

# SOFTWARE FOR PID CONTROLLER TUNING

*M. Bakošová, J. Oravec, L. Čírka*

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

## Abstract

Software for PID controller tuning PIDTOOL has been developed in the MATLAB–Simulink programming environment using its graphic user interface. The main aim of the PIDTOOL design was creating user friendly tool for simple and fast identification from the step response of the controlled process as well as for simple and fast PID controller tuning and checking the quality of control. The controlled system can be identified from its step response in the form of the 1st order plus time delay transfer function or in the form of the higher order plus time delay transfer function. In the case when the transfer function of the controlled system is known, the controller tuning can be used directly. PID controllers are tuned using various experimental and analytical methods. There is a possibility to choose the type of the tuned controller, such as P, PI, PID or PD. The PIDTOOL also enables to check the quality of control of the designed controller by simulation of the closed-loop step responses. It is possible to compare properties of the designed controllers by generating several step responses with various choices of set-points, disturbances and constraints on the manipulated variable. The quality of control can be judged by calculating several integral quality criteria. The simulation results can serve for decision, whether the designed controller can be used for control of the given process. The PIDTOOL properties determine its using especially for teaching purposes.

## 1 Introduction

The aim of this paper is to present the software for PID controller tuning PIDTOOL. It has been developed in the MATLAB – Simulink programming environment and uses its graphic user interface (GUI). The software has been created as a visual and user friendly tool for PID controller tuning, which are able to control the process in required conditions. For controller tuning, PIDTOOL uses various types of experimental and analytical methods (Bakošová and Fikar 2008, Míkľeš and Fikar 2007, Åström and Hägglund 1995, Ogunnaike and Ray 1994, Víteček and Vítečková 2006). Nowadays, PIDTOOL is used for teaching purposes at the FCFT STU in Bratislava and the communication language is Slovak.

## 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 (Čemanová 2007, Oravec 2008, Oravec 2009). The main aim of the PIDTOOL developing is to create an user friendly tool for simple and fast identification from data achieved from the step response of the controlled process as well as for simple and fast PID controller tuning and checking the quality of control in various required conditions on set-point tracking, disturbance rejection and constraints on the manipulated variable.

PIDTOOL can be used especially for teaching purposes in laboratory exercises in courses oriented on process control. This is also the reason for the choice of the communication language, and Slovak is used up to now. Everybody who wants to tune PID controllers or needs to compare various types of control algorithms with simple PID control can use PIDTOOL. The software PIDTOOL uses MATLAB GUI for simpler manipulation and better visualization, and MATLAB has been used for programming. Fig. 1 shows the basic window of PIDTOOL.

For PID controller tuning, the PIDTOOL asks the controlled process described by the transfer function

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

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

In the case, when these parameters of the controlled system are unknown, there is the possibility to use PIDTOOL for identification from the step response. The Strejc method (Bakošová and Fikar 2008, Mikleš and Fikar 2007) is used for identification from the step response. Identification can be started using the button *Identification* located in the basic window (Fig. 1).

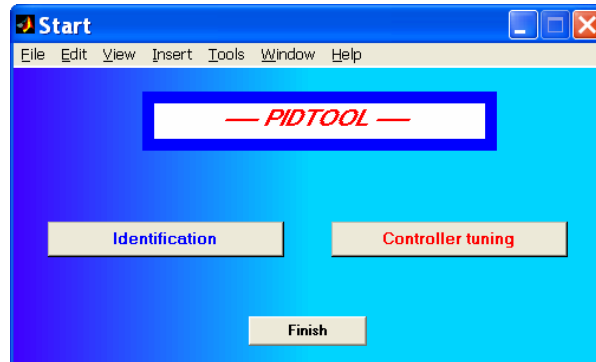


Figure 1: Basic window of PIDTOOL

Using this button opens the new window (Fig. 2), which offers three possibilities: to identify the system from the data which are obtained from the measured and the recorded step response - the button *Step response data*; to identify the transfer function from the data file obtained by measuring of the step response – the button *Data file*; and to identify the system newly from the step response generated by the known transfer function – the button *Transfer function data*.

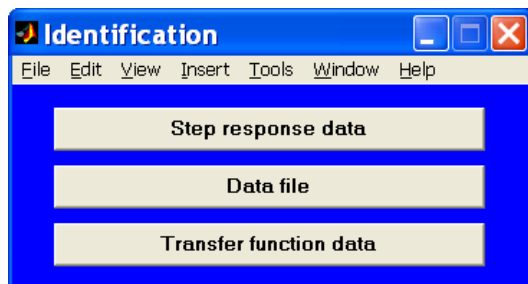


Figure 2: Basic window of identification

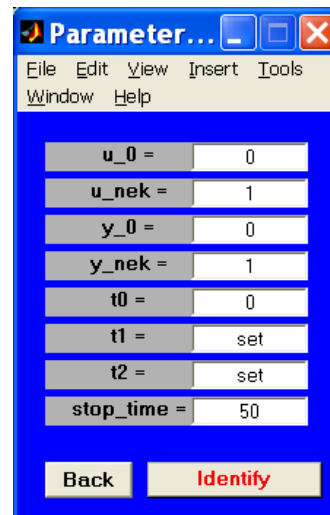


Figure 3: Data inserting for identification

Using the button *Step response data* (Fig. 2) opens the window for data inserting (Fig. 3), where software requires setting of important parameters, which describe the step response. Some of these parameters have the default values for the sake of simple using and some have to be set. Inserted data are stored and they are used as default values in the next identification unless they are changed by the user.

Using the button *Identify* (Fig. 3) opens the new window (Fig. 4), where the user can choose: the identification of the 1<sup>st</sup> order plus time delay transfer function - button *1<sup>st</sup> Order System*; or the

identification of the higher order plus the time delay transfer function - the button *n-th Order System*. The button *Back* returns the user to the previous step.

The identified parameters of the controlled process are shown in a new window (Fig. 5), and the step response obtained using the identified transfer function is displayed at the same time. The button *Compare* (Fig. 5) gives the possibility to compare different step responses. It could be helpful in the case, when the user wants to decide whether it is better to use the 1st order or the higher order transfer function of the controlled process (Fig. 6).

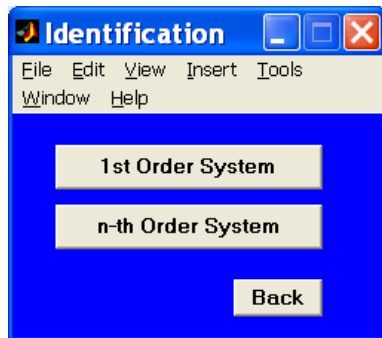


Figure 4: Choice of the identified transfer function

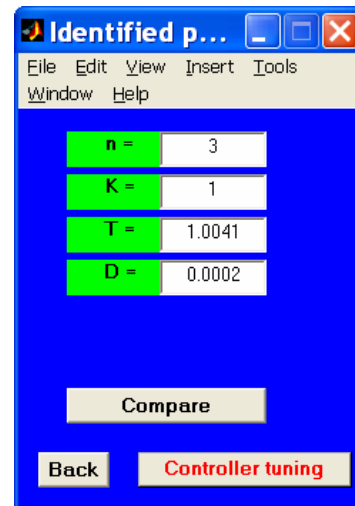


Figure 5: Results of identification

The button *Data file* (Fig. 2) enables to identify the controlled system from the data saved in the data file. The user can find the data file (Fig. 7) at first. Then, the user is asked for inserting the step time and the step change of the input variable used for measurement of the step response of the controlled process (Fig. 8).

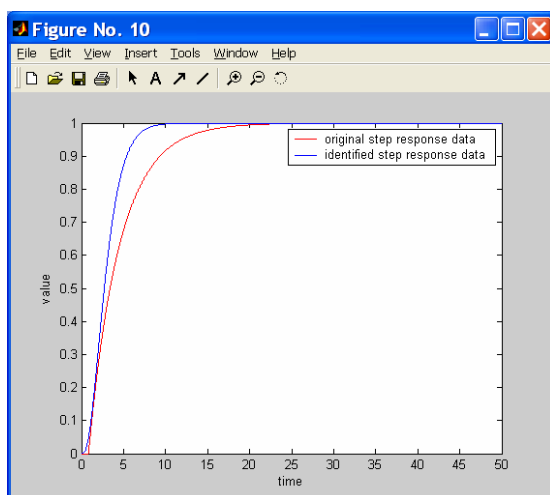


Figure 6: Comparison of the step responses of the 1<sup>st</sup> and the 3<sup>rd</sup> order transfer functions

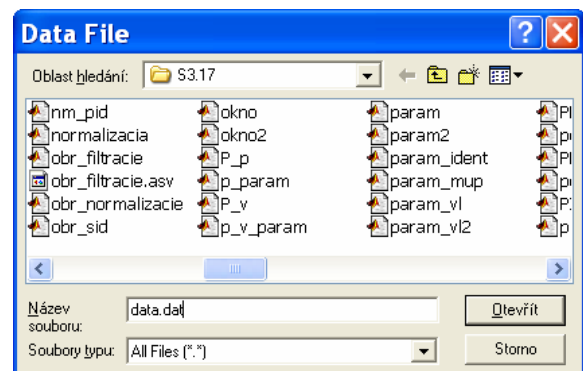


Figure 7: Loading the data file

If the data file contains noisy data, the PIDTOOL enables to run the part *Filtration* (Fig. 9). Here, the filtration conditions can be chosen and the filtration parameters have to be set. If the user is satisfied with the results of filtration (Fig. 10), the filtered data can be saved into the new data file. After successful filtration, the identification can be started.

The possibility *Transfer function data* (Fig. 2) opens the new window (Fig. 11) with the choice of the form of the transfer function describing the identified controlled system. The choice *Transfer function identified by Strejc* is useful in the case, when the controlled process is identified in the form of the higher order plus time delay transfer function and the controller tuning method needs the lower order plus time delay transfer function. This situation can occur e.g. in controller tuning for the Smith predictor, when the time-delay process is controlled and the controller tuning requires the time-delay free model.

If the controlled process is described by transfer function

$$G = \frac{N(s)}{M(s)} e^{-Ds} \quad (2)$$

the choice *General transfer function* can be used to reach the description of the controlled process in the form (1). Identified parameters are shown in a new window (Fig. 12) and the step responses of the original transfer function and the newly identified transfer function are displayed for comparison.



Figure 8: Parameters of the step change

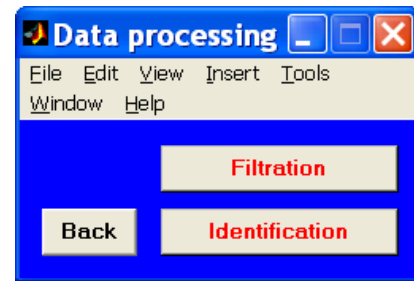


Figure 9: Choice of data processing

To start the PID controller tuning for the known controlled process, the button *Controller tuning* is used (Fig. 1). This choice opens the new window with a choice of the category of the controller tuning method (Fig. 13). There is a possibility to choose an experimental or an analytical controller tuning method.

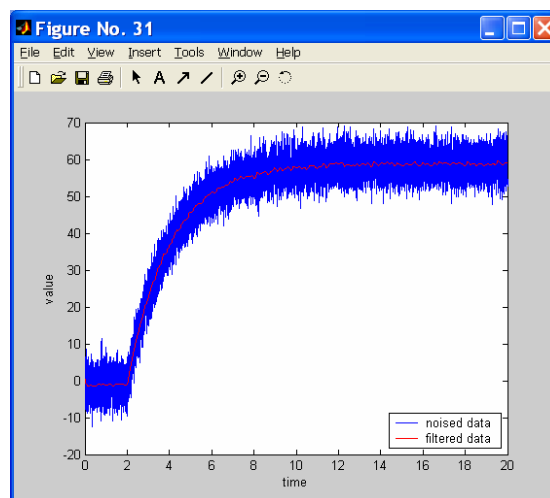


Figure 10: Filtered output

After the choice of the controller tuning method, the software asks parameters describing the controlled process. They can be inserted directly by user. If the part *Identification* was used, these parameters are preset. Then it is possible to choose the type of the tuned controller, such as P, PI, PID

or PD (Fig. 14). Various types of experimental and analytical methods are used for controller tuning (Fig. 15).

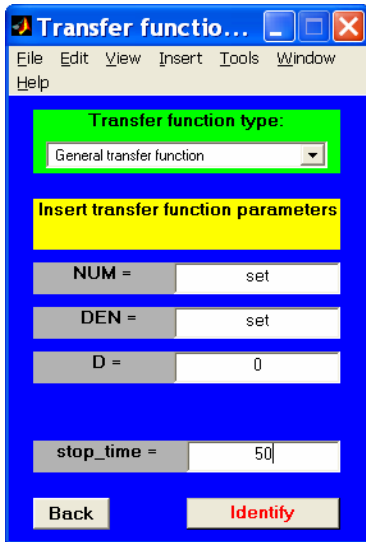


Figure 11: Transfer function data

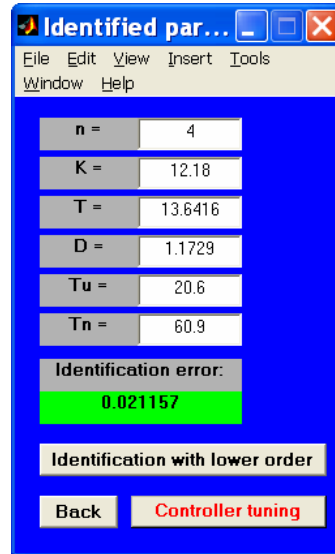


Figure 12: Identified parameters

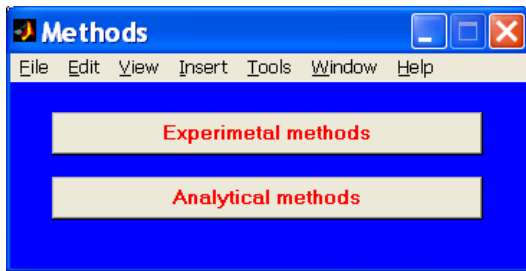


Figure 13: Categories of controller tuning methods

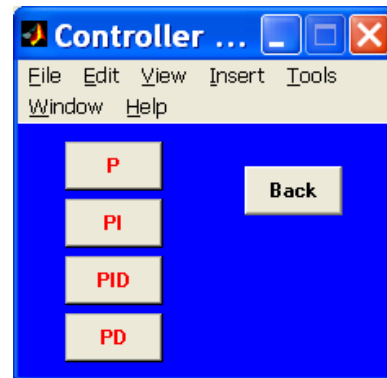


Figure 14: Types of controllers

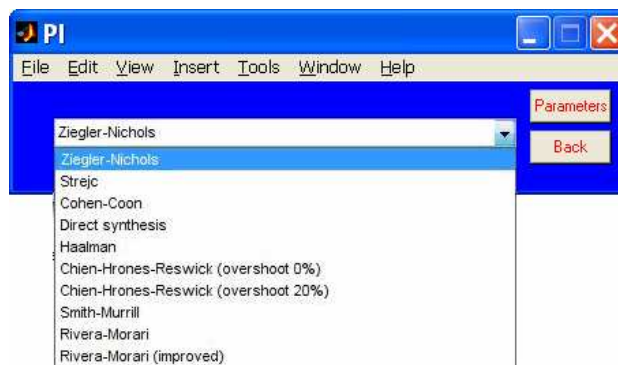


Figure 15: Methods for PI controller tuning

After the choice of the method for controller tuning, the parameters of the controller are calculated. These parameters are shown in the new window (Fig. 16). Fig. 16 shows e.g. following parameters:  $Z_R$  the gain,  $T_I$  the reset time and  $T_D$  the derivative time of the controller.

Setting the parameter  $T_R$  in *PID controller parameters* (Fig. 16) enables to use PID controller with anti-windup (3), (4) (Mikleš and Fikar 2007).

$$u(t) = Z_R e(t) + \frac{Z_R}{T_I} \int_0^t \left( e(t) - \frac{T_I}{Z_R T_R} (u(t) - u_{SAT}(t)) \right) dt + Z_R T_D \frac{de(t)}{dt} \quad (3)$$

$$T_{-R} = \frac{1}{T_R} \quad (4)$$

The window shows also the parameters of the transfer function of the controlled process. These parameters can be changed and the designed controller can be tested in the presence of model uncertainty.

PID controller parameters:		Controlled process:	
Z_R =	5.3566	NUM =	1
T_I =	1.66	DEN =	1.0122 3.0244
T_D =	0.415	D =	0.02231
T_R =	0	Transfer function	
Set-point w(t):		Disturbance r(t):	
W_initial =	1	R_initial =	0
W_final =	1	R_final =	0
W_step_time =	0	R_step_time =	0
Parametre simulácie:		Integral Criteria Parameters:	
Contol_precision =	0.05	A^2 for ISE_u =	1
Stop_time =	50	B^2 for ISE_u` =	1
U_Lower_Boundary	-50	R^2 for ISE_e` =	1
U_Upper_Boundary	50		
Back		Step Response	

Figure 16: Controller and simulation parameters

PIDTOOL enables to judge the quality of control reachable by the designed controller. The set-point tracking and the disturbance rejection can be simulated in the presence of boundaries on the control input. The parameters of the set-point and disturbance step change have pre-set values and they can be modified by user. If the manipulated variable is constrained, PIDTOOL enables to compare the closed-loop step responses and the time behavior of the manipulated variable before and after the saturation of the manipulated variable.

Pressing the button *Step response* (Fig. 16) leads to displaying the closed-loop step responses (Fig. 17). The time behavior of the manipulated variable is shown in Fig. 18. Displaying of the closed-loop control response helps to decide whether the designed controller can be used for control of the real process.

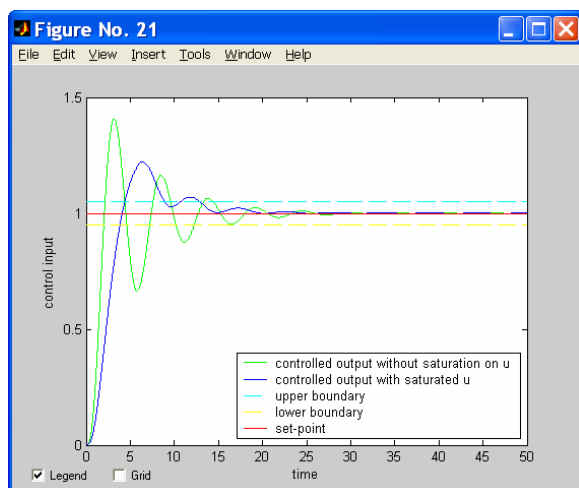


Figure 17: Closed-loop step response

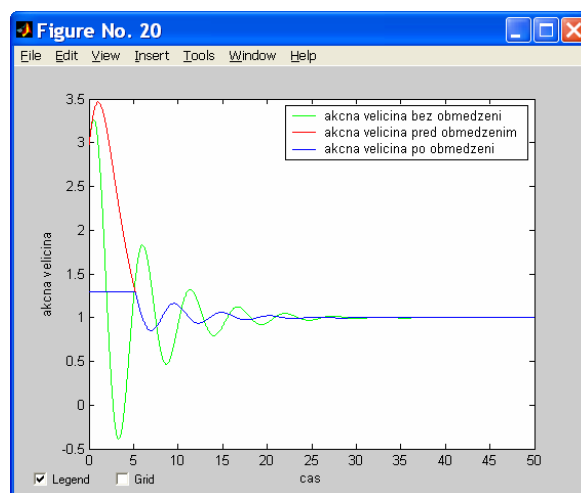


Figure 18: Control input

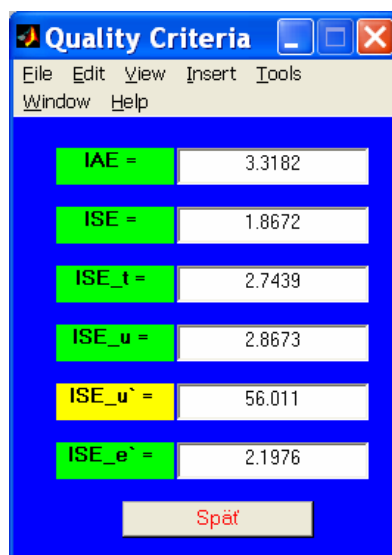


Figure 19: Resulting quality criteria

The quality of control is judged also by calculating several integral quality criteria:  $ISE$  (5),  $IAE$  (6),  $ISE(t)$  (7),  $ISE(e')$  (8),  $ISE(u)$  (9), and  $ISE(u')$  (10) (Fig. 19). The parameters  $R$ ,  $A$ ,  $B$  in (8) - (10) have default values and can be modified by user (Fig. 16).

$$I_{IAE} = \int_0^{\infty} |e(t)| dt \quad (5)$$

$$I_{ISE} = \int_0^{\infty} e^2(t) dt \quad (6)$$

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

$$I_{ISE(e')} = \int_0^{\infty} (e(t)^2 + R^2 e'(t)^2) dt \quad (8)$$

$$I_{ISE(u)} = \int_0^{\infty} (e(t)^2 + A^2 u(t)^2) dt \quad (9)$$

$$I_{ISE(u')} = \int_0^{\infty} (e(t)^2 + B^2 u'(t)^2) dt \quad (10)$$

### 3 Conclusion

PIDTOOL has been developed in the MATLAB 6.5 Simulink 5.0 Release 13 programming environment. MATLAB Graphical User Interface offers comfortable, simple and visual tool for identification, PID controller tuning and enables to compare the quality control in various conditions. Using of PIDTOOL is very simple and 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. PIDTOOL is an open system. It will be further developed and the English version will be completed.

### References

- [1] Åström, K. and T. Hägglund (1995). *PID controllers: theory, design and tuning*. 2nd Ed. Instrument Society of America, Triangle Park.
- [2] Bakošová, M. and M. Fikar (2008). *Process Control* (in Slovak). STU Press, Bratislava.
- [3] Čemanová, L. (2007). *Design of GUI for controller tuning* (in Slovak). Diploma thesis. FCFT STU in Bratislava.
- [4] Mikleš, J. and M. Fikar (2007). *Process Modelling, Identification, and Control*. Springer Verlag, Berlin Heidelberg.
- [5] Ogunnaike, B. A. and W. H. Ray (1994). *Process dynamics, modeling and control*. Oxford University Press, New York.
- [6] Oravec, J. (2008). *Development of software for controller tuning* (in Slovak). Bachelor project. FCFT STU in Bratislava.
- [7] Oravec, J. (2009). *Development of software for controller tuning* (in Slovak). Terminal project. FCFT STU in Bratislava.
- [8] Víteček, A. and M. Vítečková (2006). *Automatic control fundamentals* (in Czech). TU Ostrava.

---

M. Bakošová  
monika.bakosova@stuba.sk

J. Oravec  
juroora@yahoo.com

E. Čírka  
lubos.cirka@stuba.sk