

DECENTRALIZED CONTROL OF MULTIVARIABLE SYSTEMS USING REAL-TIME WORKSHOP

Petr Chalupa, Vladimír Bobál

Department of Control Theory, Institute of Information Technologies,
Tomas Bata University in Zlin,

Abstrakt: Příspěvek se zabývá regulací reálných laboratorních mnohazměrových systémů o stejném počtu řídicích a řízených signálů. Pro regulaci je použit decentralizovaný přístup, kdy je mnohazměrový systém rozdělen na jednorozměrové vzájemně se ovlivňující subsystemy a ty jsou řízeny pomocí adaptivních regulátorů. Použité regulátory a regulační obvody jsou vytvořeny v prostředí Simulink. Řízení v reálném čase je prováděno pomocí systému dSpace DS1104, který používá procesor PowerPC. Převod simulinkových schémat na platformu PowerPC je realizován pomocí Real-Time Workshopu.

Abstract: The contribution deals with the control of laboratory multivariable systems with the same number of control signals as the number of controlled signals. The decentralized approach is used to control this type of system, where the multivariable system is divided into single input single output (SISO) interconnected subsystems. Subsequently, these subsystems are controlled by SISO adaptive controllers. Used controllers and control circuits are designed in Simulink environment and real-time control is carried out by system dSpace DS1104, which uses PowerPC processor architecture. Conversion of Simulink schemas into PowerPc platform is performed by Real-Time Workshop.

Keywords: MATLAB, Simulink, ARX model, adaptive control, self-tuning controllers, decentralized control, Real-Time Workshop.

1 DECENTRALIZED CONTROL

Decentralized control is one of the approaches used to control a multi input multi output (MIMO) system. The basic ideas of decentralized control are decomposition of controlled multivariable system to subsystems and then design a controller for each subsystem [1]. Such controllers are of simpler structure than a central multivariable controller is.

In the simplest case the multivariable system is decomposed to SISO subsystems and the design of controllers can be based on theory used for SISO controllers. The main difference between designing a controller for SISO system and designing a controller for SISO subsystem of MIMO system is in non-negligible effects of subsystem interconnections. These interconnection effects acts as disturbances from the point of view of subsystems closed loops.

One of possible approaches of handling this problem is usage of adaptive controllers, which can tune themselves according to changes in the closed loop. Self-tuning SISO controllers as a subset of adaptive controllers were selected to be studied in this work, which focuses on decentralized control of MIMO system with the same number of inputs and outputs. The simplest case of control circuit of two inputs two outputs (TITO) system is shown in Fig 1.

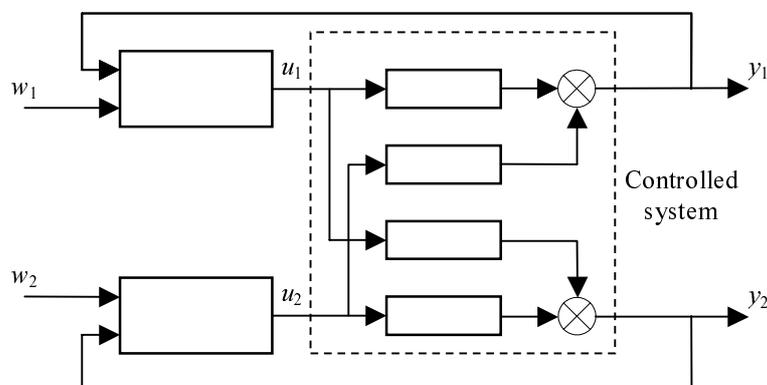


Fig 1. Decentralized control of TITO system

2 SELF-TUNING CONTROLLERS

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 on base of the characteristics of control process. Self-tuning controllers use the combination of the on-line process identification on base of a selected model of the process and the controller synthesis based on knowledge of parameters of controlled process.

The goal of self-tuning controllers is to fulfil the following tasks:

- automatic tuning of digital or analogue controller,
- increase the quality of control process in cases of presence of unsteady disturbances,
- capture changes of controlled process parameters, which can be caused by various technological sources, e.g. the operating of the controlled equipment in the different modes,
- subsequent increase of the quality of the control process by adjustment of digital controller parameters.

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. New sample of controlled variable (and eventually measured disturbances) is obtained and the parameters of controlled system model are recomputed using a recursive identification algorithm [2], [3].
2. The system parameters are assumed be equal to their current estimations. The control strategy for selected criterion of control quality is designed on base of these estimations and controller output is calculated.

3 SELF-TUNING CONTROLLERS SIMULINK LIBRARY

On base on monograph [4] was created a library of self-tuning controllers in Matlab Simulink environment [5]. The purpose was to create an environment suitable for creating and testing of self-tuning controllers. The library was created using Matlab version 6.1 and is available free of charge at internet site of Tomas Bata University in Zlin – www.utb.cz/stctool [6]. 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. The controller structure was designed concerning the usage of controllers in decentralized control systems.

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 [7]) 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.

Each self-tuning controller from the library uses three input signals and provides two outputs. The inputs are the reference signal (w) and the current output of the controlled system (y). The previous controlled system input is the last input of the controller (u_{in}). The value of this signal does not have to be the same as previous 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 system. The second controller output consists of the current parameter estimations of the controlled system model. The number of parameters this output consists of depends on the model used by on-line identification: the controllers using second order model provide four values (a_1 , a_2 , b_1 and b_2) and the controllers using third order model provide six values (a_1 , a_2 , a_3 , b_1 , b_2 and b_3).

Each library controller is constructed as a mask of a subsystem, which consists of Simulink blocks and has inputs, outputs and parameters. As stated in the previous sections, each controller uses three input signals (reference signal w , controlled signal y and control signal u_{in}) and provides two outputs (control signal u and current estimations of model 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. 2 as an example.

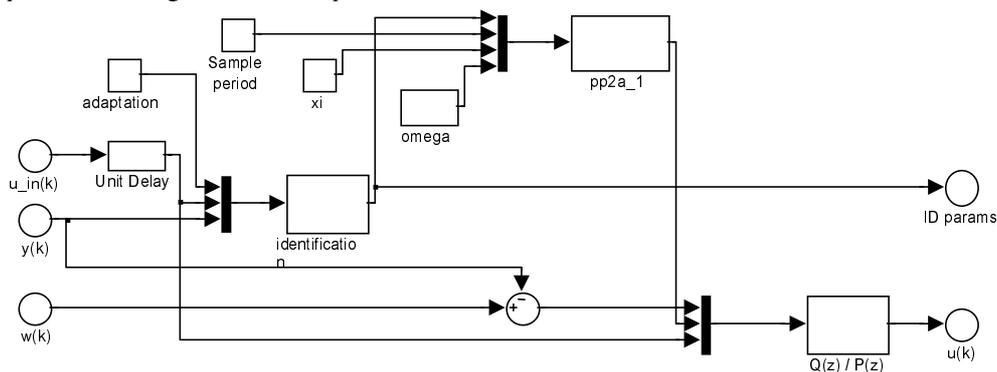


Fig. 2 Block scheme of *pp2a_1* controller

The controllers are suitable for usage in decentralized control circuit with supervisory logic because the on-line identification can be switched off by changing the adaptation constant. If the supervisory logic is present in the system, the adaptation signal should be promoted outside the controller and should act as the fourth controller input. Each controller contained in the Self-Tuning Controllers Simulink Library consists of three basic parts:

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

From the programmer point of view, each block corresponds to the stand-alone program file, which is after the simulation start interpreted by the MATLAB environment. The blocks are implemented as a *MATLAB Fcn* and an *S-Function*. The *MATLAB Fcn* block performs the call of a MATLAB function, which converts input data vector to the output data vector. The *S-Function* block is more universal and corresponding function can use also discrete and continuous states and thus work with some sample time.

4 REAL-TIME WORKSHOP

The Simulink schema which is usually created for simulation testing is selected controller can serve as a base for the real-time application. Just small changes are required – the model of the controlled system is superseded by output and input blocks. The output block ensures the transfer of control signal to hardware output port. For example if a PC with industrial measurement card is used as a controller, then this block ensures output of control signal to digital analogue converter (DAC) of the card or to digital output of the card. Accordingly, the input block performs a conversion of an input signal of an industrial card (analogue or digital) to Simulink. Furthermore, the blocks representing non-measurable noise are removed from the schema. A Simulink schema for real-time testing is presented in Fig 3.

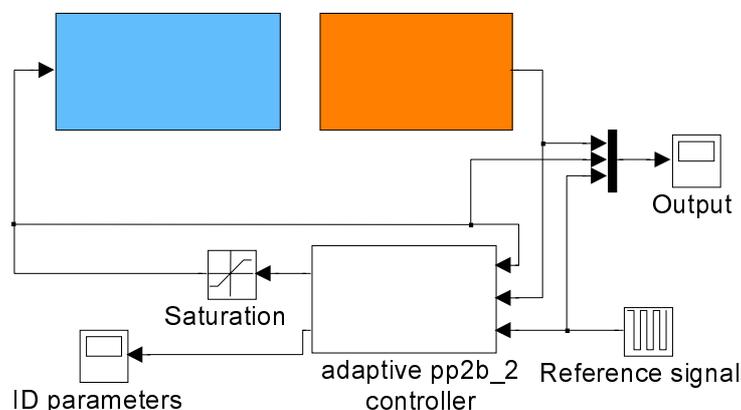


Fig. 3 Simulink schema for real-time control of SISO system

Further step in creation of real-time code consists of selecting desired output platform and starting the conversion from Simulink schema to this platform. The Real-Time Workshop is an open environment and thus the list of available platforms is not restricted to those shipped in original Real-Time Workshop package. 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 target language compiler files.

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

Many manufactures of industrial computers and controllers has created their own target language compiler files that are 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 any user can create his own target language compiler files for converting the block scheme to a source code and hence reach the compatibility with any hardware.

Application creation consists only of selecting appropriate target language compiler file, eventually setting compiler parameters and then start-up of compilation process.

5 VERIFICATION OF DECENTRALIZED APPROACH USING DSPACE SYSTEM

The real-time laboratory model, which was used to verify decentralized approach to the control of multivariable systems, is the Twin Rotor MIMO System shown in Fig 4. This system provides a high-order, non-linear system with significant cross-coupling. The main parts of the system are the pedestal, the jib connected to pedestal and two airscrews at the ends of the jib.

The system jib can freely rotate around horizontal axis by about 100 degrees and around vertical axes by about 330degrees. The system inputs are the voltages used to drive motors of the airscrews and outputs are angular rotations with respect to horizontal and vertical axes.

Despite the strong interactions in the system, decomposition to subsystems is straightforward:

- the first subsystem consists of the small airscrew which drives the angular rotation around vertical axis,
- the second subsystem consists of the big airscrew driving the angular rotation around horizontal axis.

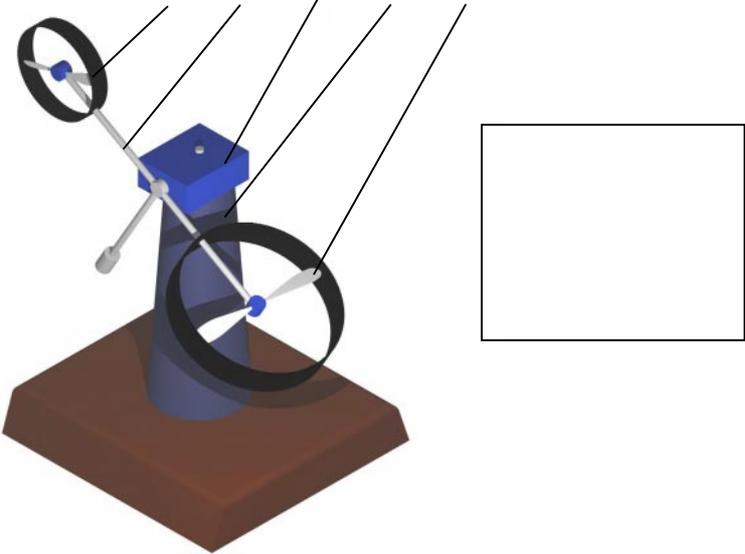


Fig. 4 Twin rotor MIMO model (helicopter)

The system was controlled using dSpace 1104 system. The hardware part of this system is based on PowerPC processor connected to digital and analogue inputs and outputs. The basic parts of affiliated software is Real-Time WorkShop link allowing a conversion of Simulink schemas into the PowerPC platform and the ControlDesk application containing graphical user interface (GUI) for the viewing and the archiving of a courses of the signals in the control circuit. The Fig. 5 presents the main screen of ControlDesk.

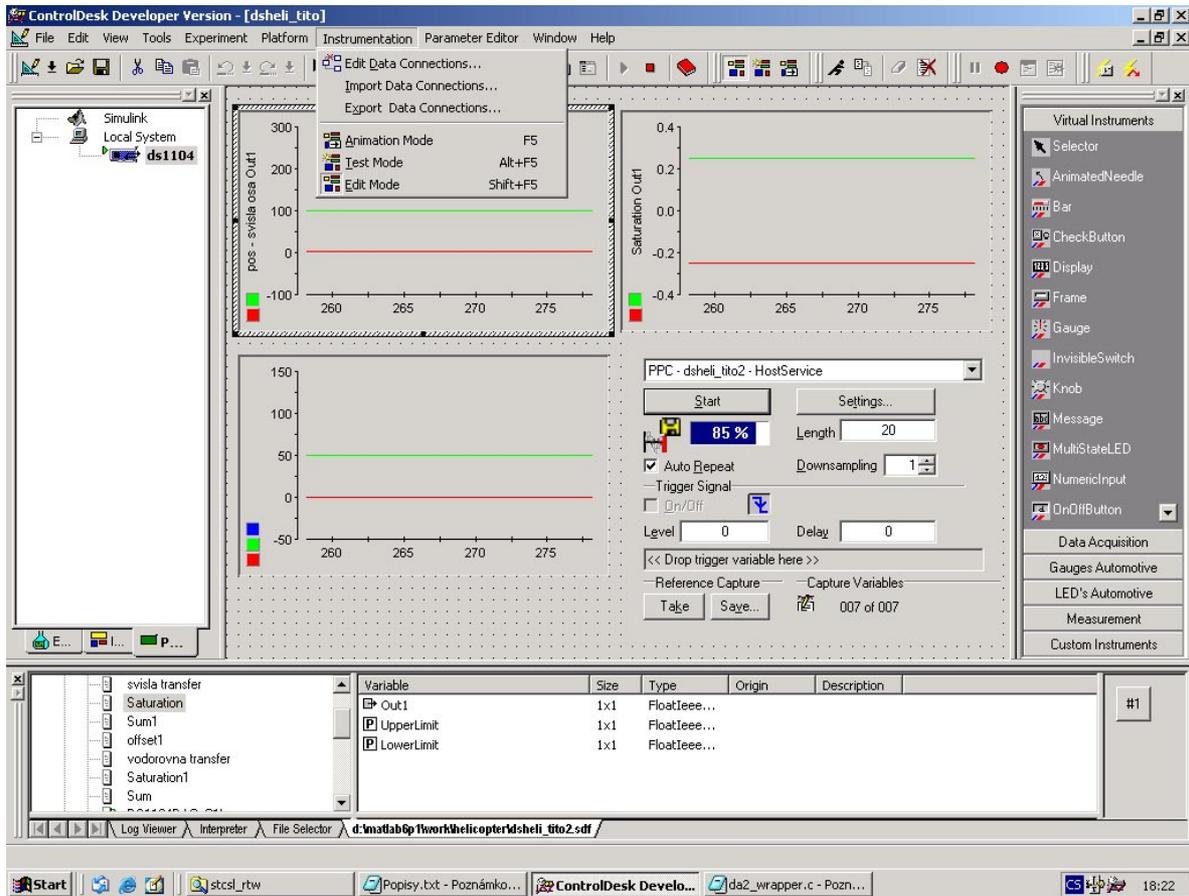


Fig. 5 ControlDesk application

Before the control circuit was connected as a closed loop, the experiments obtaining a static characteristic of the systems had been performed. The resulting static characteristics of the first subsystem are shown in Fig 6.

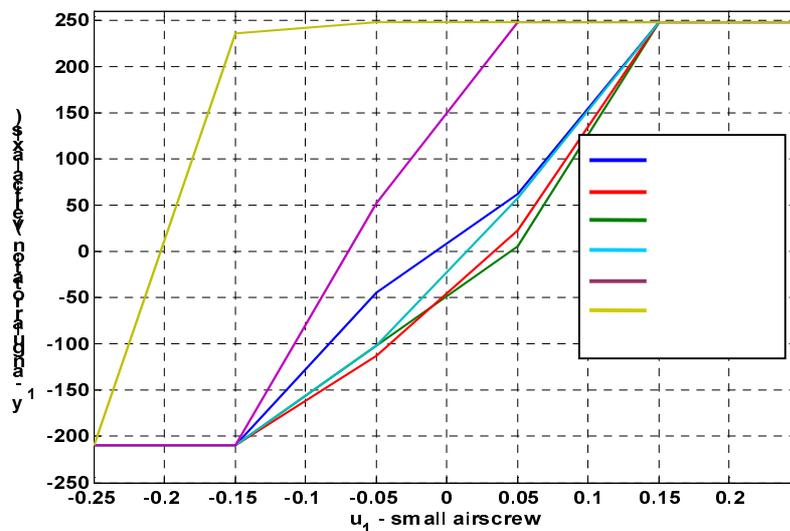


Fig. 6 The static characteristics of the first subsystem

The angular rotations obtained are not in degrees but in 1/500 of circle i.e whole circle corresponds to 500 units. It is obvious that the influence of second input (big airscrew) is great. The static characteristics of the second subsystem are shown in Fig. 7.

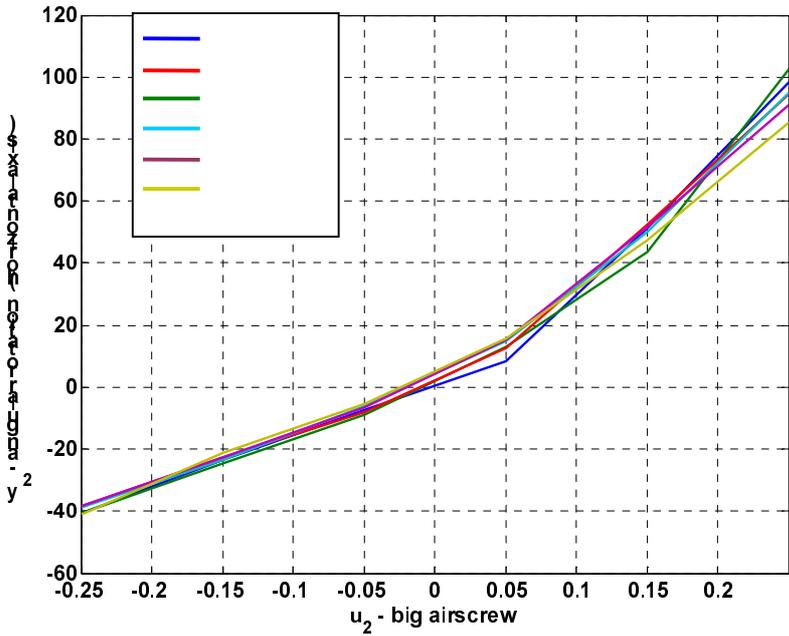


Fig. 7 The static characteristics of the second subsystem

The influence of the first system input to the second system output is very small but the course of the second output is a sign of nonlinearity of the system. The tangent of the static characteristic for $u_2 = 0.25$ is approximately twice as big as for $u_2 = -0.25$.

Another problem of the control of this system is a big hysteresis which is present in the system. The static characteristics of first subsystem, which was measured for increasing and decreasing input signal is shown in Fig 8.

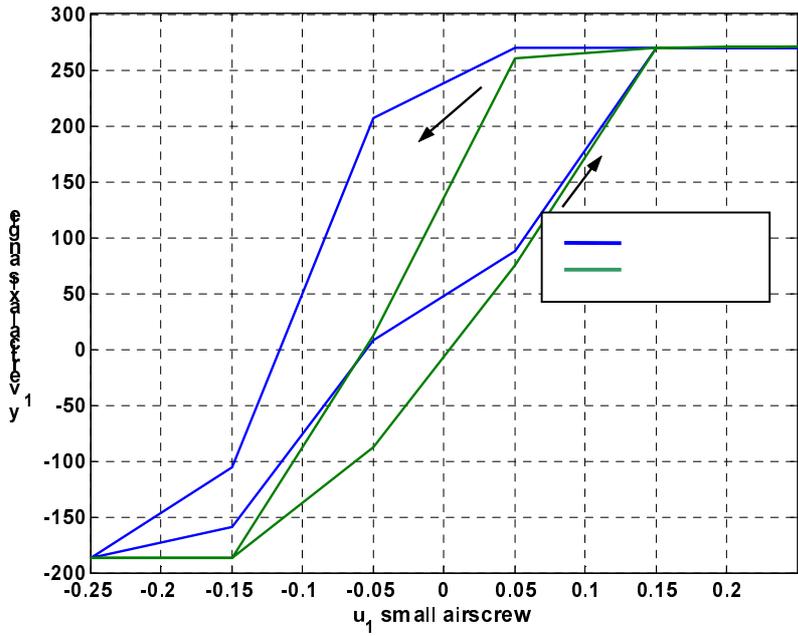


Fig. 8 The static characteristics of the first subsystem showing hysteresis

The Simulink schema used for decentralized control of this TITO system is shown in Fig 9. The schema contained two standalone adaptive controllers and corresponding input and output blocks.

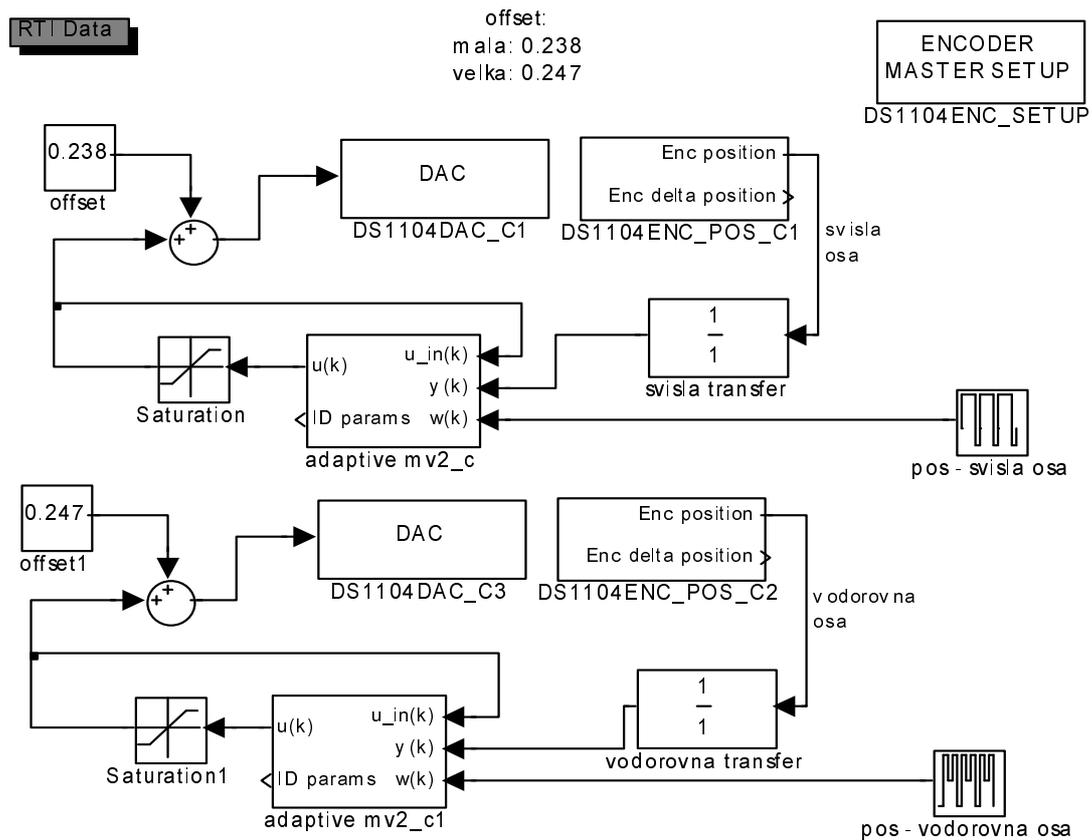


Fig. 9 Simulink scheme used to produce application to control vertical and horizontal rotations

An example of the control courses of twin rotor MIMO system are presented in Fig 10. In this case, the controller synthesis was based on minimizing the variance of output signal (minimum variance).

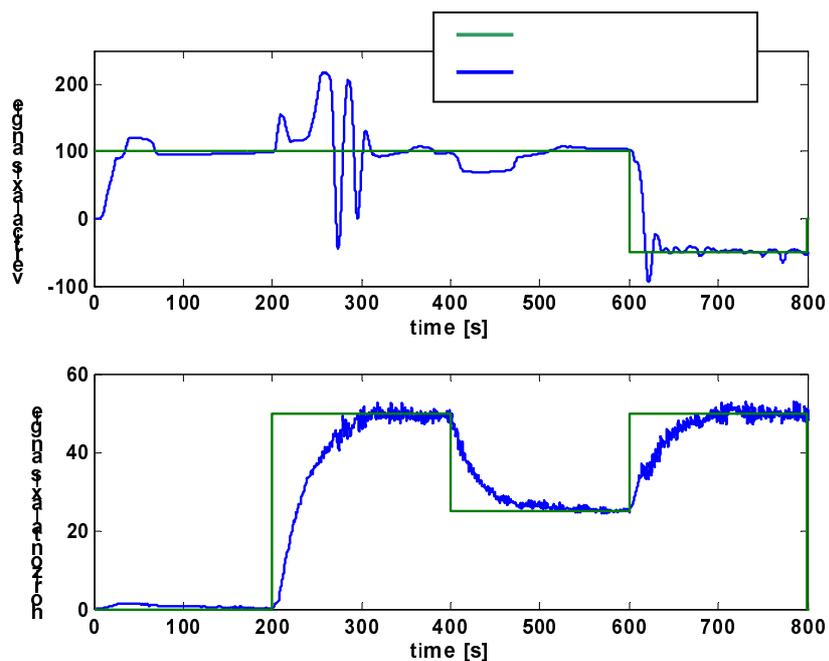


Fig. 10 Control courses using minimum variance controllers

The great influence of changes of second system input to the first output was confirmed by these control courses. Despite this fact, the controllers were able to reach stable control courses.

6 CONCLUSION

The Self-tuning controllers library is used in university course of adaptive control systems. Its architecture enables an easy user orientation in Simulink block schemes and source code of controllers' functions. The controller 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. Real-time laboratory verification confirmed that decentralized approach with usage of self-tuning controllers can be used even for systems with nonlinearity and hysteresis.

7 ACKNOWLEDGMENT

This work was supported in part by the Ministry of Education of the Czech Republic under grant MSM 281100001 and the Grant Agency of the Czech Republic under grants No. 102/02/0204 and No. 102/03/070.

8 LITERATURE

- [1] AOKI, M.: On feedback stabilizability of decentralized dynamic systems, *Automatica* 8, 1972, 163-173.
- [2] KULHAVÝ, R.: Restricted exponential forgetting in real time identification, *Automatica* 23, 1987, 586 – 600.
- [3] HANG, Ch. C, LEE, H. T., HO, W. K.: *Adaptive Control*, Instrument Society of America, 1993, ISBN 1-55617-447-2.
- [4] BOBÁL, V., BÖHM, J., PROKOP, R., FESSL, J.: *Praktické aspekty samočinně se nastavujících regulátorů: algoritmy a implementace*. Nakladatelství VUTIUM, VUT Brno, 1999, ISBN 80-214-1299-2. (in Czech)
- [5] BOBÁL, V., BOHM, J., CHALUPA, P.: MATLAB-toolbox for CAD of simple self-tuning controllers. In: *Proc. 7th IFAC Workshop on Adaptation and Learning in Control and Signal Processing ALCOSP 2001*, Cernobbio-Como, Italy, 2001, 273-278.
- [6] BOBÁL, V., CHALUPA, P.: Self-Tuning Controllers Simulink Library. <http://www.utb.cz/stctool/>, 2002
- [7] ZIEGLER, J. G., NICHOLS, N. B.: Optimum settings for automatic controllers, *Trans. ASME* 64, 1942, 759 – 768

9 CONTACT

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 57 603 3217, fax: +420 57 603 3333

e-mail: bobal@ft.utb.cz, chalupa@ft.utb.cz