A Matlab Program for Single-Parameter Tuning of PI Controllers

Radek Matušů, and Roman Prokop

***Abstract*—**The main objective of this paper is to present a freely available program for single-parameter tuning of PI controllers. The described software package takes advantage of Matlab and Simulink environment. It incorporates control design based on algebraic approach and tuning of final PI controllers by means of a single positive scalar parameter. The choice of this tuning knob is determined by the user selection of size for the first overshoot of the output signal going from the controlled plant. Although the method is appropriate mainly for stable first-order systems, it can be possibly applied also to the higher-order ones. The capabilities of the program, its advantages and limitations are demonstrated on the set of simulation examples.

***Keywords*—**PI controllers, algebraic control design, single-parameter tuning, graphical user interface, Matlab, Simulink.

I. Introduction

Since vast majority of real-life control applications are reported to utilize Proportional-Integral-Derivative (PID) algorithms, most of them are only of Proportional-Integral (PI)-type in fact [1], [2]. Not surprisingly, the popularity and practical necessity of this kind of easily applicable and reliable controllers entail great research and publication attractiveness during the last decades – e.g. [3] – [8]. In any case, convenient techniques for tuning of PI controllers are still very valuable.

An appropriate ground for simple but effective PI controller design consists in utilization of an algebraic approach which foundations were published in works of Vidyasagar [9] and Kučera [10]. Subsequently, this method using general solutions of Diophantine equations in the ring of proper and (Hurwitz-)stable rational functions (R_{PS}) with possibility of controller tuning by a single scalar positive parameter was elaborated in [11], [12], etc. Moreover, e.g. the papers [13] – [15] presented a simple methodology for proper choice of the tuning parameter on the basis of size of the first overshoot of output signal for stable first-order controlled systems. These tuning recommendations can be applied also to higher-order systems, but the predefined control performance or even stability are not generally guaranteed anymore and must be verified.

Matlab and Simulink represent very effective and widespread environment for many areas of technical computing and simulations. Apart from other things, many researchers use it for implementation of control design methods and their effortless verification – see e.g. [16]. Such programs and toolboxes allow the wide spectrum of users to take advantage of some control methods even without deeper knowledge of them.

This contribution follows e.g. the papers [13] – [15] where the theoretical basics and application potential of the tuning method were provided. This work describes the developed Matlab program which facilitates the controller computation through a simple Graphical User Interface (GUI). The software is freely available for educational and research purposes and can be downloaded from the web page [17]. The basic abilities of the package are illustrated by means of three simulation examples for the first-order, second-order, and fifth order controlled plants.

The paper is the extended version of the conference contribution [18].

The work is organized as follows. In Section II, the basic theoretical background of algebraic control design and PI controller tuning is given. The Section III then describes the program itself and provides the instructions for use. Next, the set of control and simulation examples illustrating the application of the program and both its merits and drawbacks can be found in the extensive Section IV, which consists of three subsections. Finally, Section V offers some conclusion remarks.

II. Theoretical Background

Despite the main topic of the paper, which is intended to describe the developed program and its capabilities, brief theoretical background will be also provided.

The implemented PI controller design method is based on general solutions of Diophantine equations in R_{PS}, expression of all stabilizing controllers though Youla-Kučera parameterization and selection of a suitable controller (for requirements of reference tracking, disturbance rejection, etc.) from this set using conditions of divisibility in the assumed ring (R_{PS}). The principal algebraic ideas are adopted from works of Vidyasagar [9] and Kučera [10]. Subsequently, the control design technique itself was

Radek Matušů, and Roman Prokop are with the Centre for Security, Information and Advanced Technologies (CEBIA – Tech), Faculty of Applied Informatics, Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic. The email contact is: rmatusu@fai.utb.cz.

The work was supported by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089. This assistance is very gratefully acknowledged.

elaborated e.g. in [11], [12]. One of its advantages is that the final controller can be tuned by means of the single scalar parameter $m > 0$.

For example, the assumption of classical feedback control system (with one degree of freedom), step-wise reference signal, no disturbances, and the first-order controlled system in the form of transfer function:

$$G(s) = \frac{b_0}{s + a_0} \tag{1}$$

leads to the PI controller:

$$C(s) = \frac{\tilde{q}_1 s + \tilde{q}_0}{s} \tag{2}$$

with parameters [13] – [15]:

$$\begin{aligned} \tilde{q}_1 &= \frac{2m - a_0}{b_0} \\ \tilde{q}_0 &= \frac{m^2}{b_0} \end{aligned} \tag{3}$$

where $m > 0$ is the tuning parameter.

The possible way of proper choice of this parameter for stable first-order system has been presented in [13] – [15]. The selection of m depends on the required size of first overshoot of the closed-loop output controlled signal. Practically, the m is chosen on the basis the table 1, where the parameter k is the ratio:

$$k = \frac{m}{a_0} \tag{4}$$

and thus the final m is obtained from:

$$m = k a_0 \tag{5}$$

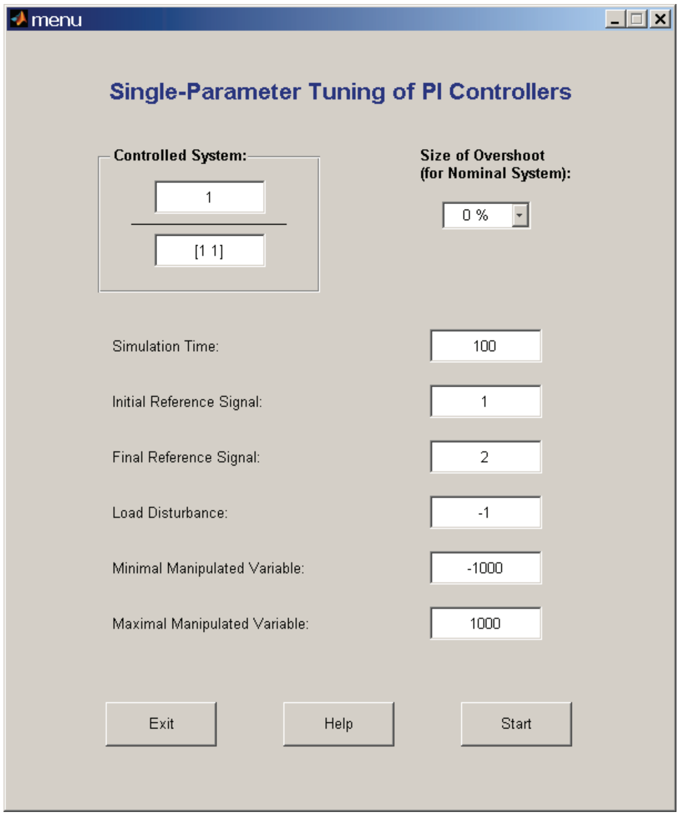
Table 1 relation between k and overshoot [13] – [15]

Overshoot [%]	k
0	1.00
1	1.62
2	1.87
3	2.14
4	2.44
5	2.80
6	3.25
7	3.81
8	4.58
9	5.67
10	7.38

Obviously, this overshoot is observed only if the really controlled plant is the same stable first-order system as the nominal one. If the system is of a higher order and thus some approximation has to be used before the controller calculation, the exactly prescribed overshoot is not obtained anymore. For that reason, the method is suitable primarily for the stable first-order systems and even though it can be applied also to the higher-order ones, the results are not guaranteed generally.

III. Program Description

The software was created in Matlab 7.9.0 (R2009b) and related Simulink environment. It can be freely downloaded from the web page [17] in the form of single “zip” file which contains 4 m-files, 2 fig-files, 1 mdl-file, 1 documentation pdf-file and 1 txt-file with very basic instructions. After unpacking, the program can be launched in Matlab using “start.m” file. The main window of the GUI which will appear is presented in fig. 1.



Single-Parameter Tuning of PI Controllers

Controlled System: 1 / [1 1]

Size of Overshoot (for Nominal System): 0 %

Simulation Time: 100

Initial Reference Signal: 1

Final Reference Signal: 2

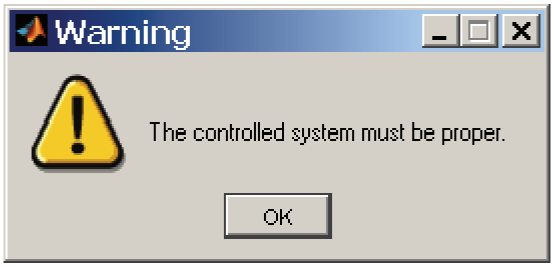
Load Disturbance: -1

Minimal Manipulated Variable: -1000

Maximal Manipulated Variable: 1000

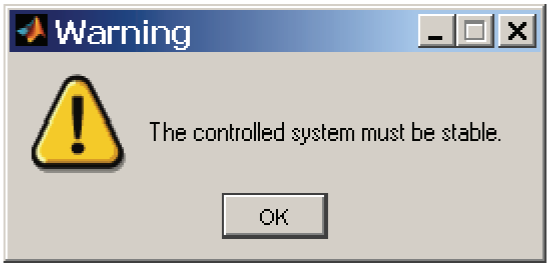
Fig. 1 main window of the program

The first piece of information which must be entered to the program is a transfer function of SISO continuous-time controlled system (upper left corner in fig. 1). It can be defined in a standard Matlab manner, i.e. by means of vector with components representing the polynomial coefficients in descending order both for numerator and denominator. The controlled system must be proper. Moreover, it must be also stable because of applied tuning methodology. If not, a warning will appear (see figs. 2 and 3) and the program will not continue with controller calculation after pressing the “Start” button.



The controlled system must be proper.

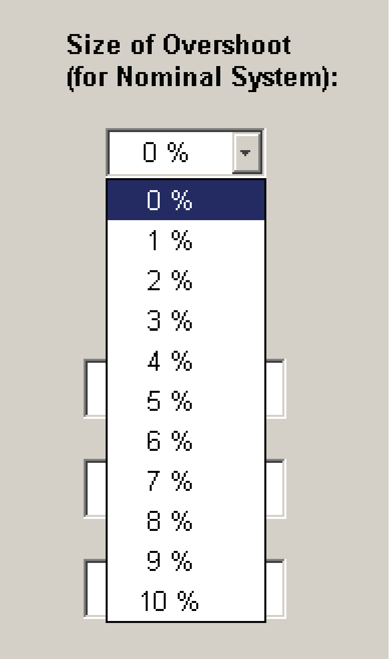
Fig. 2 warning – improper controlled system



The controlled system must be stable.

Fig. 3 warning – unstable controlled system

As can be seen in fig. 4, the pop-up menu in the upper right corner allows choosing the size of the first overshoot of closed-loop output signal from 0% to 10% (in concordance with table 1).



Size of Overshoot (for Nominal System): 0 %

Fig. 4 possible choices for size of first overshoot

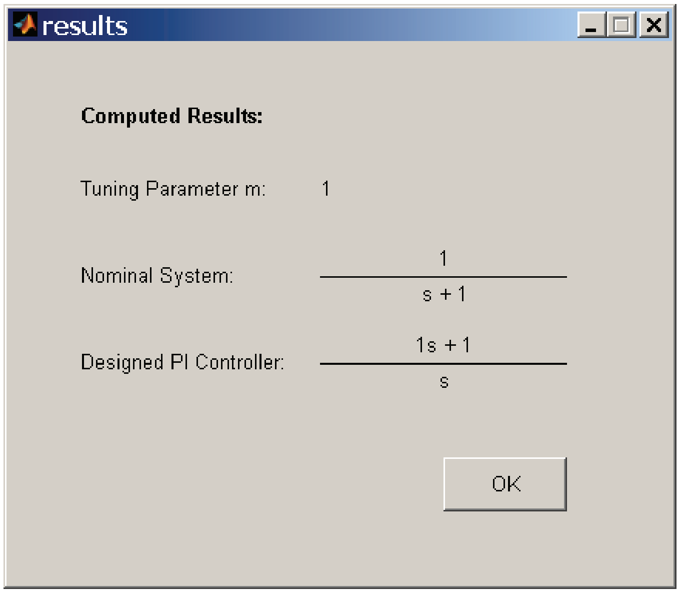
The selection of overshoot determines the size of parameter *m* which is used for controller tuning. However, this overshoot will be really observed only for the nominal system, which is considered as a first order transfer function with relative order one. If the controlled system is of a higher order, the nominal one is obtained by the simplest approximation using neglect of higher powers in numerator and denominator polynomials.

Further, a user can adjust several basic parameters of simulation (fig. 1), such as:

- Simulation time
- Initial reference signal (for the first third of simulation time)
- Final reference signal (for the other two thirds of simulation time)
- Load disturbance (injected into the input of the controlled system during the last third of simulation)
- Minimal manipulated variable (bottom saturation for controller output)
- Maximal manipulated variable (upper saturation for controller output) – the saturations do not influence the control design itself, but only simulation.

Finally, after setting all the parameters, the “Start” button executes the controller calculation, opens and runs the Simulink scheme (fig. 6) and displays the windows with results (fig. 5), including:

- Size of tuning parameter *m*
- First-order nominal system (potentially obtained by the simple approximation of the really controlled system)
- Final continuous-time PI controller



Computed Results:

Tuning Parameter m: 1

Nominal System: $\frac{1}{s+1}$

Designed PI Controller: $\frac{1s+1}{s}$

Fig. 5 display of computed results

The control results (output signal + manipulated variable) are automatically simulated and visualized using scheme from fig. 6.

As it has been already mentioned, the predefined size of overshoot will not be observed if the controlled system has to be approximated. Furthermore, even the stability of the closed-loop control system is not guaranteed in such case. The user is warned immediately after the simulation by window as shown in fig. 7.

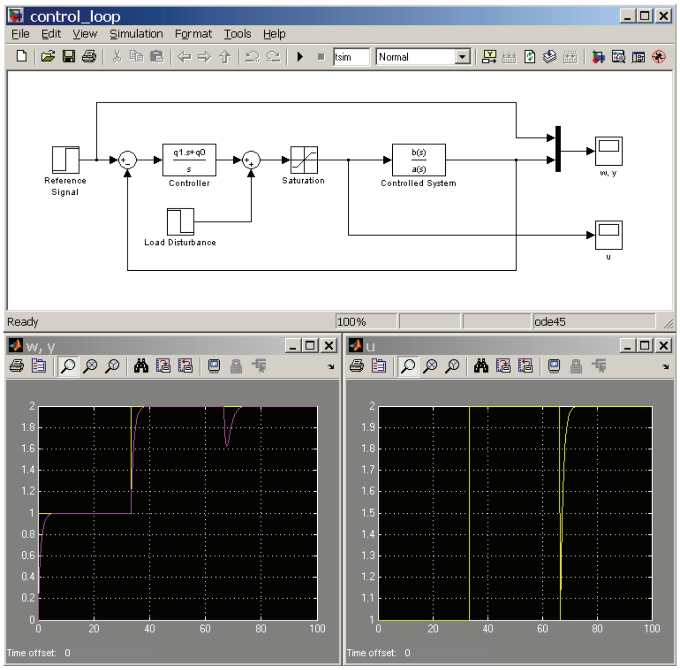
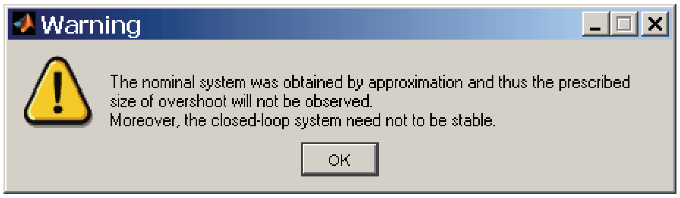


Fig. 6 Simulink scheme with control results



The nominal system was obtained by approximation and thus the prescribed size of overshoot will not be observed.
Moreover, the closed-loop system need not to be stable.

Fig. 7 warning – controlled system approximation

IV. Illustrative Examples

The capabilities of the program are presented on the following three examples. There were supposed the simulation time 60 (e.g. seconds if the controlled plant time constants are assumed in seconds), the step change of reference signal from 1 to 2, the step load disturbance -0.5, and unlimited manipulated variables for all cases.

A. First-Order Plant

First, the controlled plant is assumed to be given by the stable first-order transfer function:

$$G(s)=\frac{5}{2s+1} \tag{6}$$

Obviously, the nominal system equals to the previous function (6), but the program gives it in the form with monic polynomial:

$$G_N(s)=\frac{2.5}{s+0.5} \tag{7}$$

In the first instance, the choice of output controlled signal without overshoot (0% size of overshoot) results in the tuning parameter $m=0.5$ and subsequently in the PI controller:

$$C(s)=\frac{0.2s+0.1}{s} \tag{8}$$

The closed-loop control output obtained from the program is shown in fig. 8 and the corresponding manipulated variable in fig. 9. Evidently, the output signal really has no overshoot.

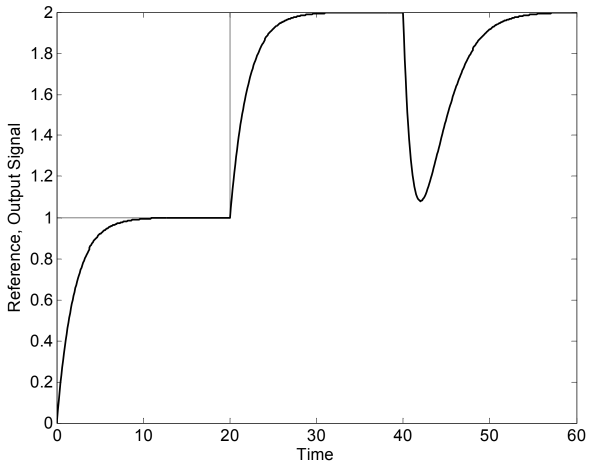


Fig. 8 output signal – plant (6) and controller (8)

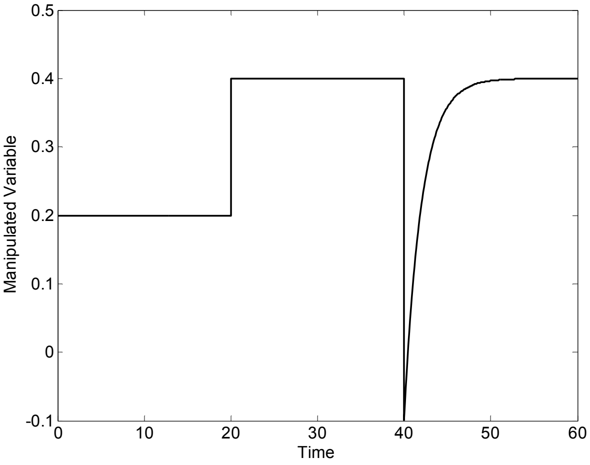


Fig. 9 manipulated variable – plant (6) and controller (8)

Next, the selection of 2% size of overshoot leads to the tuning parameter $m=0.935$ and consequently to the PI controller:

$$C(s)=\frac{0.548s+0.34969}{s} \tag{9}$$

The output signal is depicted in fig. 10 and the corresponding manipulated variable in fig. 11. The closer view to the output signal overshoot after the first step change of the reference signal is depicted in fig. 12. As can be seen, the requested 2% value is kept.

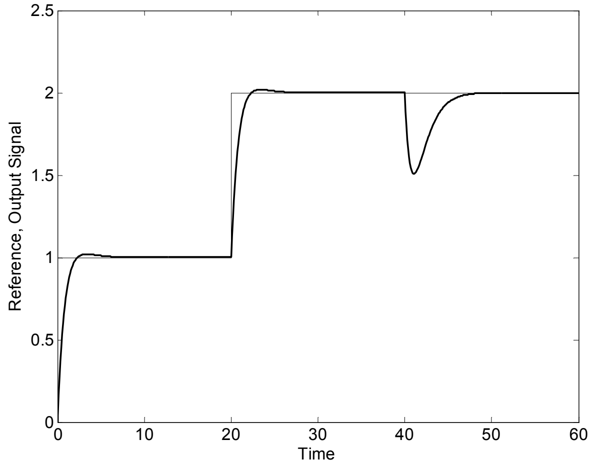


Fig. 10 output signal – plant (6) and controller (9)

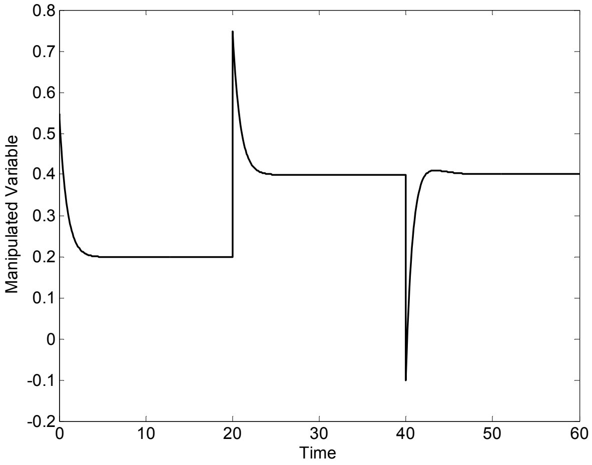


Fig. 11 manipulated variable – plant (6) and controller (9)

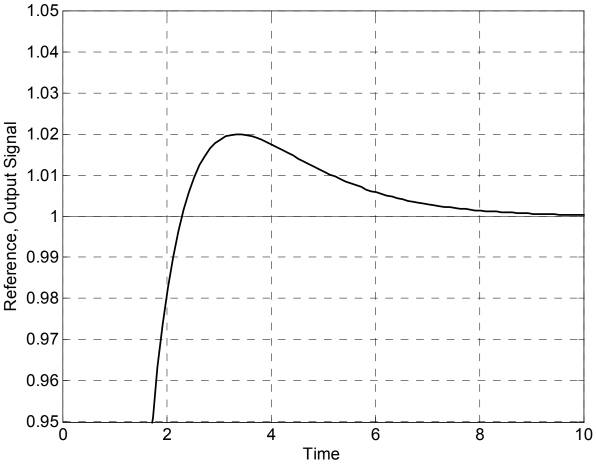


Fig. 12 detailed view of output signal – plant (6) and controller (9)

Finally, the first output signal with overshoot of 5% was chosen. The related tuning parameter can be easily derived as $m = 1.4$ and the calculated PI controller has the transfer function:

$$C(s) = \frac{0.92s + 0.784}{s} \tag{10}$$

The output signal of the closed control loop is shown in fig. 13.

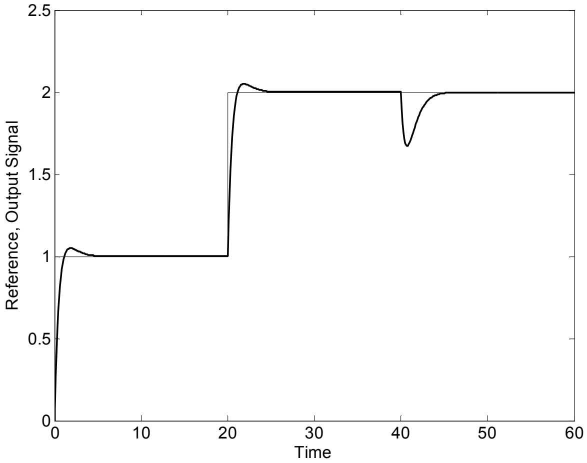


Fig. 13 output signal – plant (6) and controller (10)

Then, fig. 14 depicts the corresponding manipulated variable.

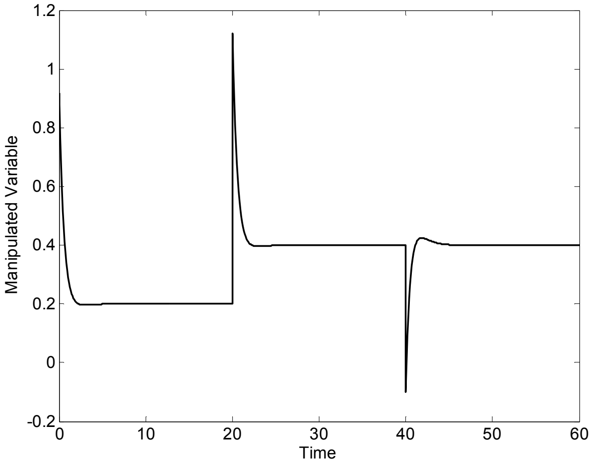


Fig. 14 manipulated variable – plant (6) and controller (10)

Lastly, the zoomed overshoot of the output signal after the first step change of reference signal is presented in fig. 15, respectively. Again, the requested 5% size of overshoot is really obtained.

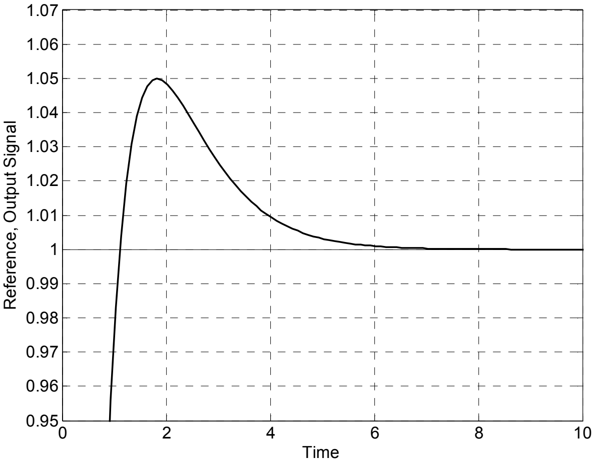


Fig. 15 detailed view of output signal – plant (6) and controller (10)

B. Second-Order Plant

The second example deals with the second-order system described by:

$$G(s)=\frac{5}{(1.5s+1)(0.5s+1)}=\frac{5}{0.75s^2+2s+1} \tag{11}$$

Since the controlled system transfer function is not of the first order with relative order one, the program approximates it simply by neglecting the second-order term in denominator, i.e. the nominal system used for the controller design is given by the same transfer function (7) as in the previous example. Thus, the assumption of the same 0% overshoot naturally results in the same tuning parameter $m=0.5$ and the same PI controller (8). The final control behaviour is visualized in figs. 16 (output signal) and 17 (manipulated variable). As can be seen, even though the control results were not guaranteed a priori for the second order controlled plant (11), the output signal without the overshoot is obtained.

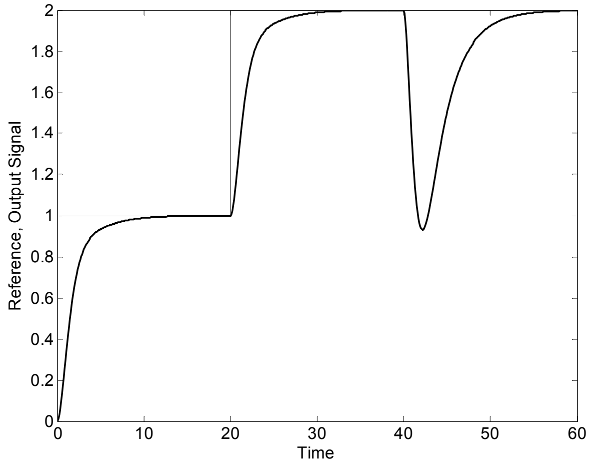


Fig. 16 output signal – plant (11) and controller (8)

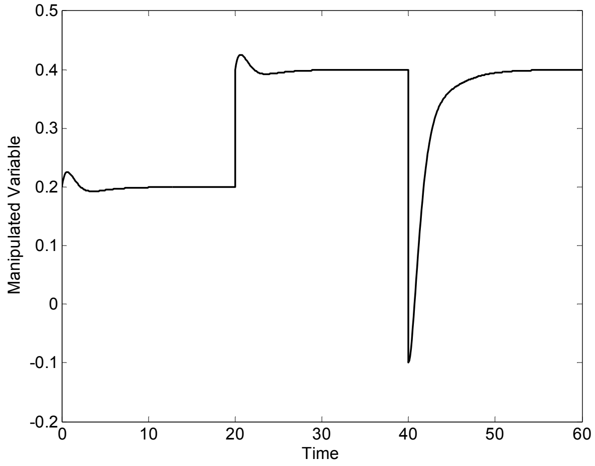


Fig. 17 manipulated variable – plant (11) and controller (8)

Next, suppose 2% overshoot. As in the previous example *A*, it leads to the same tuning parameter $m=0.935$ and to the PI regulator (9). The simulated output signal is presented in fig. 18 and manipulated variable in fig. 19.

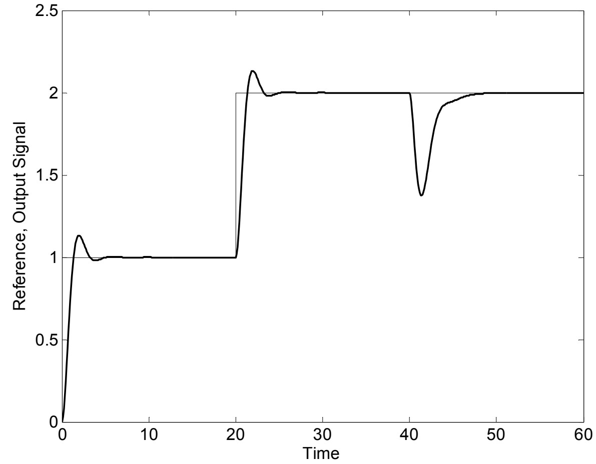


Fig. 18 output signal – plant (11) and controller (9)

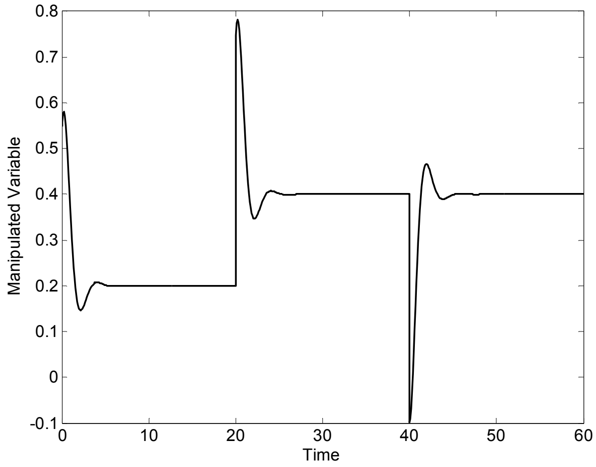


Fig. 19 manipulated variable – plant (11) and controller (9)

Unfortunately, due to the approximation, the nominally prescribed overshoot is not observed anymore. In fact, it is of about 13% as can be noticed from the detailed view in fig. 20. Nevertheless, the closed-loop system has remained stable.

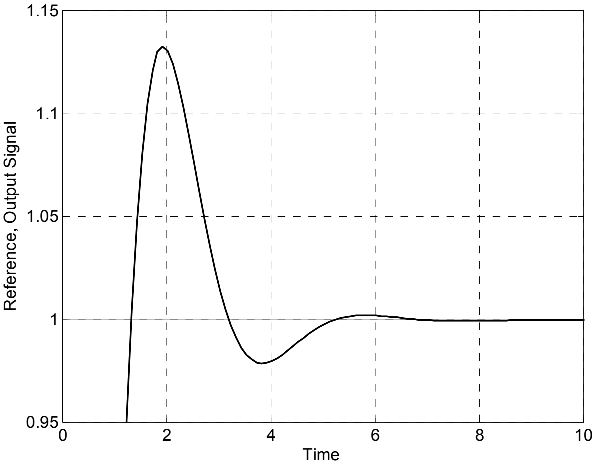


Fig. 20 detailed view of output signal – plant (11) and controller (9)

C. Fifth-Order Plant

In the final example, the fifth-order system given by transfer function:

$$G(s)=\frac{1}{\left(0.3s+1\right)^5}= \tag{11}$$

$$=\frac{1}{0.00243s^5+0.0405s^4+0.27s^3+0.9s^2+1.5s+1}$$

is going to be controlled. The nominal system is then:

$$G_N(s)=\frac{0.\overline{6}}{s+0.\overline{6}} \tag{12}$$

Analogically to the previous cases, the assumption of no-overshoot output signal means the tuning parameter $m=0.\overline{6}$. So, the computed PI controller is:

$$C(s)=\frac{s+0.\overline{6}}{s} \tag{13}$$

The obtained control results are provided in fig. 21 (output signal), fig. 22 (manipulated variable), and fig. 23 (zoomed peak of the output signal). As can be expected, the required 0% overshoot has not been observed. Due to the approximation of the fifth order controlled system, the maximal overshoot reaches almost 9%, but the closed-loop system is stable.

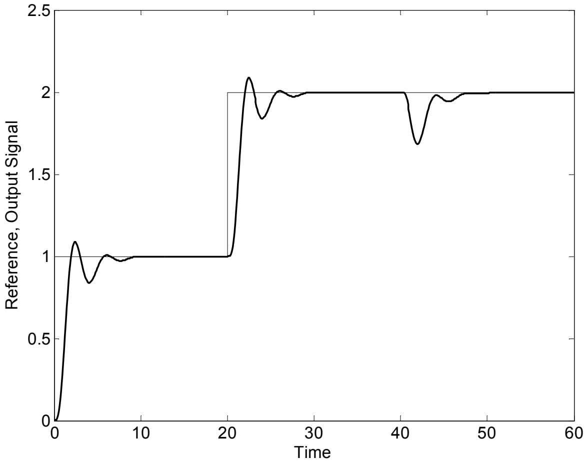


Fig. 21 output signal – plant (11) and controller (9)

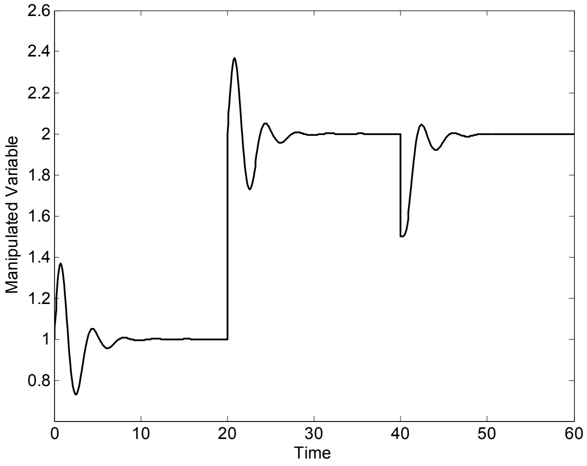


Fig. 22 manipulated variable – plant (11) and controller (9)

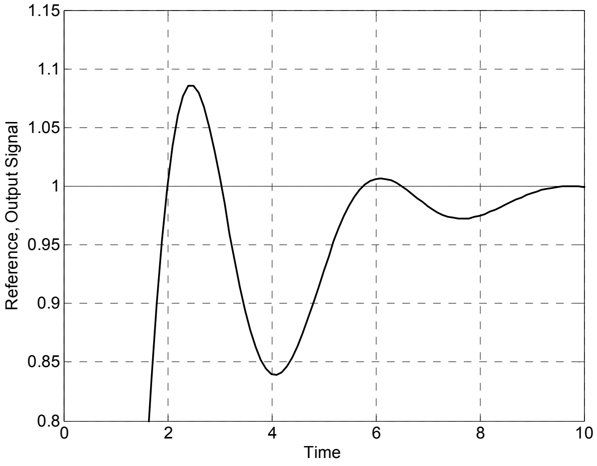


Fig. 23 detailed view of output signal – plant (11) and controller (9)

V. Conclusion

The main aim of this paper has been to present a simple Matlab + Simulink program for single-parameter tuning of PI controllers and demonstrate its capabilities through the set of three simulation examples. The software is freely available both for educational and research purposes and its potential user is not required to have deep knowledge on the applied method.

References

[1] K. J. Åström, and T. Hägglund, *PID Controllers: Theory, Design and Tuning*. Research Triangle Park, North Carolina, USA: Instrument Society of America, 1995.

[2] A. O'Dwyer, *Handbook of PI and PID Controller Tuning Rules*. London, UK: Imperial College Press, 2003.

[3] L. Pekař, and R. Prokop, "Stabilization of a Delayed System by a Proportional Controller" *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 4, no. 4, pp. 282-290, 2010.

[4] T. Emami, and J. M. Watkins, "A Unified Approach for Sensitivity Design of PID Controllers in the Frequency Domain" *WSEAS Transactions on Systems and Control*, vol. 4, no. 5, pp. 221-231, 2009.

[5] T. Emami, and J. M. Watkins, "Robust Performance Characterization of PID Controllers in the Frequency Domain" *WSEAS Transactions on Systems and Control*, vol. 4, no. 5, pp. 232-242, 2009.

[6] Md. S. Islam, N. Amin, M. Zaman, and M. S. Bhuyan, "Fuzzy based PID Controller using VHDL for Transportation Application" *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 2, no. 2, pp. 143-147, 2008.

[7] T. N. L. Vu, and M. Lee, "Multi-loop PI Controller Design for Enhanced Disturbance Rejection in Multi-delay Processes" *International Journal of Mathematics and Computers in Simulation*, vol. 2, no. 1, pp. 89-94, 2008.

[8] R. Matušů, and R. Prokop, "Experimental Verification of Design Methods for Conventional PI/PID Controllers" *WSEAS Transactions on Systems and Control*, vol. 5, no. 5, pp. 269- 280, 2010.

[9] M. Vidyasagar, *Control system synthesis: a factorization approach*. Cambridge, Massachusetts, USA: MIT Press, 1985.

[10] V. Kučera, "Diophantine equations in control – A survey" *Automatica*, vol. 29, no. 6, pp. 1361-1375, 1993.

[11] R. Prokop, and J. P. Corriou, "Design and analysis of simple robust controllers" *International Journal of Control*, vol. 66, no. 6, pp. 905-921, 1997.

[12] R. Prokop, P. Husták, and Z. Prokopová, "Simple robust controllers: Design, tuning and analysis" presented at the 15th IFAC World Congress, Barcelona, Spain, 2002.

[13] R. Matušů, R. Prokop, and J. Korbel, "Analysis and tuning of PI controllers designed through algebraic tools" presented at the 6th Asian Control Conference, Bali, Indonesia, 2006.

[14] R. Matušů, and R. Prokop, "Single-parameter tuning of PI controllers: from theory to practice" presented at the 17th IFAC World Congress, Seoul, Korea, 2008.

[15] R. Matušů, and R. Prokop, "Single-parameter tuning of PI controllers: Theory and application" *Journal of the Franklin Institute*, vol. 348, no. 8, pp. 2059-2071, 2011.

[16] J. Vojtěšek, and P. Dostál, "Use of MATLAB environment for simulation and control of CSTR" *International Journal of Mathematics and Computers in Simulations*, vol. 5, no. 6, pp. 528-535, 2011.

[17] R. Matušů, and R. Prokop, "Single-Parameter Tuning of PI Controllers", Available from: http://zamestnanci.fai.utb.cz/~matusu/PItuning.zip.

[18] R. Matušů, R. Prokop, "Single-Parameter Tuning of PI Controllers: A Matlab Program" presented at the 16th WSEAS International Conference on Mathematical and Computational Methods in Science and Engineering, Malta, 2012.

Radek Matušů was born in Zlín, Czech Republic in 1978. He is a Researcher at Faculty of Applied Informatics of Tomas Bata University in Zlín, Czech Republic. He graduated from Faculty of Technology of the same university with an MSc in Automation and Control Engineering in 2002 and he received a PhD in Technical Cybernetics from Faculty of Applied Informatics in 2007. He worked as a Lecturer from 2004 to 2006. The main fields of his professional interest include robust systems and application of algebraic methods to control design.

Roman Prokop, born in 1952, is a Vice-Dean and a Full Professor at Faculty of Applied Informatics of Tomas Bata University in Zlín, Czech Republic. He graduated from Czech Technical University in Prague in 1976 and received a PhD from Slovak University of Technology in Bratislava in 1983. He was an Associate Professor from 1996 and a Full Professor from 2004. He aims his pedagogical and research work to automatic control theory, algebraic methods in control design and optimization. The main interests of the latest period are uncertain and robust systems, autotuning of controllers and time-delay systems.