Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

of the 18$^{\mathrm{th}}$ International Conference on Process Control

Hotel Titris, Tatranská Lomnica, Slovakia, June 14 – 17, 2011

ISBN 978-80-227-3517-9

http://www.kirp.chtf.stuba.sk/pc11

Editors: M. Fikar and M. Kvasnica

PIDTOOL 2.0 – Software for Identification and PID Controller Tuning

J. Oravec * **M. Bakošová** *

* *Slovak University of Technology in Bratislava, Faculty of Chemical and Food Technology, Institute of Information Engineering, Automation, and Mathematics, Radlinského 9, 812 37 Bratislava, Slovakia*
Tel: +421 2 59325364; e-mail: juraj.oravec@stuba.sk

Abstract: The main aim of this paper is to present a new version of software for PID controller tuning called *PIDTOOL 2.0*. The software represents a user friendly tool for simple step–response–based identification of a process model, fast PID controller tuning, and effective checking the quality of control. It has been developed in the MATLAB–Simulink programming environment using its graphic user interface and can be used as useful and visual software for teaching purposes. In *PIDTOOL 2.0*, user can easily change a language of the graphic user interface. Nowadays, there is a possibility to choose between English and Slovak.

Keywords: PID controller, controller tuning, identification, filtration, graphic user interface

1. INTRODUCTION

The aim of this paper is to present a new version of software for PID controller tuning called *PIDTOOL 2.0* (Oravec (2010), Bakošová and Oravec (2009)). It has been developed at the Institute of Information Engineering, Automation, and Mathematics of the FCFT STU in Bratislava (Čemanová (2007), Oravec (2010)) in the MATLAB – Simulink programming environment and uses its graphic user interface (GUI).

As PID controllers belong to the most used types of controllers in industry (Åström and Hägglund (1995)), *PIDTOOL 2.0* is oriented mainly on PID controller tuning. The software enables to tune PID controllers using various analytical and experimental methods, and new methods described in Xue and Chen (2008), Åström and Hägglund (1995), Bakošová and Fikar (2008), Mikleš and Fikar (2007), Ogunnaike and Ray (1994), Vítečková and Víteček (2006), Lošonský (2006), Weng et al. (1995), Shaw (2006), Bakošová et al. (2003), Kubík and Kotek (1983), and Šulc and Vítečková (2004) were added. If a transfer function of the controlled process is unknown, this software enables to identify the controlled process from its step response (Fikar and Mikleš (1999)). The identified step response can be either damped periodic or aperiodic. The step–response data can be set directly or loaded from the data file. If noisy step–response data have been loaded, designed software enables to run filtration. Properties of the tuned controller can be judged visually and analytically, as *PIDTOOL 2.0* displays simulated control response, time behavior of the manipulated variable and values of various integral performance indexes. So, it is easy to compare several closed–loop step responses generated using various PID controllers with different values of set-points, disturbances and constraints on manipulated variables.

The software *PIDTOOL 2.0* can be used especially for teaching purposes in laboratory exercises oriented on process control. It can be also useful for those who need to identify controlled process from its step–response data, to filter noisy data, to tune PID controllers or to compare various types of control algorithms with simple PID control.

2. PIDTOOL 2.0

PIDTOOL 2.0 solves two basic problems, identification and controller tuning (Fig. 1).

PIDTOOL 2.0 enables to identify a controlled process from its step response. The software distinguishes identification from an aperiodic or a damped periodic step response.

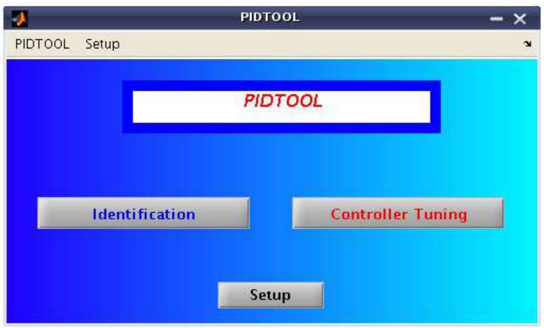

Fig. 1. Basic window of *PIDTOOL 2.0*

The Strejc method (Bakošová and Fikar (2008), Mikleš and Fikar (2007), Lošonský (2006)) is applied for identification from the aperiodic step response and the method described in Mikleš and Fikar (2007) is used for identification from the damped periodic step response. The result of identification is a controlled process model described by the transfer function (1) for aperiodic or (2) for damped periodic step response

$$G(s) = \frac{K}{(Ts+1)^n} e^{-Ds} \quad (1)$$

$$G(s) = \frac{K}{T^2 s^2 + 2\xi T s + 1} e^{-Ds} \quad (2)$$

where n is the order of the system, K is the gain, T is the time constant, ξ is the damping coefficient and D is the time delay.

The identification can be simply started using button *Identification* located in the basic window (Fig. 1). The next window (Fig. 2) offers three identification possibilities.

Fig. 2. Basic window of identification

The first button *Step Response Data* enables to identify the controlled process directly from data obtained from the measured and recorded step response. The second button *Load the Data File* (Fig. 2) opens the new window (Fig. 3), where user can comfortably find out a required data file containing recorded step–response data. The considered structure of the data file is as follows, the first column vector represents a time and the second column vector represents associated measured values of output variable. *PIDTOOL 2.0* enables to load the data file which includes also the third column vector of values of manipulated variable (Fig. 4). If several step responses are included in the loaded data file, they are automatically recognized and the nominal step response is evaluated (Fig. 5). It enables to reach the nominal transfer function of identified non-linear controlled process. This possibility makes the identification from step response using this software even user–friendlier.

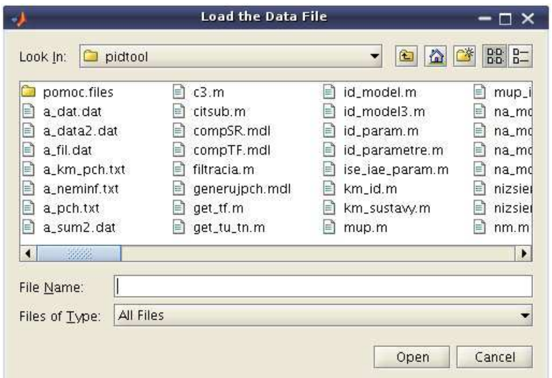

Fig. 3. Window for loading the datafile

In the next window, there is a choice of data processing (Fig. 6). To obtain the aperiodic model of controlled process, user can directly use the button *Identification*.

When the damped periodic model is required, user can simply activate the checkbox *Periodic process* and then

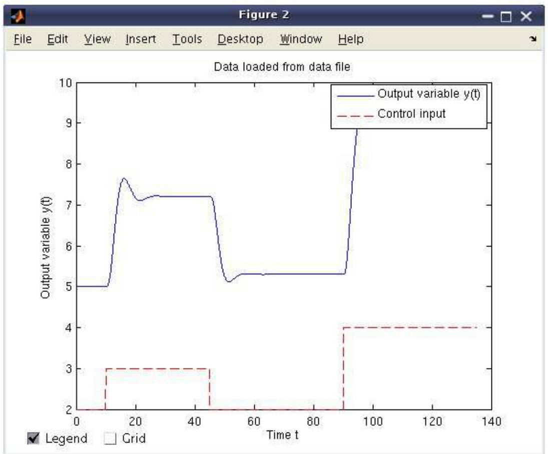

Fig. 4. Loaded data of step response

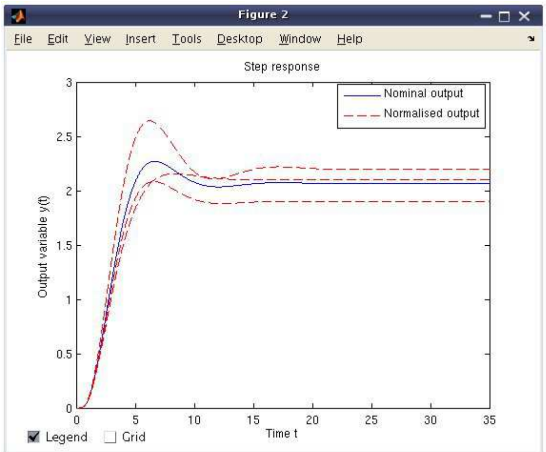

Fig. 5. Nominal and normalized step responses

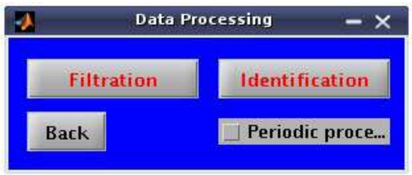

Fig. 6. Choice of processing of loaded data

use the button *Identification*. If the controlled process has been identified using the Strejc method, the tangent to the step response is also depicted and its equation is given (Fig. 7). In the new window, the parameters of model (1) or (2) of identified process are shown (Fig. 8, Fig. 9).

PIDTOOL 2.0 enables to use button *Identification Tuning* (Fig. 10) to receive the transfer function, which generates the step response, that covers the original one more precisely. In the new window (Fig. 11) the step response of identified transfer function can be simply modified by changing the slope of its tangent (Lošonský (2006)). The new parameters of identified transfer function are directly shown. It helps to check whether the identified transfer function has still required properties, e.g. the order n. If the loaded data are noisy, the user can use the filtration before identification, simply using the button *Filtration* (Fig. 6). Then the new window for filtration is opened

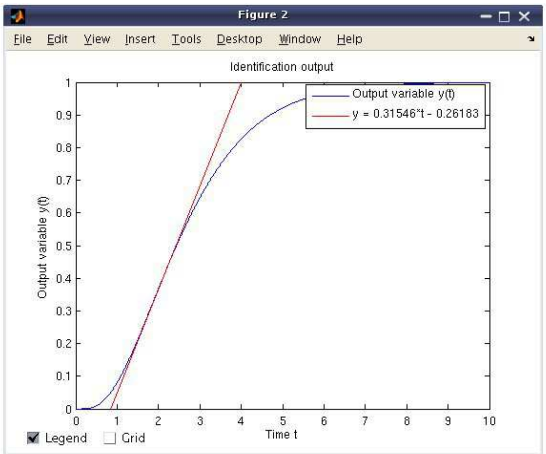

Fig. 7. Step response of identified process

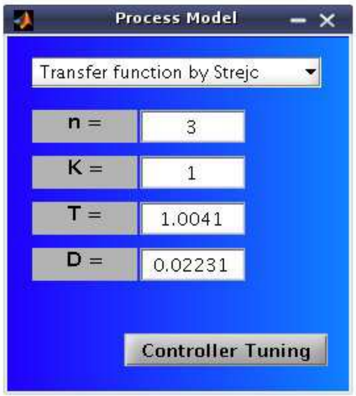

Fig. 8. Identified parameters of aperiodic system

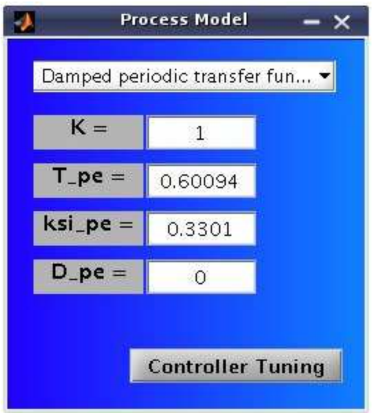

Fig. 9. Identified parameters of damped periodic system

(Fig. 12). By using the button *Save* (Fig. 12), user can simply store reached filtered data into the new data file for later usage. After filtration, identification can be started.

The third button *Process Model Data* (Fig. 2) enables to identify process model with required properties of transfer function. Using this button, the new window shown in Fig. 13 will be opened. In this window, the parameters of the model described by the transfer function (3) can be simply set.

$$G(s) = \frac{Num(s)}{Den(s)} e^{-Ds} \tag{3}$$

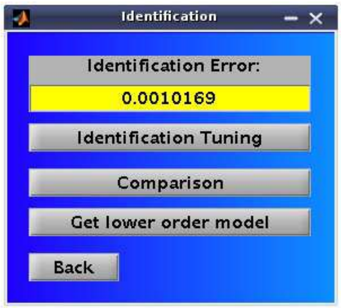

Fig. 10. Additional options of identification

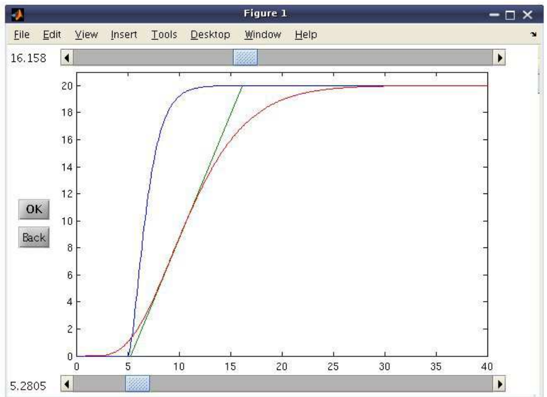

Fig. 11. Window for identification tuning

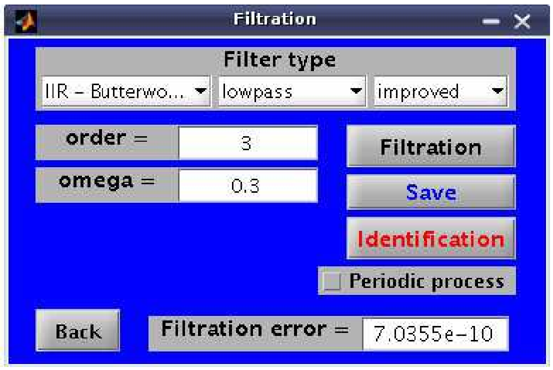

Fig. 12. Window of filtration

If the checkbox *Periodic process* has been activated (Fig. 13), given model (3) will be approximated by the transfer function (2), otherwise by the transfer function (1). The approximation is started using the button *Identification* (Fig. 13). This approximation of the given model can be useful in the case when chosen PID controller tuning method requires controlled model described by the transfer function (1) or (2).

The software *PIDTOOL 2.0* is oriented mainly on PID controller tuning. PID controllers can be designed for controlled process models with either damped periodic or aperiodic step responses described by the transfer functions (1) or (2). To run direct controller tuning, user can use the button *Controller Tuning* located in the basic window (Fig. 1), or use this button after identification (Fig. 9, Fig. 8). Then, a window is opened where the user can choose a required type of PID controller and a type of a tuning method (Fig. 14).

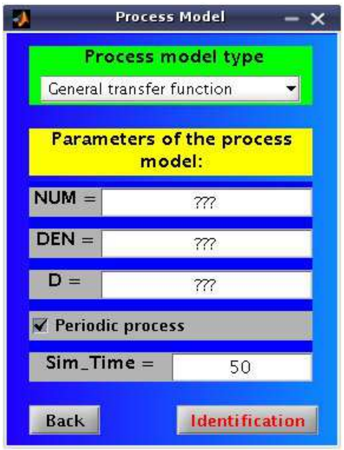

Fig. 13. Process model data

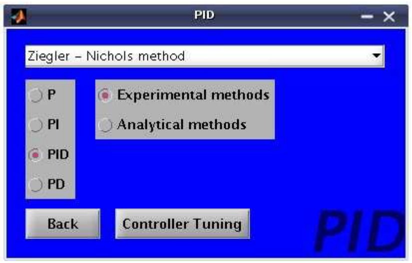

Fig. 14. Window for PID controller tuning

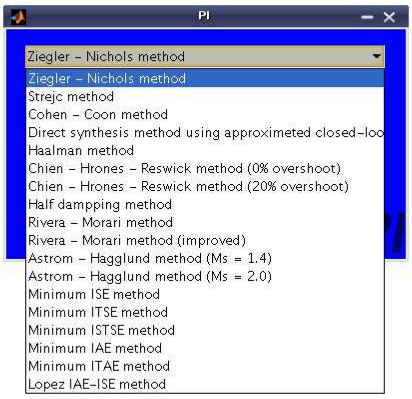

Fig. 15. List of experimental methods used for PI controller tuning

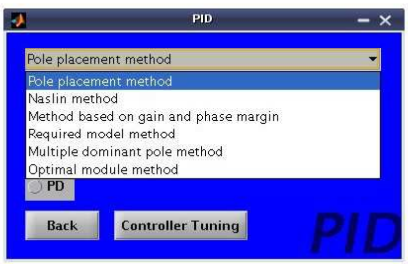

Fig. 16. List of analytical methods used for PID controller tuning

It is possible to choose a P, PI, PID or PD controller. For simpler handling, the methods for controller tuning are divided into two main groups: analytical and experimental methods. Various types of analytical (Fig. 16) and experimental methods (Fig. 15) can be used for controller tuning. Calculated parameters of the tuned controller are shown in the new window (Fig. 17), where Z_R is the gain, T_I is the reset time and T_D is the derivative time of the controller.

Fig. 17. Window for evaluating the quality of control

In the top right part of the opened window (Fig. 17), the parameters of the transfer function of the controlled process are also shown. These parameters can be modified and the tuned controller can be so tested in the presence of model uncertainty. The properties of the closed loop with the tuned controller can be judged by simulation of control. The standard control law (4) is supposed as a default control law. Setting the parameters b_W, T_R and T_F enables to use improved form of control law. The proportional part of control law is then modified by the value of the parameter b_W. Setting this parameter enables to change the weight of set–point in the control error evaluation (5). The non zero value of the parameter T_R modifies the integral part of control law (6). Setting this parameter enables to prevent integral windup. The parameter T_F modifies the derivative part of control law (7). The non zero value of this parameter represens a filter of derivative part of control law to obtain the proper transfer function of the derivative part of the controller.

$$u(t) = Z_R e(t) + \frac{Z_R}{T_I}\int_0^t e(t)dt + Z_R T_D \frac{de(t)}{dt} \tag{4}$$

$$u_P(t) = Z_R\left(b_W w(t) - y(t)\right) \tag{5}$$

$$u_I(t) = \frac{Z_R}{T_I}\int_0^t \left(e(t) - \frac{T_I}{Z_R} T_R\left(u(t) - u_{SAT}(t)\right)\right) dt \tag{6}$$

$$u_D(t) = \frac{Z_R T_D s}{1 + \frac{T_D}{T_F} s} \tag{7}$$

Using the button *Step Response* (Fig. 17) runs quality evaluation of control. The set–point tracking and the disturbance rejection can be simulated in the presence of boundaries on the control input. For simpler handling, the parameters have preset default values. In the case, the manipulated variable is constrained, user can compare the closed-loop step responses and the time behaviors of the manipulated variable before and after the saturation. After simulation, the closed–loop step responses (Fig. 18) and the time behaviors of the manipulated variable (Fig. 19) are shown. In the case the legend overlaps the displayed graph (Fig. 18), it is possible to deactivate the checkbox *Legend* and to hide the shown legend. Using the checkbox *Grid* leads to displaying the grid of the shown graph. Both of these possibilities are included in all displayed graphs, generated by this software. The quality of control (Fig. 20) can by also judged by calculating several integral performance indexes (Mikleš and Fikar (2007), Bakošová et al. (2003), Xue and Chen (2008), Oravec (2010), Čemanová (2007)).

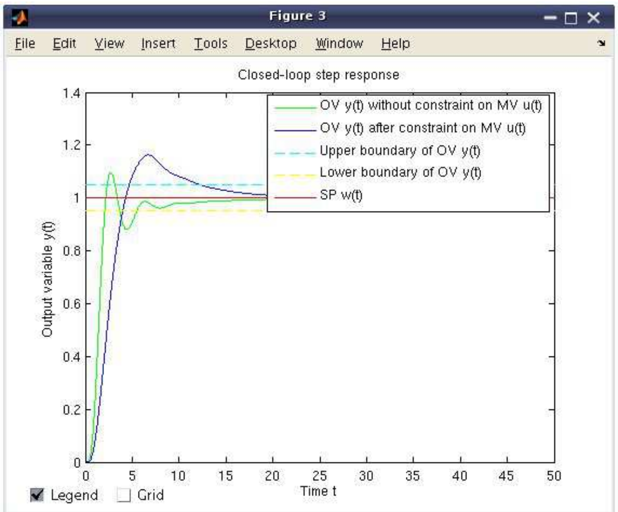

Fig. 18. Closed–loop step response

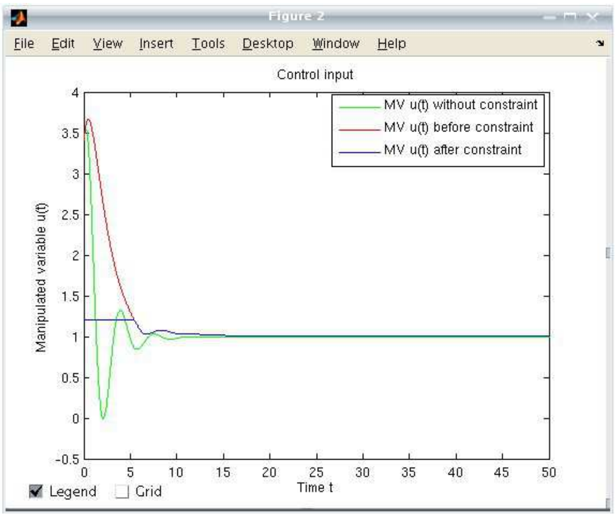

Fig. 19. Control input

Using the possibilities of window shown in Fig. 17, it is easy to compare several step responses and values of

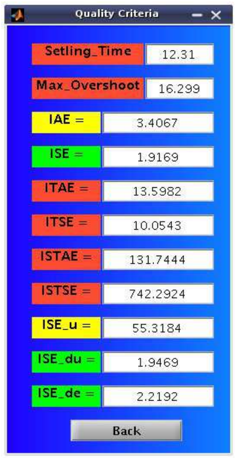

Fig. 20. Values of performance indexes

performance indexes reached with different values of set–points, disturbances and constraints on the manipulated variables. The values of all the parameters, which are necessary for simulations are stored by *PIDTOOL 2.0*. Calculated values of performance indexes are also stored. These stored data can be simply shown by using *Setup/Show results* located in basic window (Fig. 21). The stored data are transformed into *html*-file. The stored data are shown in simple summary table with date and time of simulation. This new ability can be helpful in the case, when many simulations at different conditions have been evaluated and user wants to compare obtained results to make decision, which controller is the most suitable for control.

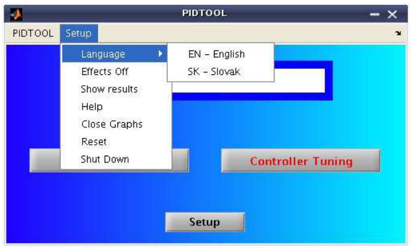

Fig. 21. Setup menu of *PIDTOOL 2.0*

It is easy to change a language of graphic user interface. Actually, there is a possibility to choose between English and Slovak language (Fig. 21). Also other languages can be simply added into the software *PIDTOOL 2.0*.

3. CONCLUSION

The software *PIDTOOL 2.0* has been developed in the MATLAB – Simulink programming environment using its Graphical User Interface and offers to the user a comfortable and visual environment for fast identification, simple PID controller tuning and effective evaluating the quality of control in various conditions. The values necessary for evaluating the quality of control and calculated quality criteria are stored and can be displayed in a simply summary table. Using this new ability helps to decide, which controller is the most suitable for control. This software has been tested by students at the FCFT in two courses; in the course Process Dynamics and Control that is taught in the first year of the Master study and the course Integrated Control in the Process Industry that is taught in the third year of the Bachelor study. Using the software is limited only for a teaching purposes. To obtain the software contact the authors.

There exist various software applications, which enables to design the PID controller, for example see (Lošonský (2006)) or (Schlegel and Čech (2004)). The comparison of the available applications will be subject of our next work.

ACKNOWLEDGMENTS

The authors are pleased to acknowledge the financial support of the Cultural and Educational Grant Agency KEGA of the Slovak Republic under grant No. 3/7245/09 and of the Scientific Grant Agency VEGA of the Slovak Republic under the grants 1/0537/10.

REFERENCES

Bakošová, M. and Fikar, M. (2008). *Process Control (in Slovak)*. STU Press, Bratislava.

Bakošová, M. and Oravec, J. (2009). Software for PID controller tuning. *Proceedings of the 17th Interantional Conference on Porcess Control'09, FCFT SUT in Bratislava*, 524–527.

Bakošová, M., Ľ. Čirka, and Fikar, M. (2003). *Automatic control fundaments (in Slovak)*. Vydavateľstvo STU, 1st ed. edition.

Fikar, M. and Mikleš, J. (1999). *System identification (in Slovak)*. STU Press, Bratislava.

Kubík, S. and Kotek, Z. (1983). *Automatic control theory (in Czech)*. SNTL, 1st ed. edition.

Lošonský, J. (2006). *System for support of control loops (in Slovak)*. Master's thesis, Trnava: MTF SUT in Bratislava.

Mikleš, J. and Fikar, M. (2007). *Process Modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg.

Ogunnaike, B. and Ray, W.H. (1994). *Process dynamics, modeling and control*. Oxford University Press, New York.

Oravec, J. (2010). *Design of a Program System for Controller Tuning (in Slovak)*. Master's thesis, FCFT SUT in Bratislava.

Åström, K. and Hägglund, T. (1995). *PID controllers: theory, design and tuning*. Triangle Park, 2nd ed. edition.

Schlegel, M. and Čech, M. (2004). Internet PID controller design : www.pidlab.com. *IBCE 04*, 1–6.

Shaw, J.A. (2006). *The PID Control Algorithm [online]*. http://www.docstoc.com/docs/20312583, 2nd ed., version 2.2 edition.

Čemanová, L. (2007). *Design of GUI for controller tuning (in Slovak)*. Master's thesis, FCFT SUT in Bratislava.

Vítečková, A. and Víteček, M. (2006). *Automatic control fundamentals (in Czech)*. TU Ostrava.

Šulc, B. and Vítečková, M. (2004). *Theory and praxis of control design (in Czech)*. ČVUT Press, Praha.

Weng, K.H., Chang, C.H., and Lisheng, S.C. (1995). Tuning of PID Controllers Based on Gain and Phase Margin Specifications. *Pergamon, Automatica*, 31(3), 497–502.

Xue, D. and Chen, Y.Q. (2008). *Analysis and Design with MATLAB (Advanced in Design and Control)*. Society for Industrial Mathematics, 1st ed. edition.