Abstract: - This contribution presents a freely available Matlab program for single-parameter tuning of PI controllers. The described software package incorporates control design based on algebraic approach and tuning of final PI controllers by means of a single parameter. The selection of this tuning knob is determined by the user choice of size for the first overshoot of output signal. Although the method is suitable mainly for stable first-order systems, it can be possibly applied also to the higher-order ones. The capabilities of the program are demonstrated on two simulation examples.

Key-Words: - PI Controllers, Algebraic Control Design, Single-Parameter Tuning, Graphical User Interface, Matlab, Simulink

1 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]. This means that 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 [3], [4]. Subsequently, this method using general solutions of Diophantine equations in the ring of proper and (Hurwitz-)stable rational functions (RPS) with possibility of controller tuning by a single positive parameter was elaborated in [5], [6], etc. Moreover, e.g. the works [7], [8], [9] 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 performance or even stability are not generally guaranteed anymore and must be verified.

This contribution follows e.g. the papers [7], [8], [9] 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 [10]. The basic abilities of the package are illustrated by means of simulation examples for first and second-order controlled plants.

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

The implemented PI controller design method is based on general solutions of Diophantine equations in RPS, expression of all stabilizing controllers though Youla-Kučera parameterization and selection of a suitable controller (for reference tracking, disturbance rejection, etc.) from this set using conditions of divisibility in the assumed ring (RPS). The principal algebraic ideas are adopted from works of Vidyasagar [3] and Kučera [4]. Subsequently, the control design technique itself was elaborated e.g. in [5], [6]. One of its advantages is that the final controller can be tuned by means of the single parameter $m > 0$. The possible way of proper choice of this parameter for stable first-order system has been presented in [7], [8], [9]. The selection of $m$ depends on the required size of first overshoot of the closed-loop output controlled signal. 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.

3 Program Description
The software was created in Matlab 7.9.0 (R2009b) + Simulink environment. It can be freely downloaded from the web page [10] 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 shown in Fig. 1.

Fig. 2: Warning – improper controlled system

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 first overshoot of closed-loop output signal from 0% to 10%.

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

![Fig. 6: Simulink scheme with control results](image)

![Fig. 7: Warning – controlled system approximation](image)

**4 Illustrative Examples**

The capabilities of the program are presented on the following two 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 both cases.

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

\[
G(s) = \frac{5}{2s+1}
\]  

(1)

Obviously, the nominal system equals to the function (1), but the program gives it in the form with monic polynomial:

\[
G_n(s) = \frac{2.5}{s+0.5}
\]  

(2)

For example, the selection of 2% size of overshoot leads to the tuning parameter \( m = 0.935 \) and consequently to the PI controller:

\[
C(s) = \frac{0.548s + 0.34969}{s}
\]  

(3)
The closed-loop control output obtained from the program is shown in Fig. 8 and the corresponding manipulated variable in Fig. 9. The closer view to the output signal overshoot after the first step change of the reference signal is depicted in Fig. 10. As can be seen, the requested 2% value is kept.

The second example deals with the second-order system:

$$G(s) = \frac{5}{(1.5s+1)(0.5s+1)} = \frac{5}{0.75s^2 + 2s + 1} \quad (4)$$

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 the same as in the previous case (2). Thus, the assumption of the same 2% overshoot naturally results in the same tuning parameter $m = 0.935$ and the same PI controller (3). The final control behaviour is visualized in Figs. 11 (output signal) and 12 (manipulated variable). 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 detail in Fig. 13. Nevertheless, the closed-loop system has remained stable.
Fig. 13: Detailed view of output signal – plant (4) and controller (3)

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

Acknowledgements:
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.

References: