

PIDTOOL - SOFTWARE FOR 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

Abstract

The main aim of the PIDTOOL software design is to create a user friendly tool for simple step-response-based identification of a process model, fast PID controller tuning and effective evaluating the quality of control. The PIDTOOL has been developed in the MATLAB-Simulink programming environment using its graphic user interface. The PIDTOOL software is oriented mainly on PID controller tuning, but in the case the process model is unknown, the PIDTOOL enables to identify the controlled process from its step response. In the new version of PIDTOOL, the step response can be either dumped periodic or aperiodic. The step response data can be set directly or loaded from the data file. In the case the noisy step response data are loaded, PIDTOOL enables to run filtration. The result of identification is a model of a controlled process in the form of the n -th order plus time-delay transfer function. In the case of aperiodic step response, there is a possibility to choose between identification of the 1st order plus time delay transfer function or the higher order plus time delay transfer function. The PIDTOOL enables to tune various types of PID controllers, such as P, PI, PID or PD controllers. Various analytical and experimental methods can be chosen. The properties of the tuned controller can be simply and visually judged by PIDTOOL using simulation of the closed-loop step response. The PIDTOOL displays the figures of the simulated control response and the time behavior of the manipulated variable. It is easy to compare several step responses generated with different values of set-points, disturbances and constraints on the manipulated variable. Values of various quality criteria of control are calculated. These visual and analytical outputs serve to decide, whether the tuned controller is suitable for control of given process. In the new version of PIDTOOL, user can easily change a language of PIDTOOL graphic user interface. Actually there is a possibility to choose between English and Slovak language. Other languages can be simply added into the software. The PIDTOOL can be used as useful and visual software for teaching purposes. The PIDTOOL is used for teaching at the IIEAM FCFT STU in courses oriented on process control.

1. Introduction

The aim of this paper is to present the new version of the software PIDTOOL. It has been developed in the MATLAB – Simulink programming environment and uses its graphic user interface. PIDTOOL has been designed as a user friendly tool for simple step identification of a process model, fast PID controller tuning and effective checking the quality of control. Because the PID controllers belong to the most used types of controllers in the industry [1], the PIDTOOL software is oriented mainly on PID controller tuning. PIDTOOL enables to tune PID controllers using various analytical and experimental methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. In the case the controlled process is unknown, PIDTOOL enables to identify the controlled process from its step response [12]. The software PIDTOOL is used for teaching purposes at the FCFT STU in Bratislava.

2. PIDTOOL

The software PIDTOOL has been developed in the MATLAB – Simulink programming environment at the Institute of Information Engineering, Automation and Mathematics FCFT STU in Bratislava [13], [14], [15], [16]. The software PIDTOOL uses MATLAB graphic user interface for simpler manipulation and better visualization. The main aim of the software PIDTOOL developing was to create a useful tool for process identification, PID controller tuning and fast evaluation the quality of control. The software PIDTOOL can be used especially for teaching purposes in laboratory exercises oriented on process control. Also everybody, who wants to tune PID controllers or needs to compare various types of control algorithms with simple PID control, can use PIDTOOL.

The new version of PIDTOOL enables to tune PID controller for controlled process models with either dumped periodic or aperiodic step responses described by transfer functions (1), (2)

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

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

where n is the order of the system, K is the gain, T is the time constant, ζ is the dumping coefficient and D is the time delay. In the case the controlled process model is unknown, the PIDTOOL enables to identify the controlled process from its step response. The step response data can be set directly or loaded from the data file. In the case the noisy step response data are loaded, PIDTOOL enables to run filtration. In the new version of PIDTOOL, software distinguishes identification from the aperiodic or dumped periodic step response. The Strejc method [2], [3], [6] is used for identification from the aperiodic step response. The method described in [3] is used for identification from the dumped periodic step response. The result of identification is a controlled process model described by the transfer function (1) for aperiodic case or (2) for periodic one. Identification can be simply started using button *Identification* located in the basic window (Fig. 1).

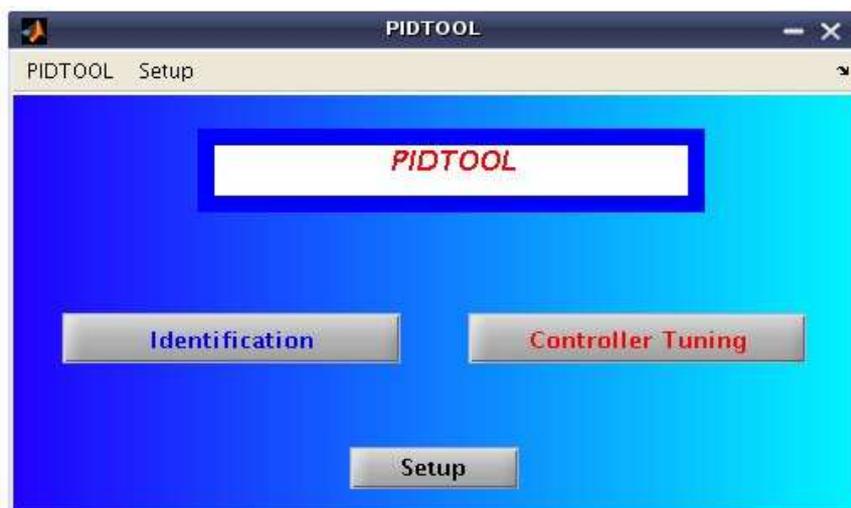


Figure 1: Basic window of PIDTOOL

Using the button *Identification* opens the new window (Fig. 2), which offers three possibilities. 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* enables to run identification from the data loaded from the data file. The third button *Process Model Data* enables to identify process model described by general transfer function which is used for data generation.

Using the button *Step response data* (Fig. 2) opens the window for direct data inserting (Fig. 3), (Fig. 4). The new version of software PIDTOOL enables to choose between setting aperiodic (Fig. 3) or dumped periodic (Fig. 4) step response parameters. In the opened window, important parameters describing the step response have to be set. Sake of simple using, some of these parameters have preset default values. Inserted values of input parameters are stored and they are used as default values in the next identification unless the user changes them.

Using the button *Identification* in Fig. 3 opens the new window (Fig. 5). The software PIDTOOL enables to choose the required order of controlled process model. The first button *1st Order System* is used when user requires the 1st order plus time delay transfer function. The calculated parameters of (1) are shown in the new figure (Fig. 6). The step response of the identified aperiodic model is displayed (Fig. 7). The tangent to the step response is also depicted and its equation is given. The second button *n-th Order System* is used for obtaining the higher order plus time delay transfer function. The calculated parameters of (2) are shown in the new figure (Fig. 8) and the step response of the model with the tangent is displayed (Fig. 9).



Figure 2: Basic window of identification

The button *Identification* in (Fig. 4) serves for identification of the controlled process from its dumped periodic step response. The results of identification are shown in Figs. 10 – 11. Using the button *Load the Data File* (Fig. 2) opens the window (Fig. 12) where user can comfortably find out a required data file containing recorded step response data. Then, the user is asked for inserting the step time and the step change of the input variable that was set for measurement of the step response of the controlled process (Fig. 13). In the next window, there is a choice of data processing (Fig. 14). To obtain the aperiodic model of controlled process, user can directly use the button *Identification*. When the dumped periodic model is required, user can simply activate the checkbox *Periodic process* and then use the button *Identification*. If loaded data are noisy, the software PIDTOOL enables to run filtration before identification using the button *Filtration* (Fig. 14). In this case, the new window for filtration is opened (Fig. 15). Using the button *Save* (Fig. 15), user can simply store reached filtrated data for later usage in the new data file. After filtration, identification can be started.

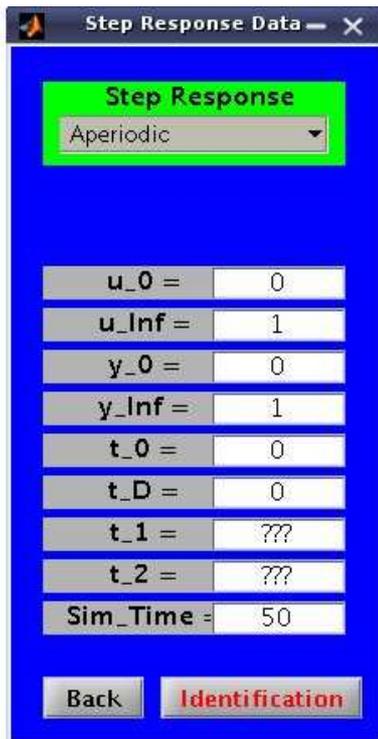


Figure 3: Inserting parameters of aperiodic step response

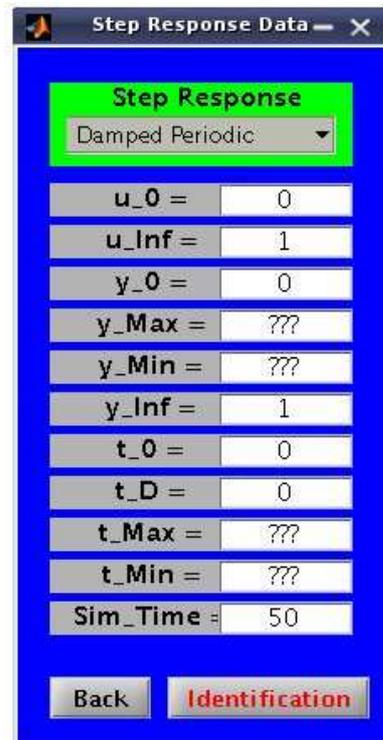


Figure 4: Inserting parameters of dumped periodic step response



Figure 5: Choice of the order of the identified transfer function

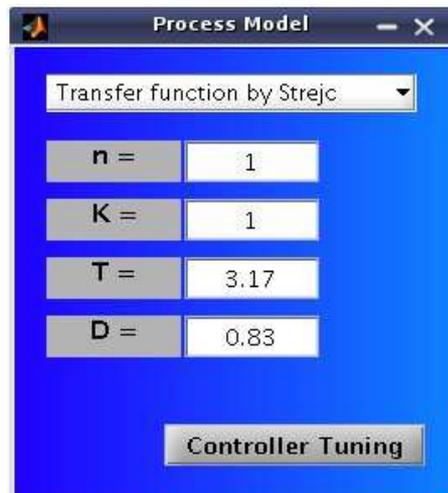


Figure 6: Calculated parameters of identified model described by the 1st order plus time delay transfer function

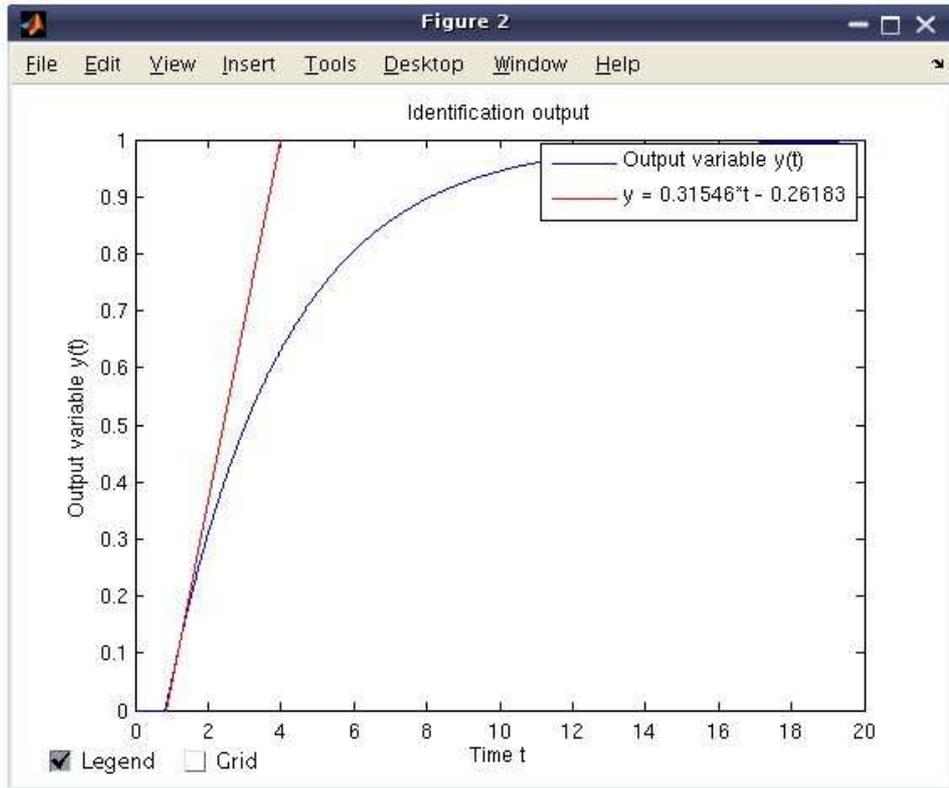


Figure 7: Step response of the identified model described by the 1st order plus time delay transfer function

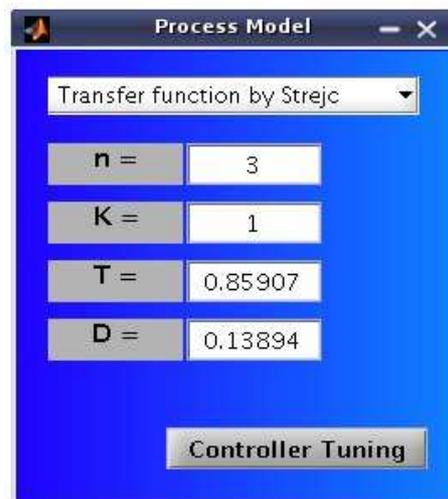


Figure 8: Calculated parameters of identified model described by the higher order plus time delay transfer function

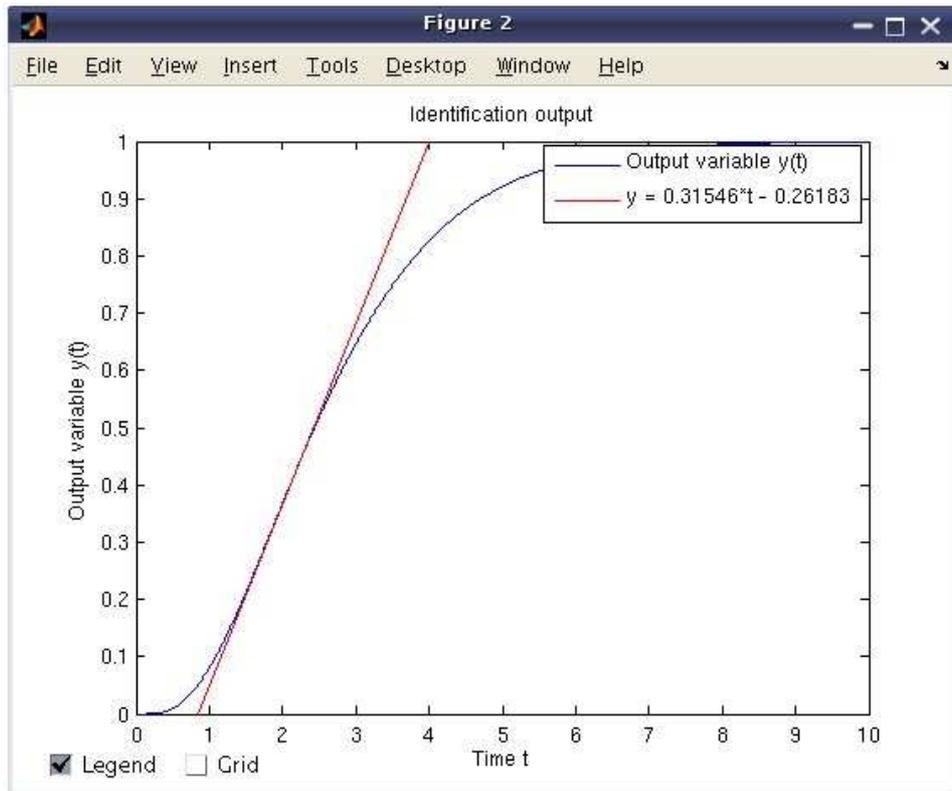


Figure 9: Step response of the identified model described by the higher order plus time delay transfer function

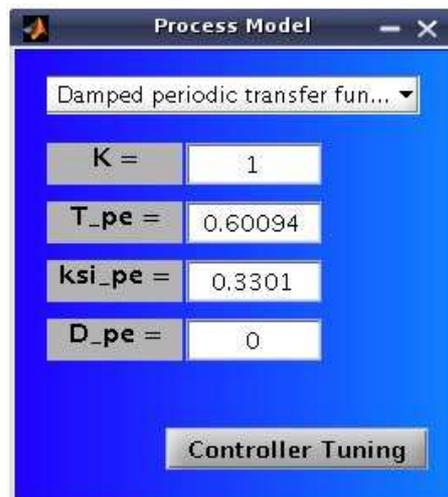


Figure 10: Calculated parameters of identified model with dumped periodic response

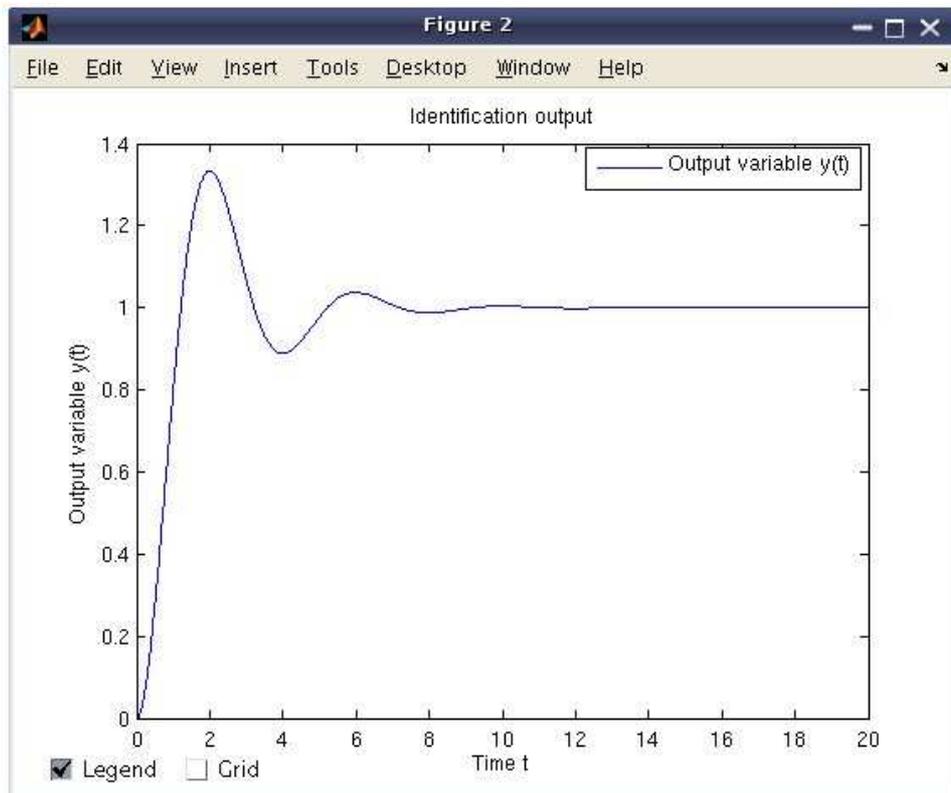


Figure 11: Step response of the identified model with damped periodic response

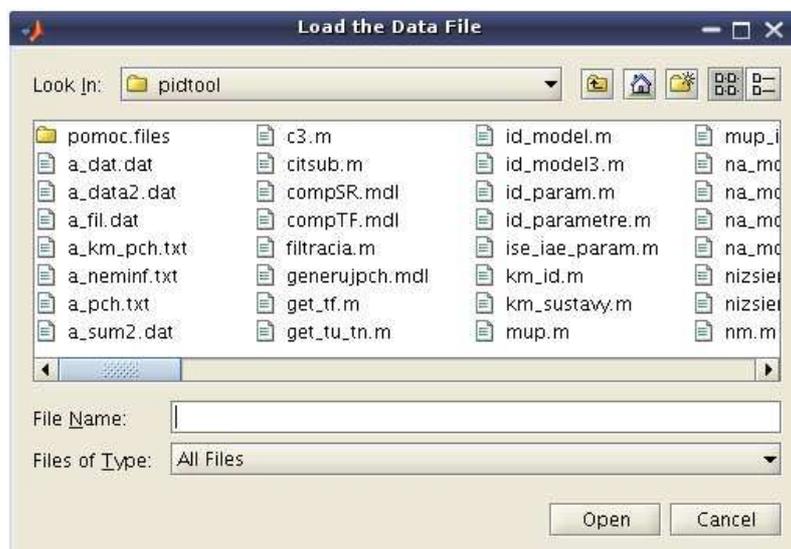


Figure 12: Window for loading the data file

Using the button *Process Model Data* (Fig. 2) opens the new window shown in Fig. 16. In this figure the parameters of the model described by the transfer function (3) can be simply set.

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

In the case the checkbox *Periodic process* is active (Fig. 16), 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. 16). This approximation of controlled model can be useful in the case when chosen PID controller tuning method requires controlled model described by the transfer function (1) or (2).



Figure 13: Input parameters of the step change

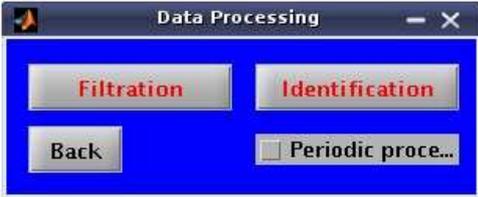


Figure 14: Choice of loaded data processing

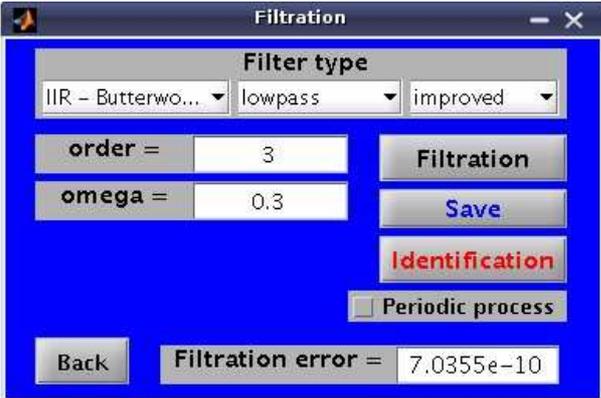


Figure 15: The window for filtration

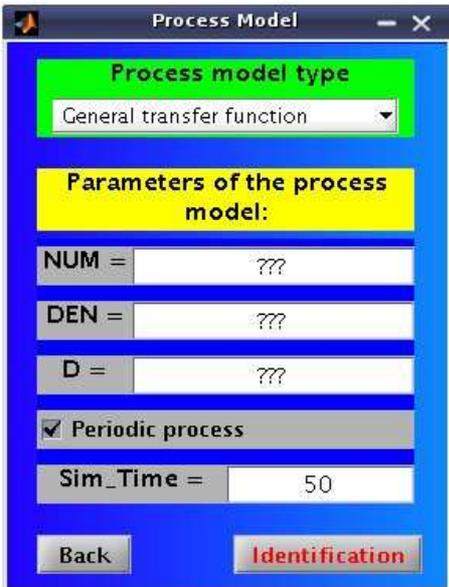


Figure 16: Process model data

The software PIDTOOL is oriented mainly on PID controller tuning. To run direct controller tuning, user can use the button *Controller Tuning* located in the main window of the software PIDTOOL (Fig. 1), or use the button *Controller Tuning* after identification (Fig. 6), (Fig. 8), (Fig. 10). The new version of software PIDTOOL opens one summary window where user can comfortable set parameters of required type of PID controller. It is possible to choose the type of the tuned controller, such as P, PI, PID and PD (Fig. 17). Various types of analytical (Fig. 18) and experimental methods (Fig. 19) can be used for controller tuning. Calculated parameters of the designed controller are shown in the new window for evaluating the quality of control (Fig. 20), where Z_R is the gain, T_I is the reset time and T_D is the derivative time of the controller. In top right part of the opened window (Fig. 20), the parameters of the transfer function of the controlled process are shown.

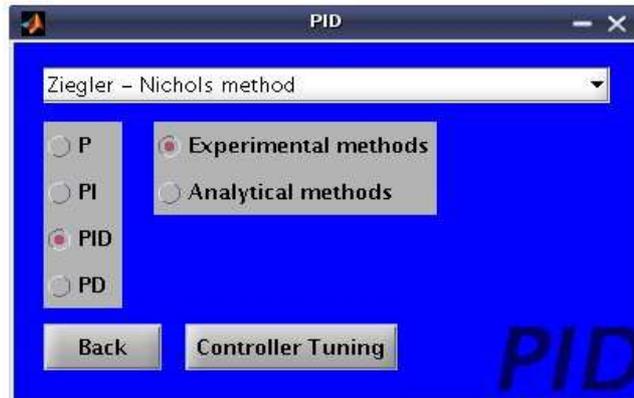


Figure 17: Window for PID controller tuning

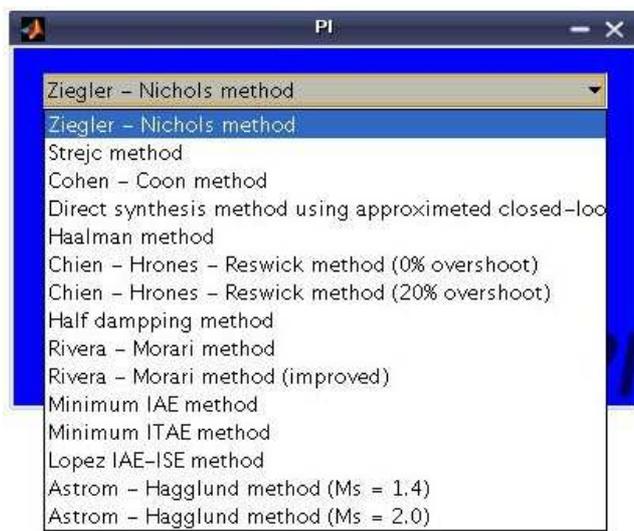


Figure 18: List of experimental methods used for PI controller tuning

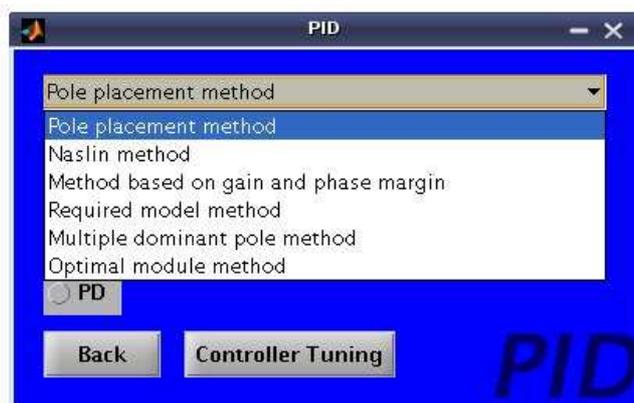


Figure 19: List of analytical methods used for PID controller tuning

These parameters can be changed and the tuned controller can be tested in the presence of model uncertainty. The software PIDTOOL enables to judge the quality of the tuned controller by simulation of control. The set-point tracking and the disturbance rejection can be simulated in the presence of boundaries on the control input.

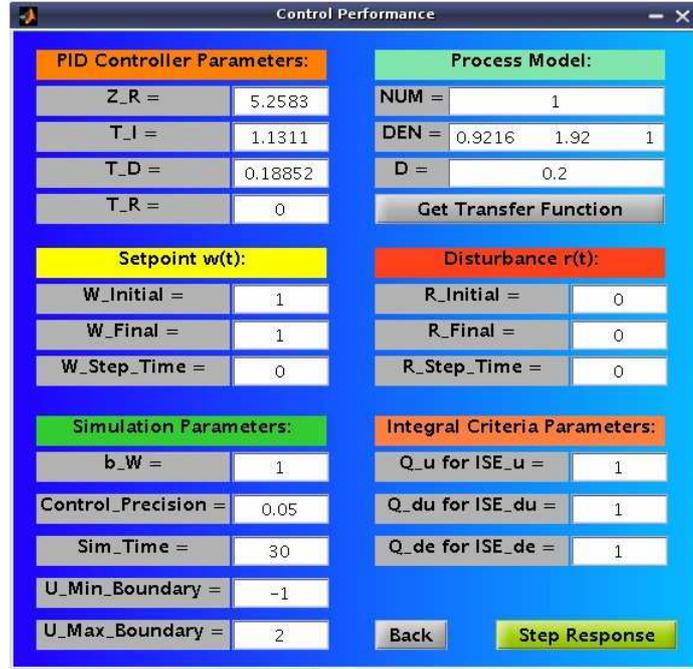


Figure 20: Window for evaluating the quality of control

In the case the manipulated variable is constrained, user can compare the closed-loop step responses and time behavior of the manipulated variable before and after the saturation. Using the button *Step response* (Fig. 20) leads to displaying the closed-loop step responses (Fig. 21) and the time behavior of the manipulated variable (Fig. 22). In the case the legend overlaps the displayed graph (Fig. 21), it is possible to deactivate the checkbox *Legend* 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 the new version of the software PIDTOOL. The quality of control can be also judged by calculating several quality criteria (Fig. 23). The new version of PIDTOOL evaluates some more quality criteria. The quality can be judged by value of the settling time t_{set} (4), maximal overshoot (5) or by calculated values of various integral criteria (6) – (10), where $y(t)$ represents the output variable, δ represents the control precision, $u(t)$ represents the manipulated variable and $e(t)$ represents the control error. The parameters Q_u , Q_{du} , Q_{de} in (8) – (10) have default values and can be modified by user (Fig. 20).

$$|y(t) - y(\infty)| \leq \delta \quad \text{for } t \geq t_{set} \quad (4)$$

$$\sigma_{max} = \frac{y_{max} - y(\infty)}{y(\infty) - y(0)} 100\% \quad (5)$$

$$I_{IAE} = \int_0^{\infty} |e(t)| dt \quad (6)$$

$$I_{ISE} = \int_0^{\infty} e^2(t) dt \quad (7)$$

$$I_{ISE(t)} = \int_0^{\infty} t e(t)^2 dt \quad (8)$$

$$I_{ISE(t)} = \int_0^{\infty} t |e(t)| dt \quad (9)$$

$$I_{ISE(u)} = \int_0^{\infty} (e(t)^2 + Q_u u(t)^2) dt \quad (8)$$

$$I_{ISE(du)} = \int_0^{\infty} \left(e(t)^2 + Q_{du} \left(\frac{du(t)}{dt} \right)^2 \right) dt \quad (9)$$

$$I_{ISE(de)} = \int_0^{\infty} \left(e(t)^2 + Q_{de} \left(\frac{de(t)}{dt} \right)^2 \right) dt \quad (10)$$

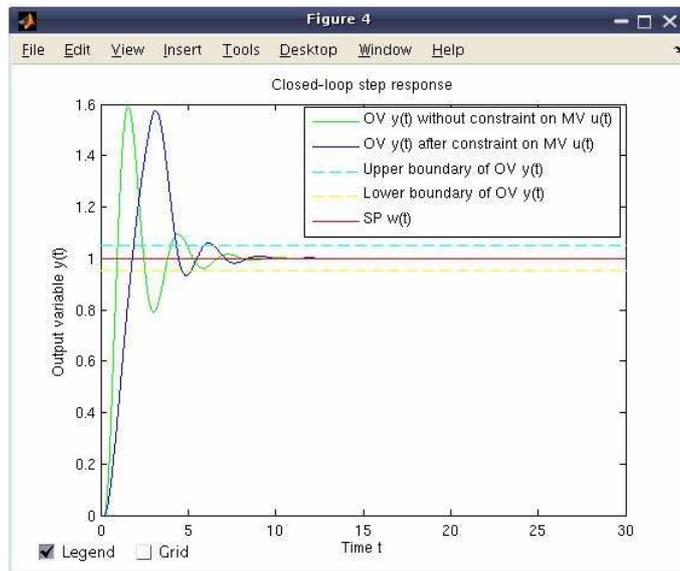


Figure 21: Closed-loop step response

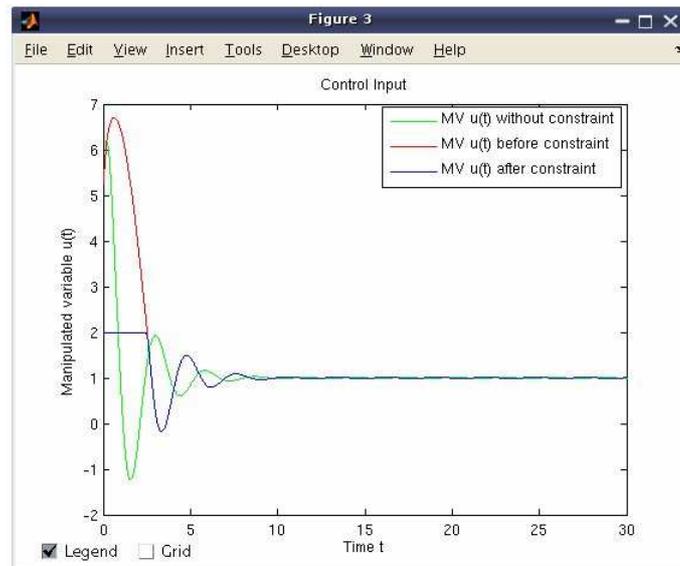


Figure 22: Control input

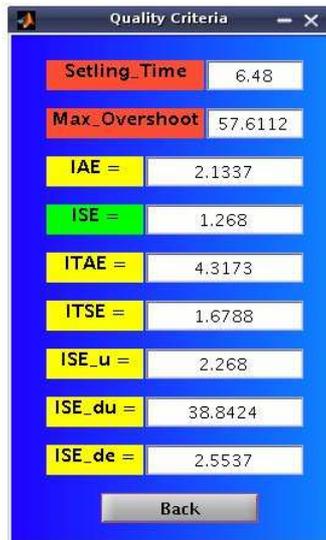


Figure 23: Values of quality criteria

Using the possibilities of window shown in Fig. 20, it is easy to compare several step responses and values of quality criteria reached with different values of set-points, disturbances and constraints on the manipulated variable.

In the new version of PIDTOOL, user can easily change a language of PIDTOOL graphic user interface. Actually, there is a possibility to choose between English and Slovak language (Fig. 24). Other languages can be simply added into the software PIDTOOL.

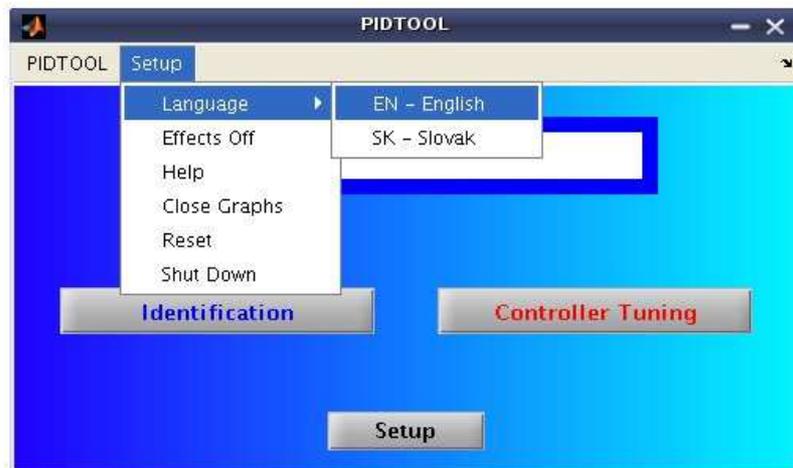


Figure 24: Change the language of PIDTOOL

3. Conclusion

The software PIDTOOL has been developed in the MATLAB – Simulink programming environment. MATLAB Graphical User Interface offers comfortable and visual tool for fast identification, simple PID controller tuning and effective evaluating the quality of control in various conditions. Using the new version of PIDTOOL is very simple. It was tested and used up to now by students at the FCFT in two courses: Process Dynamics and Control and Integrated Control in the Process Industry.

4. Acknowledgment

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 grant 1/0537/10. The support by the grant No. NIL-I-007-d from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism is also highly acknowledged. This project is co-financed from the state budget of the Slovak Republic.

References

- [1] K. Åström and T. Häggglund. *PID controllers: theory, design and tuning*. 2nd Ed. Instrument Society of America, Triangle Park, 1995.
- [2] M. Bakošová and M. Fikar. *Process Control (in Slovak)*. STU Press, Bratislava, 2008.
- [3] J. Mikleš and M. Fikar. *Process Modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg, 2007.
- [4] B. A. Ogunnaike and W. H. Ray. *Process dynamics, modeling and control*. Oxford University Press, New York, 1994.
- [5] A. Vítěček and M. Vítěčková. *Automatic control fundamentals (in Czech)*. TU Ostrava, 2006.
- [6] J. Lošonský. *System for support of control loops (in Slovak)*. Diploma thesis. Trnava: MTF STU v Bratislave, 2006.
- [7] K. H. Weng and C. H. Chang and S. C. Lisheng. *Tuning of PID Controllers Based on Gain and Phase Margin Specifications*. Pergamon, Automatica, Vol. 31. No. 3, pp. 497 – 502, 1995.
- [8] J. A. Shaw. *The PID Control Algorithm [online]*. 2nd edition, version 2.2. <http://www.docstoc.com/docs/20312583/The-PID-Control-Algorithm>, 2006.
- [9] M. Bakošová and L. Čirka and M. Fikar. *Automatic control fundamentals (in Slovak)*. 1st Ed. VYDAVATELSTVO STU, Bratislava, 2003.
- [10] S. Kubík and Z. Kotek. *Automatic control theory (in Czech)*. 1st Ed. SNTL, Praha, 1982.
- [11] B. Šulc – M. Vítěčková. *Theory and praxis of control loop design (in Czech)*. ČVUT Press, Praha, 2004.
- [12] M. Fikar – J. Mikleš. *System identification (in Slovak)*. STU Press, Bratislava, 1999.
- [13] L. Čemanová. *Design of GUI for controller tuning (in Slovak)*. Diploma thesis. FCFT STU in Bratislava, 2007.

- [14] J. Oravec. *Development of software for controller tuning (in Slovak)*. Bachelor project. FCFT STU in Bratislava, 2008.
- [15] J. Oravec. *Development of software for controller tuning (in Slovak)*. Terminal project. FCFT STU in Bratislava, 2009.
- [16] J. Oravec. *Design of a Program System for Controller Tuning (in Slovak)*. Diploma thesis. FCFT STU in Bratislava, 2010.
-

Juraj Oravec
juraj.oravec@stuba.sk

Monika Bakošová
monika.bakosova@stuba.sk