5z^{-1}+a_6z^{-2} & 1+a_7z^{-1}+a_8z^{-2}\end{bmatrix} \quad (2)$$

$$\boldsymbol{B}\left(z^{-1}\right)=\begin{bmatrix}b_1z^{-1}+b_2z^{-2} & b_3z^{-1}+b_4z^{-2}\\ b_5z^{-1}+b_6z^{-2} & b_7z^{-1}+b_8z^{-2}\end{bmatrix}$$

The assumption of diagonal matrix $\boldsymbol{A}(z^{-1})$ results in a reduction of the number of parameters. Then the resulting model has twelve parameters.

For derivation of controllers outputs are used parameter estimates of a model of a controlled process. So in the framework of self – tuning controllers it is possible to simulate control of continuous systems or control of discrete systems with a structure which is different from the structure of the model. It is also possible to connect and control various two input – two output laboratory models, as it was mentioned above.

2.2 Design of 1DOF Controller

The 1DOF control configuration is depicted in Fig. 1. It contains only a feedback controller.

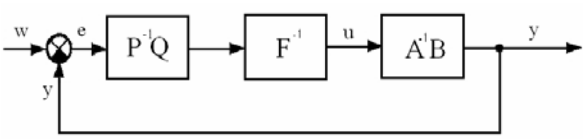

Fig. 1 Block diagram of the 1DOF control configuration

Similarly as it was for the controlled system, the transfer matrix of the controller takes the form of the following matrix fraction

$$\boldsymbol{G}_R(z)=\boldsymbol{P}^{-1}\left(z^{-1}\right)\boldsymbol{Q}\left(z^{-1}\right)=\boldsymbol{Q}_1\left(z^{-1}\right)\boldsymbol{P}_1^{-1}\left(z^{-1}\right) \quad (3)$$

Generally, the vector $\boldsymbol{W}(z^{-1})$ of input reference signals is specified as

$$\boldsymbol{W}(z)=\boldsymbol{F}_w^{-1}\left(z^{-1}\right)\boldsymbol{h}\left(z^{-1}\right) \quad (4)$$

Here, the reference signals are considered as a class of step functions. In this case $\boldsymbol{h}(z^{-1})$ is a vector of constants and $\boldsymbol{F}_w(z^{-1})$ is expressed as

$$\boldsymbol{F}_w\left(z^{-1}\right)=\begin{bmatrix}1-z^{-1} & 0\\ 0 & 1-z^{-1}\end{bmatrix} \quad (5)$$

The compensator $\boldsymbol{F}(z^{-1})$ is a component formally separated from the controller. It has to be included in the controller to fulfil the requirement on the asymptotic tracking. If the reference signals are of the same class as the step functions are, then $\boldsymbol{F}(z^{-1})$ is an integrator.

The control law (operator z^{-1} will be omitted from some operations for the purpose of simplification) is defined as

$$\boldsymbol{U}=\boldsymbol{F}^{-1}\boldsymbol{Q}_1\boldsymbol{P}_1^{-1}\boldsymbol{E} \quad (6)$$

where $\boldsymbol{E}$ is a vector of control errors. Using matrix operations it is possible to modify this vector as

$$\boldsymbol{E} = \boldsymbol{W} - \boldsymbol{Y} = \boldsymbol{P}_1\left(\boldsymbol{AFP}_1 + \boldsymbol{BQ}_1\right)^{-1}\boldsymbol{AW} \tag{7}$$

Asymptotic tracking of the reference signals is then fulfilled if $\boldsymbol{FP}_1$ is divisible by $\boldsymbol{F}_w$.
It is possible to derive the following equation for the system output

$$\boldsymbol{Y} = \boldsymbol{A}^{-1}\boldsymbol{BF}^{-1}\boldsymbol{P}^{-1}\boldsymbol{QE} = \boldsymbol{A}^{-1}\boldsymbol{BF}^{-1}\boldsymbol{P}^{-1}\boldsymbol{Q}\left(\boldsymbol{W} - \boldsymbol{Y}\right) \tag{8}$$

and this can be modified

$$\boldsymbol{Y} = \boldsymbol{P}_1\left(\boldsymbol{AFP}_1 + \boldsymbol{BQ}_1\right)^{-1}\boldsymbol{BQ}_1\boldsymbol{P}_1^{-1}\boldsymbol{W} \tag{9}$$

It is apparent, that the elements of the vector of the output signal have in their denominators the determinant of the matrix $\boldsymbol{AFP}_1$ + $\boldsymbol{BQ}_1$. This determinant is the characteristic polynomial of a MIMO system. The roots of this polynomial matrix are the ruling factors for the behaviour of a closed loop system. The roots must be inside the unit circle (of the Gauss complex plain), in order for the system to be stable. Conditions of BIBO (bounded input bounded output) stability can be defined by the following diophantine equation

$$\boldsymbol{AFP}_1 + \boldsymbol{BQ}_1 = \boldsymbol{M} \tag{10}$$

Where $\boldsymbol{M} \in R_{22}[z^{-1}]$ is a stable diagonal polynomial matrix.

$$\boldsymbol{M}\left(z^{-1}\right) = \begin{bmatrix} 1 + m_1z^{-1} + m_2z^{-2} + \\ + m_3z^{-3} + m_4z^{-4} & 0 \\ 0 & 1 + m_1z^{-1} + m_2z^{-2} + \\ & + m_3z^{-3} + m_4z^{-4} \end{bmatrix} \tag{11}$$

The degree of the controller polynomial matrices depends on the internal properness of the closed loop. The structure of matrices $\boldsymbol{P}_1$ and $\boldsymbol{Q}_1$ was chosen so that the number of unknown controller parameters equals the number of algebraic equations resulting from the solution of the diophantine equation. The method of the uncertain coefficient method was used to solve the diophantine equation.

$$\boldsymbol{P}_1\left(z^{-1}\right) = \begin{bmatrix} 1 + p_1z^{-1} & p_2z^{-1} \\ p_3z^{-1} & 1 + p_4z^{-1} \end{bmatrix} \tag{12}$$

$$\boldsymbol{Q}_1\left(z^{-1}\right) = \begin{bmatrix} q_1 + q_2z^{-1} + q_3z^{-2} & q_4 + q_5z^{-1} + q_6z^{-2} \\ q_7 + q_8z^{-1} + q_9z^{-2} & q_{10} + q_{11}z^{-1} + q_{12}z^{-2} \end{bmatrix}$$

The solution to the diophantine equation results in a set of sixteen algebraic equations with unknown controller parameters. Using matrix notation the algebraic equations can be expressed in the following form

$$\begin{bmatrix} -a_2 & -a_4 & 0 & 0 & b_2 & 0 & 0 & b_4 \\ a_2 - a_1 & a_4 - a_3 & 0 & b_2 & b_1 & 0 & b_4 & b_3 \\ a_1 - 1 & a_3 & b_2 & b_1 & 0 & b_4 & b_3 & 0 \\ 1 & 0 & b_1 & 0 & 0 & b_3 & 0 & 0 \\ -a_6 & -a_8 & 0 & 0 & b_6 & 0 & 0 & b_8 \\ a_6 - a_5 & a_8 - a_7 & 0 & b_6 & b_5 & 0 & b_8 & b_7 \\ a_5 & a_7 - 1 & b_6 & b_5 & 0 & b_8 & b_7 & 0 \\ 0 & 1 & b_5 & 0 & 0 & b_7 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_3 \\ q_1 \\ q_2 \\ q_3 \\ q_7 \\ q_8 \\ q_9 \end{bmatrix} = \begin{bmatrix} m_4 \\ m_3 + a_2 \\ m_2 - a_2 + a_1 \\ m_1 - a_1 + 1 \\ 0 \\ a_6 \\ a_5 - a_6 \\ -a_5 \end{bmatrix}$$

$$\begin{bmatrix} -a_2 & -a_4 & 0 & 0 & b_2 & 0 & 0 & b_4 \\ a_2 - a_1 & a_4 - a_3 & 0 & b_2 & b_1 & 0 & b_4 & b_3 \\ a_1 - 1 & a_3 & b_2 & b_1 & 0 & b_4 & b_3 & 0 \\ 1 & 0 & b_1 & 0 & 0 & b_3 & 0 & 0 \\ -a_6 & -a_8 & 0 & 0 & b_6 & 0 & 0 & b_8 \\ a_6 - a_5 & a_8 - a_7 & 0 & b_6 & b_5 & 0 & b_8 & b_7 \\ a_5 & a_7 - 1 & b_6 & b_5 & 0 & b_8 & b_7 & 0 \\ 0 & 1 & b_5 & 0 & 0 & b_7 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_2 \\ p_4 \\ q_5 \\ q_6 \\ q_7 \\ q_{10} \\ q_{11} \\ q_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ a_4 \\ a_3 - a_4 \\ -a_3 \\ m_4 \\ m_3 + m_4 \\ m_2 - a_4 + a_7 \\ m_1 - a_3 + 1 \end{bmatrix} \tag{13}$$

The controller parameters are obtained by solving these equations.

2.2 Design of 2DOF Controller with Two Feedbacks

The structure of the closed loop is shown in Fig. 2.

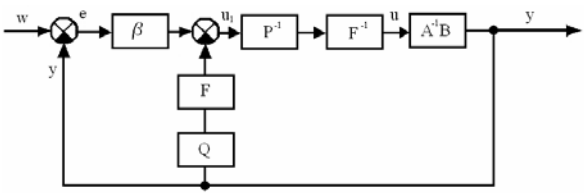

Fig. 2 Block diagram of the 2DOF control configuration

It is possible to derive the following equation for the system output

$$\boldsymbol{Y} = \boldsymbol{A}^{-1}\boldsymbol{BU} = \boldsymbol{A}^{-1}\boldsymbol{BF}^{-1}\boldsymbol{P}^{-1}\boldsymbol{U}_1 \tag{14}$$

where

$$\boldsymbol{U}_1 = \boldsymbol{\beta}\left(\boldsymbol{W} - \boldsymbol{Y}\right) - \boldsymbol{QFY} \tag{15}$$

The corresponding equation for the controller's output, as shown in the block diagram in Fig. 2, follows as

$$\boldsymbol{U} = \boldsymbol{F}^{-1}\boldsymbol{P}^{-1}\boldsymbol{U}_1 \tag{16}$$

The substitution of $\boldsymbol{U}_1$ and $\boldsymbol{Y}$ results in

$$\boldsymbol{U} = \boldsymbol{F}^{-1}\boldsymbol{P}^{-1}\left[\boldsymbol{\beta}\left(\boldsymbol{W} - \boldsymbol{A}^{-1}\boldsymbol{BU}\right) - \boldsymbol{QFA}^{-1}\boldsymbol{BU}\right] \tag{17}$$

The equation (17) can be modified using the right matrix fraction of the controlled system into the form

$$\boldsymbol{U} = \boldsymbol{A}_1\left[\boldsymbol{PFA}_1 + \left(\boldsymbol{\beta} + \boldsymbol{FQ}\right)\boldsymbol{B}_1\right]\boldsymbol{\beta W} \tag{18}$$

The closed loop system is stable when the following diophantine equation is satisfied

$$\boldsymbol{PFA}_1 + \left(\boldsymbol{\beta} + \boldsymbol{FQ}\right)\boldsymbol{B}_1 = \boldsymbol{M} \tag{19}$$

Where $\boldsymbol{M}$ is defined by (11) and the structure of the matrices $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{\beta}$ was chosen as follows

$$\boldsymbol{P}\left(z^{-1}\right) = \begin{bmatrix} 1 + p_1z^{-1} & p_2z^{-1} \\ p_3z^{-1} & 1 + p_4z^{-1} \end{bmatrix}$$
$$\boldsymbol{Q}\left(z^{-1}\right) = \begin{bmatrix} q_1 + q_2z^{-1} & q_3 + q_4z^{-1} \\ q_5 + q_6z^{-1} & q_7 + q_8z^{-1} \end{bmatrix} \tag{20}$$
$$\boldsymbol{\beta}\left(z^{-1}\right) = \begin{bmatrix} \beta_1 & \beta_2 \\ \beta_3 & \beta_4 \end{bmatrix}$$

The solution of the diophantine equation results in a set of sixteen algebraic equations with unknown controller parameters. Using matrix notation the algebraic equations are expressed in the following form

$$\begin{bmatrix} 1 & 0 & b_9 & 0 & b_{13} & 0 & b_9 & b_{13} \\ a_9-1 & a_{13} & b_{10}-b_9 & b_9 & b_{14}-b_{13} & b_{13} & b_{10} & b_{14} \\ a_{10}-a_9 & a_{14}-a_{13} & -b_{10} & b_{10}-b_9 & -b_{14} & b_{14}-b_{13} & 0 & 0 \\ -a_{10} & -a_{14} & 0 & -b_{10} & 0 & -b_{14} & 0 & 0 \\ 0 & 1 & b_{11} & 0 & b_{15} & 0 & b_{11} & b_{15} \\ a_{11} & a_{15}-1 & b_{12}-b_{11} & b_{11} & b_{16}-b_{15} & b_{15} & b_{12} & b_{16} \\ a_{12}-a_{11} & a_{16}-a_{15} & -b_{12} & b_{12}-b_{11} & -b_{16} & b_{16}-b_{15} & 0 & 0 \\ -a_{12} & -a_{16} & 0 & -b_{12} & 0 & -b_{16} & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ q_1 \\ q_2 \\ q_3 \\ q_4 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} m_1-a_9+1 \\ m_2+a_9-a_{10} \\ m_3+a_{10} \\ m_4 \\ -a_{11} \\ a_{11}-a_{12} \\ a_{12} \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & b_9 & 0 & b_{13} & 0 & b_9 & b_{13} \\ a_9-1 & a_{13} & b_{10}-b_9 & b_9 & b_{14}-b_{13} & b_{13} & b_{10} & b_{14} \\ a_{10}-a_9 & a_{14}-a_{13} & -b_{10} & b_{10}-b_9 & -b_{14} & b_{14}-b_{13} & 0 & 0 \\ -a_{10} & -a_{14} & 0 & -b_{10} & 0 & -b_{14} & 0 & 0 \\ 0 & 1 & b_{11} & 0 & b_{15} & 0 & b_{11} & b_{15} \\ a_{11} & a_{15}-1 & b_{12}-b_{11} & b_{11} & b_{16}-b_{15} & b_{15} & b_{12} & b_{16} \\ a_{12}-a_{11} & a_{16}-a_{15} & -b_{12} & b_{12}-b_{11} & -b_{16} & b_{16}-b_{15} & 0 & 0 \\ -a_{12} & -a_{16} & 0 & -b_{12} & 0 & -b_{16} & 0 & 0 \end{bmatrix} \begin{bmatrix} p_3 \\ p_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} -a_{13} \\ a_{13}-a_{14} \\ a_{14} \\ 0 \\ m_5-a_{15}+1 \\ m_6+a_{15}-a_{16} \\ m_7+a_{16} \\ m_8 \end{bmatrix} \tag{21}$$

The controller parameters are derived by solving these equations. The control law apparent from the block diagram is defined as

$$\boldsymbol{FPU} = \boldsymbol{\beta E} - \boldsymbol{FQY} \tag{22}$$

3. RECURSIVE IDENTIFICATION

In the identification part of the controller algorithms the discrete regression model (ARX) was used. Using matrix notation it can be specified as follows:

$$\begin{aligned} y_1(k) &= \boldsymbol{\Theta}_1^T(k)\phi_1(k-1)+n_1(k) \\ y_2(k) &= \boldsymbol{\Theta}_2^T(k)\phi_2(k-1)+n_2(k) \end{aligned} \tag{23}$$

where

$$\begin{aligned} \boldsymbol{\Theta}_1^T(k) &= [a_1,a_2,a_3,a_4,b_1,b_2,b_3,b_4] \\ \boldsymbol{\Theta}_2^T(k) &= [a_5,a_6,a_7,a_8,b_5,b_6,b_7,b_8] \end{aligned} \tag{24}$$

are the vectors of parameter estimates and

$$\begin{aligned} \phi_{1,2}^T(k-1) = [&-y_1(k-1),-y_1(k-2),-y_2(k-1), \\ &-y_2(k-2),u_1(k-1),u_1(k-2),u_2(k-1),u_2(k-2)] \end{aligned} \tag{25}$$

is the regression vector ($y(k)$ is the process output variable, $u(k)$ is the controller output variable). The components $n_1(k)$, $n_2(k)$ are non - measurable random signals, which are assumed to have zero mean value and constant covariance.

The parameter estimates are calculated using the recursive least square method, where adaptation is supported by directional forgetting (Kulhavý, 1987; Bittanti *et al.*, 1990).

4. DESCRIPTION OF THE LIBRARY

The library was created and tested using MATLAB version 6.0 (Release 12), but it can be ported with some changes to lower MATLAB versions. The controllers are implemented as encapsulated Simulink blocks and that enables their simple incorporation into existing Simulink 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. The controllers can be implemented into simulation schemes just by the copy or drag & drop operation and their parameters are set using dialog windows. Another advantage of the used approach is a relatively easy implementation of user – defined controllers by modifying some suitable controller in the library. The library contains not only the controllers but also blocks of controlled processes models. Both continuous and discrete models of various orders are available. So it is possible to combine various controller types with various controlled processes models. It is also possible to implement processes with time variable parameters or processes described by non – linear differential equations.

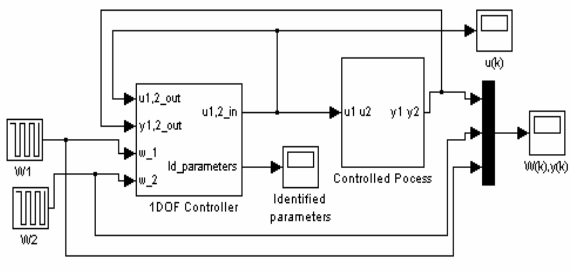

Fig. 3 SIMULINK scheme of closed loop system

The typical wiring of any library controller is shown in Fig. 3.

Each self – tuning controller from the library uses four input signals and provides two outputs. The inputs are the reference signals (w$_1$, w$_2$) and the vector of actual outputs of controlled process (y). The last controller input is the vector of the current input of controlled process (u_in). The main controller output is, of course, the vector of control signals – the input signals 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.

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 1DOF controller is presented in Fig. 4 as an example.

Each library controller consists of three basic parts: on – line identification block, block computing controller parameters, block computing controller outputs. Controller parameters are set using dialog windows.

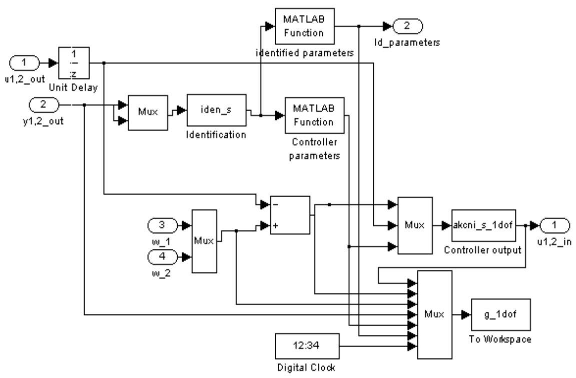

Fig. 4 Block scheme of 1DOF controller

It is possible to set poles of the right side matrix ***M*** (pole assignment), sampling period and parameters of the recursive identification algorithm.

From the programmer point of view, standalone programme files, which are interpreted after simulation start by the MATLAB environment, correspond to particular parts of the controller. For this purpose Simulink blocks of Matlab Fcn and S – Function were used. Matlab Fcn block just converts an input vector to an output vector. S – Function block further enables to put both discrete and continuous states and consequently perform with particular sampling period. The algorithm of the recursive identification is realized by S – Function, derivation of the controller parameters is realized by Matlab Fcn. Computation of controller output is performed by S – function.

In the basic version of the library are all functions programmed in the programming language of the MATLAB environment as m – files. This version can be also used for real – time control of some laboratory model. The connection of the laboratory model and the Simulink environment was realized through control and measurement PC card Advantech PCL – 812 PG. Blocks of Real Time Toolbox (different toolbox from Real Time Workshop) for reading analogue inputs and for writing to analogue outputs on the PC card were used to communicate with the model.

The MATLAB – Simulink environment can also be used to generate code to be used in controllers in industrial practice. Real Time Workshop 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 describe how to convert Simulink schemes to target language. Applications for various microprocessors and operating systems can be created by selecting corresponding target language compiler files. Many manufactures of industrial computers and controllers has created their own target language compiler files used to create applications for equipment they produce. Real Time Workshop provides a relatively opened environment for the conversion of block schemes to various platforms where users can create their own target language compiler files for converting the block scheme to a source code and hence reach the compatibility with any hardware.

The version of the library which is compatible with Real Time Workshop was written in C programming language. Unfortunately, Real Time Workshop does not support all blocks of the Simulink environment. The Matlab Fcn block is not supported at all and the S – Function block is supported only in case when it is written in C language. To ensure full compatibility with Real Time Workshop, the basic version of the library was revised. Performance and internal structure of the controllers in both versions are the same. Different are tools which were used for programming of the controllers. Matlab Fcn blocks, which are used for derivation of controllers' parameters, were replaced by S – Functions. These S – Functions simply convert their inputs (current parameter estimates) to outputs (controllers' parameters). Nor discrete neither continuous states are used. Blocks performing system identification and derivation of controllers' outputs were transformed from the language of the MATLAB environment to C language.

5. EXPERIMENTAL EXAMPLE

The real – time control of an experimental laboratory model CE 108 – the coupled drives apparatus is given here as an example. This apparatus is based on experience with authentic industrial control applications. It allowed us to investigate the ever-present difficulty of controlling the tension and speed of material in a continuous process. Process like this may require the material speed and tension to be controlled within the defined limits. Practical examples are in the paper-making industry, strip metal and wire manufacturing and, indeed, any process where the product is manufactured in the form of a continuous strip. The principle scheme of the model is shown in Fig. 5. In the apparatus, the produced material strip is replaced by a continuous flexible belt. The apparatus consists of three pulleys mounted on a vertical panel to form a triangle resting on its base. The two base pulleys are directly mounted on the shafts of two nominally identical servo motors and the apparatus is controlled by manipulating the drive torques to these servo motors. The third pulley, the jockey, is free to rotate and it is mounted on a pivoted arm. The jockey pulley assembly, which simulates a material work station, is equipped with a special speed sensor and tension measuring equipment. The jockey pulley speed and the belt tension are the principle system outputs. The belt tension is measured indirectly by monitoring the angular deflection of the pivoted tension arm to which the jockey pulley is attached.

The manipulated variables are the inputs to the servo motors and the controlled variables are the belt tension and speed at the work station.

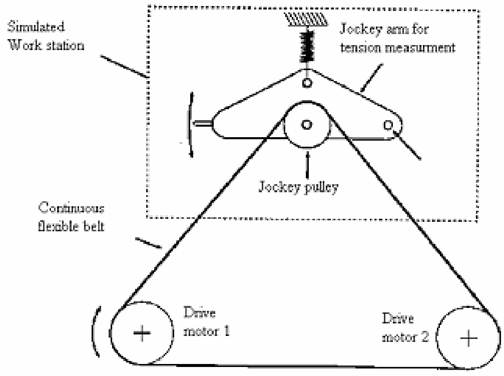

Fig. 5 Principal scheme of CE 108

The coupled drives apparatus is a nonlinear system with variable parameters which is, therefore, difficult to control deterministically. The nonlinear dynamics was described by the linear model (1) in the neighbourhood of a steady state. Adaptive control was performed using the 2DOF controller with two feedbacks.

The right side matrix ***M***, obtained from a number of experiments, was

$$\boldsymbol{M}\left(z^{-1}\right)=\begin{bmatrix} 1-0{,}9z^{-1}+0{,}19z^{-2}- & \\ -0{,}009z^{-3}-0{,}002z^{-4} & 0 \\ 0 & 1-0{,}9z^{-1}+0{,}19z^{-2}- \\ & -0{,}009z^{-3}-0{,}002z^{-4} \end{bmatrix} \tag{26}$$

The initial parameter estimates were chosen without any a priori information

$$\boldsymbol{\Theta}_1^T(0)=[0.1,0.2,0.3,0.4,0.1,0.2,0.3,0.4] \tag{27}$$
$$\boldsymbol{\Theta}_2^T(0)=[0.5,0.6,0.7,0.8,0.5,0.6,0.7,0.8]$$

The sampling period was chosen T_0=0,25 *s*.
The time responses of the control are shown in Fig. 6 and Fig. 7. The controlled variable y_1 is the speed and the controlled variable y_2 is the tension.

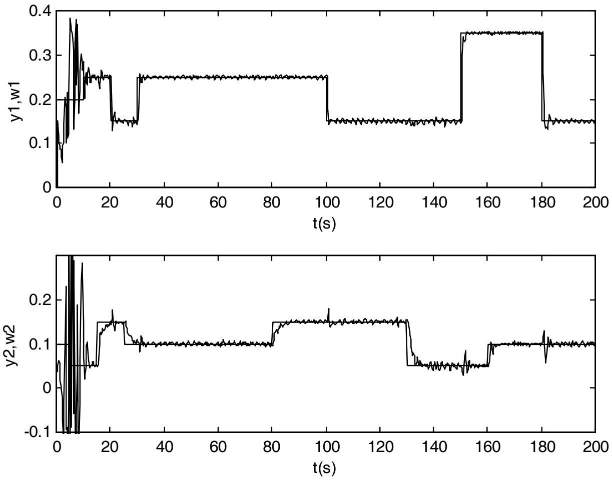

Fig. 6 Adaptive control of the laboratory model

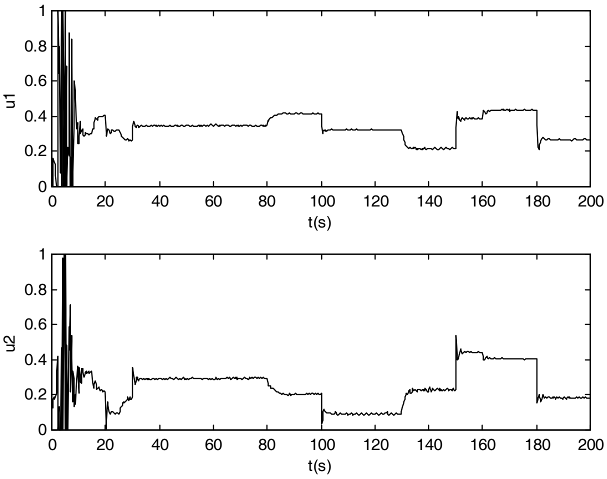

Fig. 7 Adaptive control of the laboratory model – controller output

6. CONCLUSIONS

Architecture of the self – tuning controllers Simulink library enables an easy user orientation in Simulink block schemes and source code of 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 Grant Agency of the Czech Republic under grant No.102/05/0271 and in part by the Ministry of Education of the Czech Republic under grant MSM 7088352101.

REFERENCES

Bittanti, S., Bolzern, P., Campi, M. (1990). Convergence and Exponential Convergence of Identification Algorithms with Directional Forgetting Factor. *Automatica*, 26, 5, pp. 929 – 932.

Krishnawamy, P.R. et al. (1991). Reference System Decoupling for Multivariable Control. *Ind. Eng. Chem. Res.*, 30, pp. 662-670.

Kubalčík, M. and Bobál, V. (2002). Polynomial Design of Controllers for Two – Variable Systems – Practical Implementation. *15th Triennial IFAC World Congress*, Barcelona, Spain, Paper No. 210.

Kubalčík, M. and Bobál, V. (2002). Adaptive Control of Coupled Drives Apparatus Based on Polynomial Theory. *IEEE International Conference on Control Applications*, Glasgow, Scotland, 594 - 599.

Kubalčík, M. and Bobál, V. (2003). Application of Controllers Based on Polynomial Methods to Coupled Drives Laboratory Process. *IEEE Mediterranean Conference Control and Automation*, Rhodos, Greece, IV05 - 03.

Kučera, V. (1980). Stochastic Multivariable Control: a Polynomial approach. *IEEE Trans. of Automatic Control* , 5, pp. 913 – 919.

Kučera, V. (1991). *Analysis and Design of Discrete Linear Control Systems*. Academia, Prague.

Kulhavý, R. (1987). Restricted exponencial forgetting in real – time identification. *Automatica*, 23, 589 – 600.

Ortega, V., and Kelly R.: (1984). PID self – tuners: Some theoretical and practical aspects. *IEEE Trans. Ind. Electron.*, IE - 3, 332 – 338.

Peng, Y. (1990). A General Decoupling Precompensator for Linear Multivariable Systems with Application to Adaptive Control. *IEEE Trans. Aut. Control*, AC-35, 3, pp. 344-348.