DE GRUYTER OPEN

Journal of ELECTRICAL ENGINEERING, VOL 67 (2016), NO3, 160–168

PID CONTROLLER DESIGN BASED ON GLOBAL OPTIMIZATION TECHNIQUE WITH ADDITIONAL CONSTRAINTS

Stepan Ozana — Tomas Docekal *

This paper deals with design of PID controller with the use of methods of global optimization implemented in Matlab environment and Optimization Toolbox. It is based on minimization of a chosen integral criterion with respect to additional requirements on control quality such as overshoot, phase margin and limits for manipulated value. The objective function also respects user-defined weigh coefficients for its particular terms for a different penalization of individual requirements that often clash each other such as for example overshoot and phase margin. The described solution is designated for continuous linear time-invariant static systems up to 4th order and thus efficient for the most of real control processes in practice.

K e y w o r d s: PID, control design, control system, global optimization, Matlab

1 INTRODUCTION

Control design plays a significant part in control engineering and its application in various industrial fields. In general scale it is one of the most discussed issues all over the world on technical conferences regarding automation, technical cybernetic and control design, always representing main topics in conference call for papers. The methodology has been developed since 50's when classical methods started to spread and apply for automation of various industrial plants. Although later on there came the era of so called modern control theory based on special algorithms such as for example LQ, robust, predictive or adaptive control, the development of neither classical methods nor modern methods has not been finished despite the vast number of books, scientific papers and effort put into the idea of finding a controller fulfilling certain criteria regarding control quality, allowing easy implementation and the highest versatility of its use as possible. The latter mentioned idea of fulfilling and merging rigorous requirements on controller performance and general usability of the designed controller is one of the fundamental keystones of the paper. It is supported by a wide use of computational technique that keeps developing at a high pace in terms of the benchmarked performance. As the algorithms described in this paper have iteration character using special functions of Optimization Toolbox determined for global nonlinear optimization problems, they pose a challenging problem even for the latest modern PC working stations. On the other hand, this is not a restrictive factor for use of these methods as the algorithms can also be adapted for mobile devices and Matlab-like software tools.

2 PROBLEM DEFINITION

2.1 Motivation and Objectives

The main motivation factors for development of presented solution lie in

- *The need for the best controller performance as possible in terms of control quality*. Most of the classical methods for PID control design lead to the resulting controller with given parameters and properties and satisfying general requirements. If one of more properties does not meet other possible additional requirements, the controller must be modified subsequently. For example, Ziegler-Nichols tuning method usually finds a controller having quite a high overshoot compared to other methods.

- *The need for the most universal usability as possible*. Most of the classical methods for PID control design suffer from usability restrictions. This may be caused by a lot of reasons such as character of the plant (regarding zeros and poles character and position) or its order. Moreover, in case of the methods based on analytical solution, there might be a lack of equations (number of usable equations less than number of parameters to be computed), analytical solution might not exist, complex controller's coefficients may be found or no solution may be found at all. For example the use of Modulus Optimum Method requires setting up a certain number of equations (according the type of controller) for a given plant, which may be impossible to meet in some cases (this is connected to system order). Other typical example may be using of the optimum coefficients for the ITAE criterion where the closed-loop transfer function must follow the exact form prescribed by the table; this is impossible to achieve in most of the cases, so the use of this method is limited a lot.

* VSB-Technical University of Ostrava, Department of Cybernetics and Biomedical Engineering, 17. listopadu 15/2172, Ostrava-Poruba, 708 33, Czech Republic, stepan.ozana@vsb.cz

DOI: 10.1515/jee-2016-0023, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

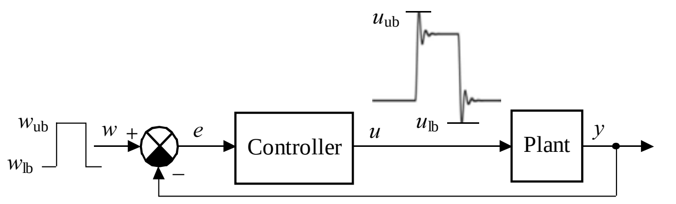

Fig. 1. Block scheme of typical application of the proposed solution

The main objectives set up by the authors can be summed up as follows (see also subsection 3.1).

• Achievement of method of PID design with effective application of integral criteria (ISE, IAE, ITAE, ITSE) and a numerical approach instead of analytical computation techniques

• Possibility to consider, specify and weigh additional requirements on the controller.

So far, the solution has been designed for a certain class of the systems as described in subsequent subsection.

2.2 Classification

The proposed solution has been designed for continuous linear time-invariant static systems up to 4th order. The classification of the systems in terms of mathematical description with the use of Laplace transform for particular cases including classification according types of the poles of the systems is as follows:

- 1st order system

$$G(s) = \frac{k}{(s-p_1)}\,, \quad k \in \mathbb{R}\,,\ p_1 \in \mathbb{R}\,,\ p_1 < 0\,.$$

- 2nd order system

$$G(s) = \frac{k}{(s-p_1)(s-p_2)}\,, \quad k \in \mathbb{R}\,,$$
$$p_n = \mathrm{Re}\{p_n\} + j\,\mathrm{Im}\{p_n\} \in \mathbb{C}\,,$$
$$\mathrm{Re}\{p_n\} < 0 \text{ for } n = 1,2\,.$$

Distinguishing in terms of position of the poles:

– one pair of complex poles, two different single real poles, one double-pole

- 3rd order system

$$G(s) = \frac{k}{(s-p_1)(s-p_2)(s-p_3)}\,, \quad k \in \mathbb{R}\,,$$
$$p_n = \mathrm{Re}\{p_n\} + j\,\mathrm{Im}\{p_n\}\,,$$
$$\mathrm{Re}\,p_n < 0 \text{ for } n = 1,2,3\,.$$

Distinguishing in terms of position of the poles:

– one pair of complex poles + one single real pole,

– three real poles:
a) three different single poles, b) one double-pole + one different single pole, c) triple-pole.

- 4th order system

$$G(s) = \frac{k}{(s-p_1)(s-p_2)(s-p_3)(s-p_4)}\,, \quad k \in \mathbb{R}\,,$$
$$p_n = \mathrm{Re}\{p_n\} + j\,\mathrm{Im}\{p_n\}\,,$$
$$\mathrm{Re}\,p_n < 0 \text{ for } n = 1,\dots,4\,.$$

Distinguishing in terms of position of the poles:

– two pairs of complex poles,

– one pair of complex poles + two single real poles, a) double-pole, b) two different single poles,

– four real poles, a) all different single poles, b) two double-pole + different double-pole, c) triple-pole+single pole, d) quadruple-pole.

2.3 Typical Example of Application

The proposed algorithm is effective when looking for controller working for the plant in a given operating point with known possible technological bound values for the setpoint and manipulated value according Fig. 1. Overshoot of the output signal, phase margin and chosen integral criterion are specified by a user.

For example, a speed control of DC motor by duty cycle of PWM signal $u \in \langle 0,1\rangle$ can be considered at the setpoint $w = 500$ rpm. Let us suppose manipulated value (duty cycle) $u = 0.5$ in a steady-state operating point corresponding to 500 rpm and time constants $T_1 = 5$ s, $T_2 = 20$ s are experimentally identified. Then it is possible to describe the model of DC motor as the 2$^{\mathrm{nd}}$ order system $G(s) = \frac{1000}{(5s+1)(20s+1)}$. Let us suppose admissible range of the setpoint between 400 and 600 rpm. Applying ITAE criterion and standard requirements on control quality, phase margin 45 %, overshoot 15 % lead to the following input parameters of *mainGlobal* function (see section 3.2): $idx = 4$, $peak = 15$, $phl = 30$, $phh = 60$, $wlb = 400$, $wub = 600$, $ulb = 0$, $uub = 1$.

2.4 Reference Solutions

This section is focused on comparison of three reference solutions concerning similar topic with the proposed solution while common features will be pointed out.

The proposed solution complies with the following statements in the reference solutions performed by other authors regarding motivation of this paper:

• in case of nonlinearities in the plant or actuator saturation, the existing classical tuning methods can no longer be used [1],

• for the time varying plant model, it is extremely difficult, if not impossible, to design satisfactory PID controllers with traditional approaches [1].

The reference [1] mentions that overshoot constraints can also be used in the interface in case of presence of actuator saturation. However, it only seems to take the saturation into account but does not offer a remedy action to avoid occurrence of saturation at all. The algorithm proposed in this paper handles saturation in a different way: having known the technological limits for manipulated value the algorithm takes them into account and designs such a controller that guarantees the manipulated value within a specified interval.

The reference [7] declares that ISE-based minimization often results in a relatively oscillatory step response since the large errors which must occur for small time contribute significantly to the performance index. A lot of tests performed during the development of the proposed method described in section 3 of this paper confirm that statement. This is one of the reasons why algorithms based on minimization of integral performance indices, especially the ones using a global minimization, should always consider overshoot limit, like the proposed solution in this paper.

The reference [2] is focused on ITAE performance index only. The solution is rather complex, it shows some common signs that also apply for the proposed solution in this paper. However, that approach applies a local minima optimization problem with the use of $fminsearch$ function. Before the case studies described in sections 4.2 and 4.4, the authors of this paper also tried to use this approach and carried out tests that proved certain limitations. The most crucial issue is the initial point representing an initial guess of a controller. Convergence of the solution and solution itself is very sensitive to this initial value. Algorithm then performs searching for a local minimum around this point. Usually, the first guess is done through some generally-usable method like Ziegler-Nichols for example. However, generally there can be a lot of other minimum (following a given performance index better) in different part of the searched interval that may not be found. The reference [2] introduces a piece of code with a given x_0 that seems to restrict more universal usage of this solution.

3 IMPLEMENTATION OF THE PROPOSED SOLUTION

3.1 Novelty of the Solution

Although there are known referenced similar solutions regarding the use of optimization tuning techniques for PID controllers, there is no solution implemented at such comparable extent. The solutions referred in this paper mentioned in previous section 2.4 and the newly proposed solution have a lot of common features. However, the following significant novelties can be indicated in this paper:

• Use of global optimization methods

There is no need for specification of any range for controller parameters. According [4], this type of problem needs a constrained type of minimization problem. For Matlab-based solutions, *fmincon* function [4] is a great choice. Together with overshoot limits and manipulated value limits the algorithms prevents from finding possible solutions that may lead to unwanted oscillatory behavior.

• Weighting functions

They allows user to prefer one requirement to another. Apart from setting these four coefficients, user can also relax the bounding intervals for overshoot, manipulated value and phase margin so that they don't actually apply at all. In this case the found solution must be additionally and carefully analyzed in terms of following the control quality.

• Phase margin control

No reference solutions referred in this paper takes phase margin into consideration. However, it is an important respect for feedback control systems. The proposed solution generally guarantees a good result in this respect.

3.2 Introductory Description

Optimization algorithm has been implemented in Matlab environment as it includes many useful functions to handle control design problems. Particularly, some functions of Control System toolbox [5] and Optimization toolbox [4] have been used. The basic idea of the design algorithm is finding a global minimum of objective function containing PID controller parameters. The global optimization problem is set up and run according the following lines of code:

```
J=@function;
problem =
createOptimProblem('fmincon','x0',x0,
'objective',J,'lb',lb,'ub',ub);
gs = GlobalSearch;
[x,fval,flag,outpt,allmins] = run(gs,problem);
```

The search is carried out with the use of $fmincon$ function which is generally determined for finding minimum of constrained nonlinear multivariable function. Initial point is compulsory, but unlike local optimization, this value does not affect the process of finding the minima or the result, because it is a global optimization problem. Base on type of the controller (I, PI, PID) the initial values of all its components are set as 0.1. Function named *J=@function* calculates the value of objective function for a given vector x. Values lb and ub represent lower bound a upper bound of the interval where the search is performed, in this case $lb = 0$ and $ub = \infty$. According *GlobalSearch* function documentation, this setting means that the values of the components of optimized solution are limited to $10^4 + 1$. It can be supposed that this limitation does not degrade the usability of the proposed method in case of all various industrial processes.

Table 1. Syntax of *trapz* function for different integral criteria

Criterion	Syntax
ISE	`trapz(t,e.^2)`
IAE	`trapz(t,abs(e))`
ITSE	`trapz(t,(e.^2).*t)`
ITAE	`trapz(t,abs(e).*t)`

Searching for global minimum uses generation of random numbers to create a set of testing points. For this purpose Matlab uses so called global random number stream. As the default type of global random number stream in Matlab is *mt19937ar*, it was changed to *mcg16807* to achieve more robust results of optimization procedure.

Running the process of controller design is carried out by calling the *mainGlobal* function with a given set of input parameters:

```
mainGlobal(G, type, idx, peak, phl, phh, wlb,
wub, ulb, uub, k1, k2, k3, k4)
```

G – regulated linear static system of the 1st to the 4th order,
type – type of the desired controller, possible values: 'I', 'PI' , 'PID',
idx – type of integral criterion, possible values: 'ISE', 'IAE', 'ITSE', 'ITAE',
peak – overshoot limit (%),
phl – lower bound for the phase margin (°),
phh – higher bound for the phase margin (°),
wlb – lower bound for the setpoint,
wub – upper bound for the setpoint,
ulb – lower bound for manipulated value,
uub – upper bound for manipulated value,
k1, k2, k3, k4 – weighing coefficients for particular terms of the objective function, see section 3.8.

3.3 Objective Function

The objective function takes into consideration four respects: integral criterion value, overshoot, phase margin, manipulated value. It tries to minimize overall costs, while minimizing integral criteria and applying penalties if: overshoot is higher than its high limit specified by a user, or phase margin or manipulated values are outside the interval specified by a user. Generally, it can be expressed as the sum of particular components, yet the individual terms are additionally weighed, see section 3.8.

The process is carried out in iterations. Within each of iterations the values of particular components of the objective functions must be calculated, using open-loop or closed-loop properties. Transfer function of PID controller is set up with the use of *pid* function of the Control System Toolbox with P,I,D elements as the inputs. In case of PID controller, real PID is considered, being extended by additional time constant (filter coefficient) calculated as 1/100 th of the highest system time constant. In case of unstable closed-loop or any other numerical issues (NaN values, INF values) the process of calculation of the objective function is omitted and the highest possible value of 1.7977×10^{308} is assigned (maximum for double data type handled by Matlab).

3.4 Integral Criterion

The basic component that affects the resulting value of the objective function is a chosen integral criterion to be minimized by optimization algorithm. It is the main term that creates the set of possible optimal controllers while the other terms are designed in order to limit this set while following given limits and requirements.

The proposed solution offers four different types of integral criteria. Transfer function of control error $\mathrm{G_E}$ is computed firstly and the *step* function is applied:

```
e=step(Ge,t);
```

Particular input time vector is based on settling time of the closed-loop step response. Although the upper bound for the integral criteria is infinite, adequate precision of computation can be achieved by proper choice of the finite time t_f as described and discussed in [1] by formula (4). In this case, the finite time is taken as threefold of the settling and the entire time interval is divided into 30000 equidistant segments, this is calculated with the use of *stepinfo* function. As the time vector is also used in other part of the algorithms, it is firstly defined in twofold and modified consequently.

```
stpinfo_all=stepinfo(Gw);
st=3*stpinfo_all.SettlingTime;
t=linspace(0,2*st,60000); t=t(1:30000)';
```

Calculation of individual integrals by the use of *trapz* function is described by Table 1.

The value of integral criterion depends on time course (character) of control error and its length. Other terms of the objective function depend on the shape of this curve. For example, if there are two same time waveforms of the control with different time scale (different duration), then the influence of the integral criterion in the entire value of the objective function would be different. Provided that this duration depends on time constant of the system, it is desirable to have the same influence of integral criterion compared to the rest of the terms in the objective function. One of the ways of handling this issue is normalization considering the settling time — the integral value would be divided by the settling time value. However, if the duration is affected by parameters of the currently tested controller, diversity in the values of the integral values is necessary in order to be able to choose the better controller. Normalization is thus not considered, the problem is handled in a different way - the other terms in the objective function are multiplied by the settling time.

3.5 Overshoot

Selection of convenient result from various controllers designed during the iteration process while solving the global optimization task is affected by overshoot given in percents. If the controller with lower overshoot also meets the other requirements, it is of course preferred. If the overshoot exceeds maximum allowed limit, penalization on the resulting value of the objective function is applied proportionally to the difference between current overshoot and the limit.

The overshoot value is independent of the setpoint values; its calculation in percents always leads to the same results for different (step) changes of the setpoint. This value is then calculated based on the assumption of unit Heaviside step used as the setpoint

$$e = 1 - y\,.$$

Then the overshoot in percents is expressed as

$$overshoot = (\max(y) - 1) \times 100\,(\%).$$

Combining two previous formulas leads to

$$overshoot = \max(-e) \times 100\,(\%)$$

The value contributing to the overall sum in the objective function is then determined as difference between currently observed overshoot and specified allowed maximum, in case of a negative value of the overshoot it has a zero contribution (no penalty).

3.6 Phase Margin

General recommended interval for phase margin is usually defined as 30-60° . However, lower and upper bound can be customized outside this range. The phase margin itself is calculated based on open-loop transfer function G_O with the use of *margin* command. Its contribution to the overall sum in the objective function is proportional to the difference between currently observed value against lower or upper bounds. If it falls within the interval, it causes a zero contribution (no penalty). In order to provide a sufficient relevance for the phase margin, it has been set up that 10° overlap will correspond to 100 % overshoot.

3.7 Limits for Manipulated Value

As it is described in [1], "If there is no actuator saturation in the control system, the step response can be of any speed, and the control signal could be extremely large. In real applications, this is unacceptable". Therefore, the need for introducing of the limits for manipulated value is natural.

In order to apply the designed controller in practice, manipulated value must fall into the interval achievable in real control circuit assembly. Its value can be derived from setpoint values, particularly from specified range for the setpoint.

The first step during evaluation of a contribution caused by manipulated value includes calculation of staircase waveform consisting of two intervals where $w = wlb$ and consequently $w = wub$. The length of these segments is defined as above mentioned threefold of the closed-loop settling time. Simulation of the manipulated value is firstly performed at the first segment (setpoint changes between 0 and *wlb*) and then at the second segment (setpoint changes between *wlb* and *wub*). Performing the computation at both segments is the reason why time vector was generated at twofold range. The time waveform corresponding to manipulated value signal is calculated by transfer function of control error G_E, transfer function of the controller $\mathrm{G_R}$ and the time vector:

```
w=zeros(1,length(t)); deltat1=1:(length(t)/2);
deltat2=(length(t)/2+1):length(t);
w(deltat1)=wlb; w(deltat2)=wub; e=lsim(Ge,w,t);
u=lsim(Gr,e,t);
```

Maximum of the manipulated value is determined from the second segment, the minimum of the waveform (corresponding to setpoint change between *wub* and *wlb*) is not determined by the simulation itself but simply computed (a faster way then simulation), assuming time-axis symmetry applied for average value u_{mean} due to linearity of the system. This average value is computed from manipulated value at the steady state u_1 and u_2 (corresponding to $w = wlb$ and $w = wub$). Minimum of the manipulated value u_{min} is then determined as a difference between maximum and average subtracted from average

$$u_{\mathrm{mean}} = \frac{u_1 + u_2}{2}\,,$$
$$u_{\mathrm{min}} = u_{\mathrm{mean}} - (u_{\mathrm{max}} - u_{\mathrm{mean}})\,,$$
$$u_{\mathrm{min}} = \frac{u_1 + u_2}{2} - \Big(u_{\mathrm{max}} - \frac{u_1 + u_2}{2}\Big) = u_1 + u_2 - u_{\mathrm{max}}\,.$$

If maximum or minimal value is beyond a given limit, contribution of this issue is applied to the objective function in the form of penalization proportionally to the difference between current value and the bound value. In order to follow the comparable default weigh of this contribution, it is transferred to percent units related to the entire range:

```
umax = 100 * (umax - uub) / (uub - ulb);
umin = 100 * (ulb - umin) / (uub - ulb);
```

In case of no penalization the term has a zero value (no penalty).

3.8 Weighing Coefficients

Description of individual terms of the objective function introduced a way of normalization to keep the same influence of all of these components. However, formula defining the objective function has been extended by introducing weighing coefficients k_1 to k_4 being specified as the inputs when calling the optimization task, having a unit value by default. The following formula represents final form of the objective function including weighing coefficients $(st = setling\ time \cdot 3)$:

$$J = k_1 \times integral\,criteria + k_2 \times st \times overshoot + k_3 \times st \times 10 \times phase + k_4 \times st \times (u_{\max} + u_{\min})\,.$$

3.9 Recursion Add-on Mechanism

Some of thorough tests of the algorithm failed to find a solution. Optimization task was terminated prematurely roughly after 2000 iterations, incapable of finding a region where optimal solution might have been located. This situation was caused by the way of generation of testing points based on specified initial point which posed a convergence problem. Parameters of optimal controller in these cases were below than the initial point. Adjustment of the initial point towards lower values may seem to solve the problem, but in practice the number of iterations raised significantly and unacceptably. Therefore the algorithm was extended by an additional recursive procedure being initiated in such cases, while the initial value is set to its one hundredth and the task is then launched again.

4 TESTING AND EVALUATION ON CASE STUDIES

Verification of the proposed algorithm has been carried out for the linear system described by its transfer function

$$G(s) = \frac{1}{(12s+1)^2}\,.$$

The design of I-controller will be the subject of the following subsections that will introduce these two case studies:

- manual analytical solution (Section 4.1) based on ISE criterion, being verified by the proposed algorithm in (Section 4.2),
- manual analytical solution (Section 4.3) based on ITAE criterion, being verified by the proposed algorithm in (Section 4.4)

4.1 Minimization of ISE-based Objective Function — Manual Solution Using Parseval's Theorem

This subsection will introduce manual solution of optimization task considering ISE-based objective function for the design of I-controller $G_R(s) = \frac{k}{s}$ for a given linear system. The main idea of this computation is obtaining analytical form of the objective function $J(k)$ to be minimized using a basic knowledge of differential calculus. With the use of Parseval's theorem, it is possible to simplify the computation of ISE performance index as follows:

$$J_k = \int_0^{\infty} [y(t) - y(\infty)]^2 \mathrm{d}t = \frac{1}{2\pi j}\int_{-\infty}^{\infty} |E(j\omega)|^2 \mathrm{d}(j\omega) = \frac{1}{\pi}\int_0^{\infty} |E(j\omega)|^2 \mathrm{d}\omega = \frac{1}{2a_n}\frac{\det H_1}{\det H_n}$$

where

$$|E(j\omega)|^2 = E(j\omega)\,E(-j\omega)\,,$$

$$\mathcal{L}\{E(j\omega)\} = E(s) = Y(s) - \frac{1}{s}y(\infty) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{m-1}s^{n-1} + \cdots + a_1 s + a_0},$$

H_n – Hurwitz matrix of the n-th order set up based on coefficients of $E(s)$,
H_1 – Hurwitz matrix of the n-th order having 1st row substituted as

$$\begin{aligned} h_{1,1} &= (-1)^0 b_{n-1}^2\,,\\ h_{1,2} &= (-1)^1(b_1^2 - 2b_{n-1}b_{n-3})\,,\\ h_{1,n-1} &= (-1)^{n-2}(b_1^2 - 2b_0 b_2)\,,\\ b_{1,n} &= (-1)^{n-1}b_0^2\,. \end{aligned}$$

Let us apply the above mentioned procedure:

$$G_R(s) = \frac{k}{s}\,,\quad W(s) = \frac{1}{s}\,,$$

$$Y(s) = G_W(s)W(s) = \frac{G_R(s)G(s)}{1+G_R(s)G(s)}W(s) = \frac{k}{144s^4 + 24s^3 + s^2 + ks}\,,$$

$$y(\infty) = \lim_{s\to 0} sY(s) = \lim_{s\to 0} s\frac{k}{144s^4 + 24s^3 + s^2 + ks} = 1\,,$$

$$\overline{Y}(s) = \mathcal{L}\{y(t) - y(\infty)\} = Y(s) - \frac{1}{s}y(\infty)\,,$$

$$\overline{Y}(s) = -\frac{144s^2 + 24s + 1}{144s^3 + 24s^2 + s + k} = \frac{b_2 s^2 + b_1 s + b_0}{a_3 s^3 + a_2 s^2 + a_1 s + a_0}\,,$$

$$b_0 = -1\,,\ b_1 = -24\,,\ b_2 = -144\,,\ a_0 = k\,,\ a_1 = 1\,,\ a_2 = 24\,,\ a_3 = 144\,.$$

Consequently, Hurwitz matrices H_n $(n=3)$ and H_1 are defined as

$$H_n = \begin{bmatrix} a_2 & a_0 & 0\\ a_3 & a_1 & 0\\ 0 & a_2 & a_0 \end{bmatrix} = \begin{bmatrix} 24 & k & 0\\ 144 & 1 & 0\\ 0 & 24 & k \end{bmatrix},$$

$$h_{11} = (-1)^0 b_2^2 = 20736\,,$$

$$h_{12} = (-1)^1(b_1^2 - 2b_0 b_2) = -288\,,$$

$$h_{13} = (-1)^2 b_0^2 = 1\,,$$

$$H_1 = \begin{bmatrix} 20736 & -288 & 1\\ 144 & 1 & 0\\ 0 & 24 & k \end{bmatrix}.$$

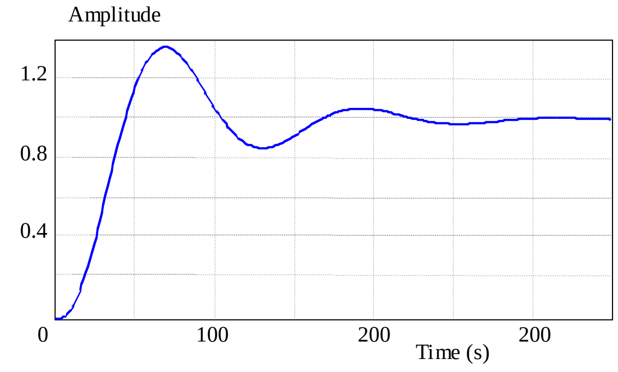

Fig. 2. Closed-loop step response (ISE criterion)

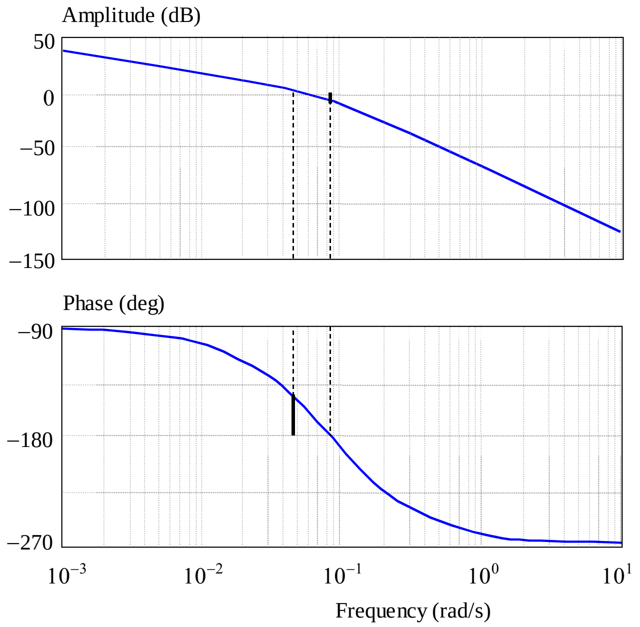

Fig. 3. Open-loop Bode diagram (ISE criterion): G_m =9.54 dB at 0.0833 rad/s, P_m = 34.8 deg at 0.0436 rad/s

The value of ISE-based performance, thus the value of the objective function, is calculated from coefficient a_n, determinants H_n and H_1 as follows:

$$J_k = \frac{\det H_1}{2a_n \det H_n} = \frac{\det H_1}{2a_3 \det H_n} = \frac{-18k-1}{12k^2-2k}\,.$$

The first derivative is used to find any possible local extremes, preferably with the use of *diff* and *solve* commands [6]:

$$\frac{\mathrm{d}J_k}{\mathrm{d}k} = 0\,.$$

Two solutions are found:

$$k_1 = \frac{1}{18}\,, \quad k_2 = -\frac{1}{6}\,.$$

Negative value is not considered by default. Verification of the first solution is done through the second derivative, corresponding to the minimum. Resulting controller is then expressed as follows:

$$G_R(s) = \frac{1}{18s}\,.$$

4.2 Verification by Proposed Solution with the Use of ISE Criterion

This section focuses on verification of the solution found in previous section by the proposed algorithm. To be able to compare the obtained result, no strict limitations are applied for overshoot, manipulated value and phase margin, and a unique step change is considered for the setpoint. Calling of the following syntax

```
s=tf('s'); G=1/(12*s+1)^2;
mainGlobal(G,'I','ISE',100,0,90,0,1,-10,10)
```

will find the solution after 2485 iterations, thus transfer function of I-controller is as

$$G_R(s) = \frac{0.0556}{s}\,.$$

This is identical result as the one found in Section 4.1. Figuress 2 and 3 show basic properties of the control circuit in time and frequency domain. Evaluation of $J(k)$ for $k = 0.0556$ determines the ISE for this case study $J = 27$. This value is identical as the one found with the used of proposed optimization algorithm.

4.3 Minimization of ITAE-based Objective Function — Manual Solution Using Optimum Coefficients

This subsection will introduce manual solution of optimization task considering ITAE-based objective function for the design of I-controller $G_R(s) = \frac{k}{s}$ for a given linear system. The procedure uses predefined optimum ITAE coefficients as generally described in [3] for example.

$$G_W = \frac{k}{144s^3 + 24s^2 + s + k} = \frac{\frac{k}{144}}{s^3 + \frac{1}{6}s^2 + \frac{1}{144}s + \frac{k}{144}}\,.$$

Normalized denominator of the closed-loop is then compared to the predefined ITAE polynomial [3]:

$$s^3 + 1.72as^2 + 2.17a^2 s + a^3\,.$$

It leads to a set of equations

$$a^3 = \frac{k}{144}\,, \quad 2.17a^2 = \frac{1}{144}\,, \quad 1.72a = \frac{24}{144}\,.$$

Solution of these equations determines two values of k:

$$k_1 = 0.1310\,, \quad k_2 = 0.0261\,.$$

Further analysis is needed to verify if these values represent optimal solution in terms of minimization of ITAE criterion. Equation (3) indicates a possible root $a = 0.0969$. Then, according (1), $k = 0.131$. However, according (2), $2.17 \cdot 0.0969^2 = 0.0204 \neq \frac{1}{144}$. Similarly, at the same time, (2) indicates a possible root $a = 0.0566$ (considering a positive root). Then, according (1), $k =$

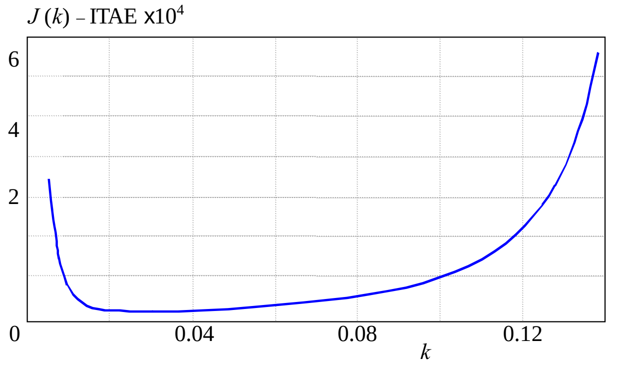

Fig. 4. Graphical representation of $J(k)$ function

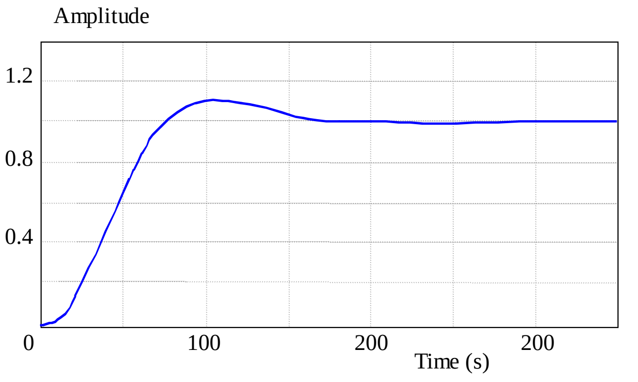

Fig. 5. Closed-loop step response (ITAE criterion)

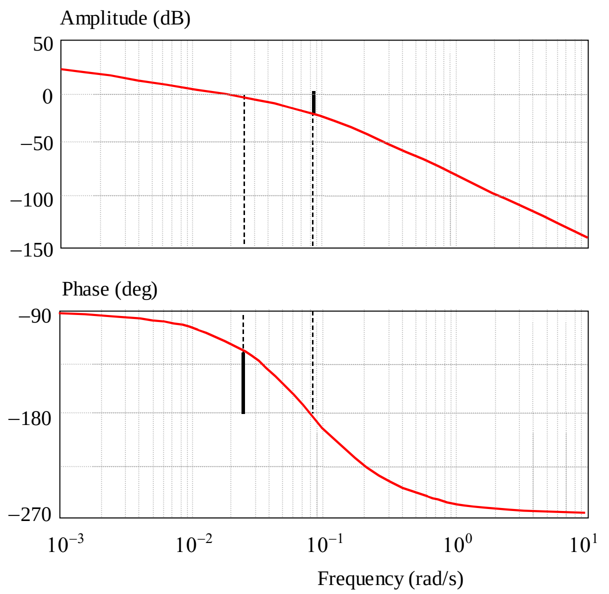

Fig. 6. Open-loop Bode diagram (ITAE criterion): $G_m = 16$ dB at 0.0833 rad/s, $\mathrm{P}_m = 57.4$ deg at 0.0243 rad/s

0.0261. However, according (3), $1.72a = 1.72 \cdot 0.0566 = 0.0974 \neq \frac{24}{144}$. This analysis proves that there is no unique adjustable parameter a that satisfies all of the equations. Regarding this case study, the existence of minimum would be verified numerically, plotting the objective function $J(k)$, see Fig. 4. Numerical analysis confirms $k = 0.0261$ to be the optimal value minimizing the objective function.

4.4 Verification by Proposed Solution with the use of ITAE Criterion

This section focuses on verification of the solution found in previous section by the proposed algorithm. To be able to compare the obtained result, no strict limitations are applied for overshoot, manipulated value and phase margin, and a unique step change is considered for the setpoint. Calling of the algorithm will be very similar to the one applied in Section 4.1, using $idx =$ 'ITAE' (indicating ITAE), see the following syntax:

```
s=tf('s'); G=1/(12*s+1)^2;
mainGlobal(G,'I','ITAE',100,0,90,0,1,-10,10)
```

The solution has been found after 2093 iteration. The transfer function of I-controller is

$$G_R(s) = \frac{0.0264}{s}\,.$$

Resulting value $k = 0.0264$ is very slightly different when compared to the one found in Section 4.3, $k = 0.0261$. There may be more explanation of this difference, apart from numerical accuracy of the solution there may also be different forms of the ITAE polynomials in various literature resources. For example, the following polynomial in the form of $s^3 + 1.75\omega s^2 + 2.15\omega^2 s + \omega^3$ can be found widely in the literature resources. Figures 5 and 6 show basic properties of the control circuit in time and frequency domain.

Table 2 gives a summary of basic properties of closed-loop circuit for methods introduced in Sections 4.3 and 4.4.

Table 2. Comparison of results achieved by different methods of ITAE-based minimization

Parameter	Procedure 4.3 ITAE coefficients	Procedure 4.4 proposed solution
Overshoot	10 %	10.3 %
Settling time	162 s	161 s
Phase margin	57.8°	57.4°

5 DISCUSSION

The number of iterations needed for finding a solution depends on complexity of the task (type of the controller versus the plant) and may differ, but typical amount for the category of the systems stated in Section 2.2 may be 2–3 thousands of iterations. This is of course closely connected to the computation time. Despite the higher computation times related to solution of global optimization problems, the advantages are prevailing significantly over the drawbacks.

The proposed algorithm is compatible with the latest Matlab R2016a and R2015b. It has been tested on two different configurations for the case study in Section 4.2:

- Matlab R2014b, PC Intel(R) Core(TM)2 Quad CPU Q8300 @ 2.5 GHz 2.5 GHz, 4 GB RAM, Windows 7 32b. (Benchmarks: unknown). The typical elapsed computation time was about 286 seconds (2485 iterations).
- Matlab R2015b, PC Intel(R) Core(TM)i5-4210U CPU @ 1.70 GHz 2.40 Ghz, 4 GB RAM, Windows 7 64b. (Benchmarks: 3DMark06: 4759 pts, PCMark 7 (Overall): 4194 pts). The typical elapsed computation time was about 182 seconds (2492 iterations).

There are some possible measures to be taken into consideration to speed up the next generation of the algorithm and the usability in general scale:

- computation of initial states for a given setpoint $w = wlb$ before running the algorithm core,
- adjustment of finding global minimum regarding necessity of the recursion add-on, possible different way of initial guess of the controller (based on plant roots in case of linear systems)
- possible replacing *GlobalSearch* by *MultiStart*, possibility of the use of multiprocessor cores, parallel computing techniques, different generation of testing points (different global random number stream),
- consideration of particle swarm optimization (PSO) techniques,
- extension of the categories of the systems suitable for the use of the proposed algorithm (transport delays, nonlinearities, astatic systems)

6 CONCLUSION AND SUMMARY

The proposed algorithm showed that the use of the integral criterion as the solely component of the objective function may not be a general solution; therefore other respects are applied as described in Section 3.

Performance indices used in this paper can affectively apply for a certain class of the systems. The choice of performance index is always up to an expert, just like the bounds for overshoot, manipulated value, type of controller, phase margin interval, and the weighing coefficient. The algorithm can be easily extended by IT^{2}SE or IT^{2}AE if needed.

The algorithm can also be adapted to nonlinear plants, replacing command line of the control system toolbox (*stepinfo, series, feedback*, ...) by nonlinear Simulink model. In this case this algorithm would apply for such a nonlinear system in an operating point without necessity of performing linearization.

Case studies documented in this paper focus on ISE and ITAE performance indices. Unlike the reference solution [1, 2, 7], the obtained results from case studies are not confronted with the ones based on different method, but they are compared to the same method with the use of different way of computation. This verifies correctness of the proposed algorithm that removes crucial drawbacks of classical tuning methods. As for ISE performance index, it leads to identical result compared to the one using Parseval's theorem. Regarding ITAE, it shows negligible differences as described in Table 2.

The use of analytical solution of optimization problems either for ISE (Parseval's theorem) or ITAE (use of predefined optimum coefficients) is very limited and applicable for a very narrow set of systems. Starting already with the design of two-parameters controllers (PI, PD) for the second order system the analytical solution with the use of Symbolic Toolbox [6] may end up with "Warning: Explicit solution could not be found" despite the existence of such solution.

Acknowledgment

This work was supported by project SP2016/162, Development of algorithms and systems for control, measurement and safety applications II of Student Grant System, VSB-TU Ostrava.

References

[1] DINGYU, X.—YANGQUAN, C.: OptimPID: A MATLAB Interface for Optimum PID Controller Design, 2005, http://www.mathworks.com/matlabcentral/fileexchange/34505-optimpid–an-optimal-pid-controller-design-interface.

[2] MARTINS, F. G.: Tuning PID Controllers using the ITAE Criterion, International Journal of Engineering Education **21** (2005), 867–873.

[3] LEVINE, S.: The Control Handbook, Second Edition, CRC Press, 2010.

[4] MathWorks Inc (2014b). Global Optimization Toolbox User's Manual.

[5] MathWorks Inc (2014b). Control System Toolbox User's Manual.

[6] MathWorks Inc (2014b). Symbolic Toolbox User's Manual.

[7] ZHUANG, M.—ATHERION, D. P.: Tuning PID Controllers with Integral Performance Criteria, In Proceedings of IEE International Conference Control'91, Edinburgh, UK, 1991, pp. 481–486.

Received 25 January 2016

Stepan Ozana was born in 1977 in Bílovec, Czech Republic. He studied electrical engineering at VSB — Technical university of Ostrava where he has got MSc degree in Control and Measurement Engineering, PhD degree and Assoc Prof degree in Technical Cybernetics. He works as an Associate Professor at the Department of Cybernetics and Biomedical Engineering, Faculty of Electrical Engineering and Computer Science. At present he gives lectures on Cybernetics and Control systems. Professional orientation: modeling and simulation of dynamic systems, control theory, automation, design and implementation of control algorithms (μPC/PC/sofPLC systems).

Tomas Docekal was born in 1992. He studied electrical engineering at VŠB — Technical university of Ostrava where he got the bachelor degree and MSc degree in control and measurement engineering. His professional orientation is mainly to robotics, embedded systems, and low-level programming.