Applied Soft Computing 40 (2016) 526–543

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Interval-model-based global optimization framework for robust stability and performance of PID controllers

Gorazd Karer*, Igor Škrjanc

Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 3 April 2015
Received in revised form 3 November 2015
Accepted 30 November 2015
Available online 14 December 2015

Keywords:
Robust PID controller
Parameter tuning
Interval model
Constrained particle swarm optimization
Robust stability and control performance

ABSTRACT

PID controller structure is regarded as a standard in the control-engineering community and is supported by a vast range of automation hardware. Therefore, PID controllers are widely used in industrial practice. However, the problem of tuning the controller parameters has to be tackled by the control engineer and this is often not dealt with in an optimal way, resulting in poor control performance and even compromised safety. The paper proposes a framework, which involves using an interval model for describing the uncertain or variable dynamics of the process. The framework employs a particle swarm optimization algorithm for obtaining the best performing PID controller with regard to several possible criteria, but at the same time taking into account the complementary sensitivity function constraints, which ensure robustness within the bounds of the uncertain parameters' intervals. Hence, the presented approach enables a simple, computationally tractable and efficient constrained optimization solution for tuning the parameters of the controller, while considering the eventual gain, pole, zero and time-delay uncertainties defined using an interval model of the controlled process. The results provide good control performance while assuring stability within the prescribed uncertainty constraints. Furthermore, the controller performance is adequate only if the relative system perturbations are considered, as proposed in the paper. The proposed approach has been tested on various examples. The results suggest that it is a useful framework for obtaining adequate controller parameters, which ensure robust stability and favorable control performance of the closed-loop, even when considerable process uncertainties are expected.

1. Introduction

Despite recent advances in control theory and several frameworks for advanced control approaches, PID controllers are still widely used in industrial practice. The reason for this is that the PID structure is well-known in the control-engineering community and is supported by a vast variety of automation hardware. PID controllers can be used for a broad range of processes in a simple and relatively robust manner. However, the problem of tuning the controller parameters has to be tackled by the control engineer and this is often not dealt with in an optimal way.

In literature, there are several PID tuning strategies and rules that are well known and often used (e.g. [1]). These rules generally allow the engineer to adequately set the PID parameters and achieve a certain degree of control performance and robustness. However, one cannot be certain that the controller is optimally tuned. On one hand, if the parameters are set too conservatively, the control performance will suffer. On the other hand, if the parameters are set too aggressively, the robustness properties are lost and the closed-loop system can even become unstable.

Therefore, in order to design a controller for a process with uncertain parameters, it is essential to take these uncertainties into account. In such manner, it can be assured that the controller will put the closed-loop dynamics on the edge of the required robustness properties and thus ensure high control performance in operation. By considering the bounds of the uncertain parameters' intervals it is possible to derive the required robustness properties by limiting the complementary sensitivity function. If the imposed constraints are not violated, closed-loop stability can be ensured.

In the last years, several heuristic-based and partially bio-inspired optimization approaches, such as genetic algorithm, ant colony optimization, particle swarm optimization etc. have been gaining popularity, due to their fast convergence and especially their ability to avoid local extrema and result in the global extremum. Particle swarm optimization (PSO) has been introduced in [2]. Since then, it has been used in many global optimization applications. The basic approach has been modified in many regards. Various techniques for the alleviation of local minima and for detecting multiple minimizers are described in [3]. Furthermore, the paper also presents results on the ability of the PSO in tackling multiobjective, minimax, integer programming and several other approaches. Various advanced multiobjective PSO (MOPSO) algorithms have been presented lately. MOPSO was first introduced in [4]. These algorithms tend to be more effective when dealing with multiobjective optimization. A MOPSO algorithm, where Pareto dominance is used to determine the flight direction of a particle, is presented in [5]. The computational complexity is treated ad reduced in [6]. A review of MOPSO state-of-the-art can be found in [7–9]. Besides some later MOPSO applications (e.g. in the field of grid optimization [10–12]), a survey of applications is presented in [13]. Vector evaluated PSO (VEPSO) is also suitable for multiobjective optimization [14,15,8]. Some

* Corresponding author. Tel.: +386 14768701
E-mail address: gorazd.karer@fe.uni-lj.si (G. Karer).

http://dx.doi.org/10.1016/j.asoc.2015.11.046
1568-4946/

promising results have been obtained by involving fuzzy logic in the PSO algorithm [16–18].

There are several interesting approaches to PID-type controller tuning in the literature. In [19], a tuning algorithm is presented, which is used to design and tune PI controllers for stable processes with a small dead time while meeting specified gain and phase margins. In [20] a predictive PI controller is used for dead-time dominated linear time-invariant first-order processes. In [21,22], the author graphically computes all feasible gain and phase margin specifications-oriented robust PID controllers for open-loop unstable processes with time delay. Another graphical approach to finding stabilizing parameter sets of PID parameters with regard to gain and phase margins for a class of processes with time-delay is presented in [23]. An approach with internal model control based PID tuning method for 2nd order processes with time delay, again considering gain and phase margins is presented in [24]. A genetic algorithm and bacterial foraging based PID tuning is proposed in [25]. Advanced tuning approaches involving fuzzy controller and iterative tuning with applications are presented in [26–28].

PSO algorithms have also been used for PID tuning. Conventionally, in order for the controller to satisfy a specified gain and phase margin specification, the design is carried out graphically by trial-and-error use of Bode plots. The authors in [29] propose a PI controller tuning optimization based on gain and phase margin specifications. The approach uses PSO combined with Hooke Jeeves method in the second stage. An modified PSO for tuning robust PID controllers is presented in [30]. A design approach to PI Controllers based on non-convex optimization of load disturbance rejection is proposed in [31]. A PI controller design for linear systems with parametric uncertainty with respect to sensitivity margin is presented in [32]. A PID controller tuned using PSO is compared to a PID controller tuned using Ziegler–Nichols rule in [33,34]. A modified PSO is used to tune a PID controller in [35], whereas a PID-type fuzzy controller is tuned using PSO in [36].

Various methods for tuning PID controllers using PSO have been implemented, but approaches considering the robustness of the solution with regard to parameter uncertainty are rather scarce. The goal of this paper is to present an approach, where a PSO algorithm is used to obtain the best performing robust PID controller. Several performance criteria can be considered by including them in the fitness function. At the same time, the solution takes into account the frequency-domain based complementary sensitivity function constraints, so as to ensure robustness within the bounds of the uncertain parameters' intervals. Furthermore, the relative stability requirements are given using the maximum value of sensitivity function as defined in [37,38]. Unlike some other approaches, the considered maximum value of sensitivity function takes into account both gain and phase margins and is thus more appropriate as a relative stability measure.

Hence, the presented approach can be regarded as a framework for obtaining a simple, computationally tractable and efficient optimization solution for tuning the parameters of the controller, while considering the gain, pole, zero and time-delay uncertainty of the controlled process. The goal is to derive a robust PID controller that provides good control performance while assuring robust stability within the prescribed uncertainty constraints that affect the robustness of the closed-loop system. From the a priori estimations of the uncertainties of the process, the presented approach ensures that robust stability constraints are not violated. Therefore, it provides good control performance and at the same time robustly safe closed-loop operation, even when dealing with relatively complex uncertain process dynamics.

The paper is structured as follows. In Section 2, the problem is introduced and the basic preliminaries are given. The idea behind the representing system that is used in the optimization is proposed. Gain, delay, zero and pole uncertainties are treated and a simple illustrative example is shown. In Section 3, the controller design is dealt with. The PSO algorithm mechanism is explained and the particular features used in the approach are pointed out. Using the simple illustrative example, an appropriate PID controller is derived and the closed-loop properties are analyzed. Section 4 presents a three tanks system case study. A nonlinear mathematical model of the system is derived and linearized for a general operating point. The proposed PSO approach is used to tune the PID controller parameters and the closed-loop behavior is analyzed. Finally, the controller is tested on the nonlinear plant. The results show that the controller performance is adequate only when the relative system perturbations are considered as proposed in the paper. The framework is discussed with regard to its repeatability and convergence properties. In Section 5, another case study is treated. The considered system is a batch reactor used in medicine production by a pharmaceutical company in Slovenia. The closed-loop results are shown and the repeatability and convergence properties are discussed. The results further support the usability of the proposed framework. Section 6 gives some concluding remarks. In Appendix A, the pseudo code of the PSO algorithm is presented and in Appendix B, the derivation of the representing system parameters is explained.

2. Problem description and preliminaries

Consider a process with a known structure and several uncertain process parameters that lie in a known bounded interval. The goal is to design a suitable PID controller that will ensure adequate stability properties and at the same time fast operation for any of the process varieties within the boundaries.

Assume the system is formulated in the factorized form, as shown in Eq. (1).

$$G_P(s) = K_{DC} \cdot \frac{\prod_{j=1}^{m}\left(1+\frac{s}{b_j}\right)}{\prod_{i=1}^{n}\left(1+\frac{s}{a_i}\right)} \cdot e^{-\tau s} \tag{1}$$

In Eq. (1), K_{DC} is the process gain, τ stands for the delay, $-b_j$ (where $j \in \{1, \ldots, m\}$) denote the process zeros and $-a_i$ (where $i \in \{1, \ldots, n\}$) are the process poles.

When dealing with such processes, the actual uncertainties in the process parameters that eventually influence closed-loop behavior have to be taken into account. When the controller is tuned for good control performance for one particular variation of the uncertain process, as it is often the case in industrial practice, it cannot be ensured that the control performance will still remain adequate when a particular uncertain process-parameter changes its value. Furthermore, it is also possible that the closed-loop eventually becomes unstable when that controller is used.

If there is one uncertain process-parameter with known boundaries, the most obvious control-design approach would be to derive two bounding systems that consider the minimum and the maximum value of the uncertain parameter, respectively. The controller would then be designed and tuned by taking into account both extreme cases, assuming that the actual process' dynamics lie somewhere in between. However, if there are n uncertain process parameters with known boundaries, 2^n bounding systems need to be considered. Therefore, the number of bounding systems can easily become quite high and it is at least unpractical to tune the controller by taking into account the whole set of the bounding systems.

The goal is to derive the parameters of the controller using PSO [2]. In the proposed approach the optimization is run only once. In order to be able to do that, the representing process used for the optimization must be derived and robustness of the solution according to the process-parameter uncertainties must be ensured.

2.1. Derivation of the representing system

Before going into further design steps, it is sensible to transform the problem of dealing with the system with uncertain parameters to a system without any uncertainties, which nevertheless best represents the original system with uncertain parameters. The representing system is then used in the PSO algorithm in order to obtain the optimal controller parameters.

Let us take into account the robust stability properties expressed in Eq. (2).

$$\left|\frac{\Delta G_P(j\omega)}{G_P(j\omega)}\right| \leq \left|\frac{1}{T(j\omega)}\right| \tag{2}$$

Here, $G_P(j\omega)$ is the original system, $\Delta G_P(j\omega)$ stands for the system perturbation[1] and $T(j\omega)$ is the complementary sensitivity function – see Eq. (3). The maximum value of the complementary sensitive function is denoted as M_t – see Eq. (4). However, a more general solution can be obtained by limiting the maximum value of the complementary sensitive function depending on the

[1] In this case, the perturbations are considered as a consequence of uncertain parameters.

frequency. Thus, from Eq. (2), it is possible to derive the robustness requirement as given in Eq. (5).

$$T(j\omega)=\frac{G_P(j\omega)\cdot G_R(j\omega)}{1+G_P(j\omega)\cdot G_R(j\omega)} \tag{3}$$

$$M_t=\max_{\omega}|T(j\omega)| \tag{4}$$

$$|T(j\omega)|\leq T_{\max}(\omega);$$
$$T_{\max}(\omega)=\frac{1}{\left|\frac{\Delta G_P(j\omega)}{G_P(j\omega)}\right|} \tag{5}$$

Here, $G_R(j\omega)$ stands for the controller.

The closed-loop system will remain stable for system perturbations ΔG_P (or for that matter relative system perturbations $\frac{\Delta G_P}{G_P}$) that comply with Eq. (2).

Hence, in order to relieve the stability constraints with regard to the admissible complementary sensitivity function, it is vital to derive such representing system $G_{P0}(j\omega)$ so that the relative system perturbations $\frac{\Delta G_{P0}}{G_{P0}}$ due to the uncertain parameters are the smallest possible for even the worst case scenario.

Therefore, four possible types of system perturbations due to uncertain parameters are considered and each of them is analyzed separately:

- gain uncertainty,
- delay uncertainty,
- zero uncertainty,
- pole uncertainty.

2.1.1. Gain uncertainty

In this case, the gain of the system is assumed to be uncertain (parameter K_{DC} of the system $G_P(s)$), as given in Eq. (6).

$$G_P(s)=K_{DC}\cdot G_{Pfix}(s)$$
$$K_{DC1}<K_{DC}<K_{DC2} \tag{6}$$

Here, $G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected by any uncertainties. In order to ensure that the relative system perturbations due to the uncertain parameter are as small as possible, the representing system $G_{P0}(s)$ should be defined as given in Eq. (7).

$$G_{P0}(s)=K_{DC0}\cdot G_{Pfix}(s) \tag{7}$$

The resulting parameter K_0 can be derived as defined in Eq. (8). The proof is given in Appendix B.

$$K_{DC0}=\frac{K_{DC1}+K_{DC2}}{2} \tag{8}$$

2.1.2. Delay uncertainty

The time-delay of the system is assumed to be uncertain (parameter τ of the system $G_P(s)$), as given in Eq. (9).

$$G_P(s)=e^{-\tau s}\cdot G_{Pfix}(s);$$
$$\tau_0\leq\tau\leq\tau_{\max} \tag{9}$$

$G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected byany uncertainties. Assume the representing system $G_{P0}(s)$ is defined as given in Eq. (10) – see Appendix B.

$$G_{P0}(s)=e^{-\tau_0 s}\cdot G_{Pfix}(s) \tag{10}$$

2.1.3. Zero uncertainty

In this case, the position of one of the zeros in the transfer function of the system is assumed to be uncertain (parameter $-b$ of the system $G_P(s)$), as given in Eq. (11).

$$G_P(s)=\left(1+\frac{s}{b}\right)\cdot G_{Pfix}(s)$$
$$b_1<b<b_2 \tag{11}$$

Here, $G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected by any uncertainties. In order to ensure that the relative system perturbations due to the uncertain parameter are as small as possible, the representing system $G_{P0}(s)$ should be defined as given in Eq. (12).

$$G_{P0}(s)=\left(1+\frac{s}{b_0}\right)\cdot G_{Pfix}(s) \tag{12}$$

The resulting parameter b_0 can be derived as defined in Eq. (13). The proof is given in Appendix B.

$$b_0=\frac{2b_1b_2}{b_1+b_2} \tag{13}$$

2.1.4. Pole uncertainty

The position of one of the poles in the transfer function of the system is assumed to be uncertain (parameter $-a$ of the system $G_P(s)$), as given in Eq. (14).

$$G_P(s)=\frac{1}{1+\frac{s}{a}}\cdot G_{Pfix}(s)$$
$$b_1<b<b_2 \tag{14}$$

$G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected by any uncertainties. In order to ensure that the relative system perturbations due to the uncertain parameter are as small as possible, the representing system $G_{P0}(s)$ should be defined as given in Eq. (15).

$$G_{P0}(s)=\frac{1}{1+\frac{s}{a_0}}\cdot G_{Pfix}(s) \tag{15}$$

The resulting parameter a_0 can be derived as defined in Eq. (16). The result depends on the most interesting frequency band. The proof is given in Appendix B.

$$a_0=a_1+\frac{a_2-a_1}{1+\sqrt{\frac{\omega^2+a_2^2}{\omega^2+a_1^2}}} \tag{16}$$

This means that a_0 lies in the interval bounded by the low-frequency solution ($\omega\to 0$) and the high-frequency solution ($\omega\to\infty$) as defined in Eqs. (17)–(19).

$$a_0\in[a_0|_{\omega=0},\quad a_0|_{\omega=\infty}] \tag{17}$$

$$a_0|_{\omega=0}=\frac{2a_1a_2}{a_1+a_2} \tag{18}$$

$$a_0|_{\omega=\infty}=\frac{a_1+a_2}{2} \tag{19}$$

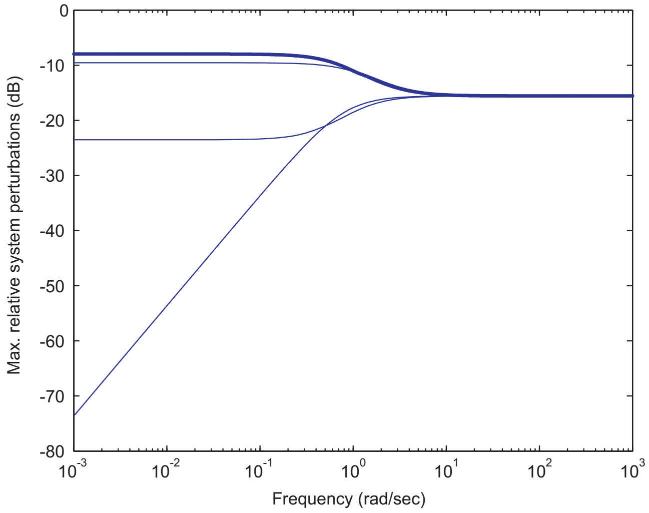

Fig. 1. Maximum relative system perturbations $|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}|$.

2.2. A simple illustrative example

In the illustrative example, a 3rd order process with a variable gain and a variable dominant pole is considered, as shown in Eq. (20).

$$\begin{aligned} &G_P(s) = \frac{K}{(s+a)(s+1.5)(s+2)};\\ &25 < K < 35,\\ &0.8 < a < 1.2 \end{aligned} \tag{20}$$

There are $2^2 = 4$ bounding systems for the illustrative example in Eq. (20). The bounding systems are given in Eq. (21).

$$\begin{aligned} G_{P1}(s) &= \frac{25}{(s+0.8)(s+1.5)(s+2)}\\ G_{P2}(s) &= \frac{35}{(s+0.8)(s+1.5)(s+2)}\\ G_{P3}(s) &= \frac{25}{(s+1.2)(s+1.5)(s+2)}\\ G_{P4}(s) &= \frac{35}{(s+1.2)(s+1.5)(s+2)} \end{aligned} \tag{21}$$

2.2.1. The resulting representing system

In the aforementioned case (see Eq. (20)) gain uncertainty and pole uncertainty is dealt with. Therefore, the representing system is derived by taking into account Eqs. (7), (8), (15) and (18). In this manner, the representing transfer function can be formulated as in Eq. (22).

$$G_{P0}(s) = \frac{30}{(s+0.96)(s+1.5)(s+2)} \tag{22}$$

Fig. 1 depicts the maximum relative system perturbations of the representing system G_{P0} with regard to G_{P1}, G_{P2}, G_{P3} and G_{P4}, as defined in Eq. (23). The bold line represents the maximum relative system perturbations perturbation for a given frequency.

$$\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right| = \max_i \left|\frac{G_{Pi}(j\omega) - G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|, \quad \text{where} \quad i = 1, 2, 3, 4 \tag{23}$$

3. Controller design

The transfer function of the PID controller is given in Eq. (24).

$$G_R(s) = K_P\left(1 + \frac{1}{T_I s} + \frac{T_D s}{T_f s + 1}\right) \tag{24}$$

There are four parameters needed to fully define the PID controller in Eq. (24), namely K_P, T_I, T_D, and T_f. The approach tries to alleviate the computational burden of the optimization algorithm by only seeking the optimal parameters K_P, T_I and T_D. The filtering parameter is defined in a standard manner as $T_f = \frac{T_D}{5}$.

In this case, the maximum values of the sensitivity function S (Eq. (26)) and complementary sensitivity function T (Eq. (3)) are considered in order to ensure stability and robustness requirements.

According to Eq. (2), the closed-loop system will remain stable for relative process perturbations $|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}|$ that are bounded with regard to the complementary sensitivity function T. Therefore, by knowing the maximum relative process perturbations as depicted in Fig. 1, it is possible to set the constraint for the value of $M_{t,max}$, which is defined in Eq. (4), as given in Eq. (25).

$$M_{t,\max} = \frac{1}{\max_\omega \left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|} \tag{25}$$

In this case, $M_{t,\max} = \frac{1}{0.4} = 2.5$. Hence, if this constraint is fulfilled, the closed-loop system will keep its stability properties for the whole range of uncertain systems within the prescribed limit.

The stability requirements can be specified using the sensitivity function, or rather its maximum value – see Eqs. (26) and (27).

$$S(j\omega) = \frac{1}{1 + G_P(j\omega)\cdot G_R(j\omega)} \tag{26}$$

$$M_s = \max_\omega |S(j\omega)| \tag{27}$$

The inverse of the maximum value of the sensitivity function $d = \frac{1}{M_s}$ can be graphically interpreted as the minimum distance of the Nyquist diagram from the critical point $-1 + 0j$. In this regard, it is a better stability criterion than the classic gain margin g_m and the phase margin φ_m. However, the criteria are connected as given in Eqs. (28) and (29).

$$g_m \geq \frac{M_s}{M_s - 1} \tag{28}$$

$$\varphi_m \geq 2 \arcsin \frac{1}{2M_s} \tag{29}$$

Assume $g_m \approx 6$dB and $\varphi_m \approx 30°$ are required. This means that the solution should aim for $M_{s,max} \approx 2$.

3.1. Particle swarm optimization algorithm

The particle swarm optimization (PSO) algorithm is employed in order to find the optimal solution of the problem. A particle represent one of the possible solutions, in this case as defined in Eq. (30).

$$x = \begin{bmatrix} T_I \\ T_D \\ K_P \end{bmatrix} \tag{30}$$

3.1.1. PSO parameters

The algorithm is initiated using the parameters[2] in Eq. (31).

$$
\begin{aligned}
x_{minmax} &= \begin{bmatrix} 0.0001 & 20 \\ 0.0001 & 5 \\ 0.0001 & 2 \end{bmatrix} \\
v_{minmax} &= \begin{bmatrix} -50 & 50 \\ -50 & 50 \\ -5 & 5 \end{bmatrix} \\
n_{iter} &= 40 \\
n_{part} &= 100 \\
\omega &= 0.7 \\
c_1 &= 1.5 \\
c_2 &= 1.5
\end{aligned} \tag{31}
$$

Note that the solutions obtained with the proposed framework indicate that the choice of parameters hardly presents a practical problem to the user as long the suggested values are set: the PSO algorithm yields similar results with various sets of parameter values.

Here, x_{minmax} denotes the minimum and maximum value of particles, v_{minmax} stands for the minimum and maximum particle velocity. Note that there are as many rows as the dimension of the solution space. n_{iter} is the maximum number of iterations, n_{part} denotes the number of particles, ω is the inertial factor, and c_1 and c_2 represent the self-cognitive constant and social-component factor, respectively.

The aforementioned parameters influence the performance of the PSO algorithm especially from the convergence and efficiency aspects. Despite there are several useful methods for determining the values of the parameters available (see e.g. [39–41]), in this case the choice of proper parameters can be done heuristically, due to the relatively simple optimization task.

The minimum and maximum value of particles x_{minmax} are set so that the expected optimal solutions lie within the boundaries, which is not too difficult for x in Eq. (30). If in doubt, a larger interval can be used. The drawback is that this might require more iterations n_{iter} and particles n_{part}, rising the computational complexity. Anyways, when the resulting solution lies very close to the border of the interval x_{minmax}, a wider interval needs to be selected.

The minimum and maximum velocity of particles v_{minmax} influence the convergence rate. If a smaller interval is used, the solution will converge at a slower rate, but the local search around the extremum will be smoother. However, due to the use of inertial factor ω, the algorithm is not so sensitive to the values in v_{minmax}.

The number of iterations n_{ite} and particles n_{part} influence the computational complexity. The values were selected by examining the convergence trajectories. Since the task is not computationally critical, a rather large number of iterations n_{ite} and particles n_{part} was used in the algorithm.

The inertial factor ω influences how the velocity of a particle in the previous iteration influences the velocity in the next iteration. Thus, low inertial factors are more suitable for local extremum search (exploitation), whereas higher inertial factors favor global extremum search (exploration). The self-cognitive constant and social-component factor c_1 and c_2 influence the algorithm in of rate of convergence and alleviation of local extrema. In the literature, several benchmark optimization problems are treated and different sets of parameters are proposed. The parameters ω, c_1 and c_2 are set as suggested in [42] – see Eq. (31).

3.1.2. Fitness function

The fitness function for the optimization is defined in Eq. (32). It considers two continuous criteria in Eqs. (33) and (34), and one discrete criterion representing the robustness and stability constraint in Eq. (35).

$$F(G_{ZZ}) = -J_1(G_{ZZ}) - 0.1 \cdot J_2(G_{ZZ}) - J_c(G_{ZZ}) \tag{32}$$

$$J_1(G_{ZZ}) = -f_b(G_{ZZ}) \tag{33}$$

$$J_2(G_{ZZ}) = T_{ST,2\%}(G_{ZZ}) \tag{34}$$

$$J_c(G_{ZZ}) = \left\{ \begin{array}{ll} \infty; & G_{ZZ} \text{ is unstable} \bigvee Ms(G_{ZZ}) > M_{s,\max} \bigvee Mt(G_{ZZ}) > M_{t,\max} \\ 0; & \text{otherwise} \end{array} \right\} \tag{35}$$

Here, G_{ZZ} denotes the closed-loop transfer function, $f_b(G_{ZZ})$ is the bandwidth of the closed-loop system, and $T_{ST,2\%}(G_{ZZ})$ stands for the settling time of the step response of the closed-loop system.

A practically usable fitness function can be constructed quite easily. It is however important to include several objectives that are in conflict with each other in the fitness function. This makes the approach related to multiobjective optimization methods. In literature, there are several multiobjective optimization methods that could be implemented in the proposed approach [7–9]. However, in order to keep the algorithm as simple as possible, all the objectives are gathered into one fitness function and the basic PSO algorithm described in Appendix A is used.

The simple criterion function enables the use of a simpler algorithm. On one hand, the algorithm could seek e.g. Pareto fronts so as to make the optimization more transparent, but this would make the approach more difficult to implement for an average user. Furthermore, a more advanced global optimization algorithm could also be used to improve the convergence and computational complexity aspects. However, that could also make the approach more complicated and further research is needed to support the potential advantages in this regard.

The criteria in Eqs. (33) and (34) can be pondered arbitrarily. Changing the ponder values affects the closed-loop behavior, however they can be set in a relatively simple manner. The criteria J_1, J_2 and J_c have several different purposes in the fitness function:

- The continuous criterion J_1 in Eq. (33) treats the bandwidth of the closed-loop system, which should be as high as possible. However, this criterion alone would provide a solution with a very high value of K_P, resulting in a high bandwidth of the closed-loop system with oscillatory response, rendering a very long settling time.
- The continuous criterion J_2 in Eq. (34) treats the settling time of the step response of the closed-loop system, which should be as short as possible. This criterion counteracts the high-bandwidth requirement by keeping the value of K_P low and thus preventing the unwanted closed-loop oscillations. In this case, the settling time was pondered with regard to the bandwidth heuristically, as shown in Eq. (32). This can easily be achieved by rising the ponder of J_2 as much as required to prevent excessive oscillation. It needs to be noted that fine adjustments of the ponder value are not needed as it does not influence the final closed-loop behavior dramatically.

[2] Note that the solutions obtained with the proposed framework indicate that the choice of parameters hardly presents a practical problem to the user as long as the suggested values are set: the algorithm yields similar results with various sets of parameter values.

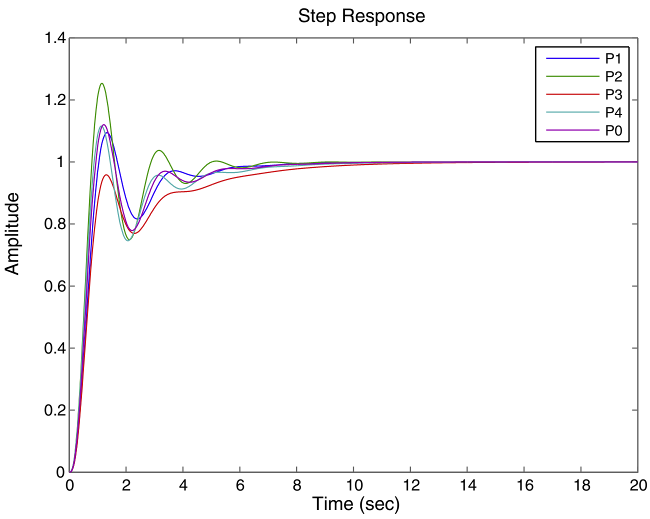

Fig. 2. Closed-loop step response $y(t)$ for G_{P1}, G_{P2}, G_{P3}, and G_{P4}.

- The discrete criterion J_c in Eq. (35) treats the robustness and stability and is used for constraining the optimization algorithm. The particles that violate the robustness or stability constraints cannot be considered as a final solution. The appropriate constraints are set by considering the expected parameter uncertainties of the representing system. Therefore, J_c ensures safe and robustly stable closed-loop operation within the uncertainties' limits.

3.2. Closed-loop behavior

The PSO algorithm described above is run in order to obtain a solution that represents the optimal parameters of the PID controller to be used. The solution considers the uncertain parameters, which are constrained with the interval bounds. The goal of the presented framework is thus to provide the best performing PID controller with regard to several possible criteria, but at the same time taking into account the complementary sensitivity function constraints, which ensure robustness within the bounds of uncertain parameters' intervals.

The closed-loop step response for the representing system and all the bounding systems using the optimized controller is shown in Figs. 2 and 3.

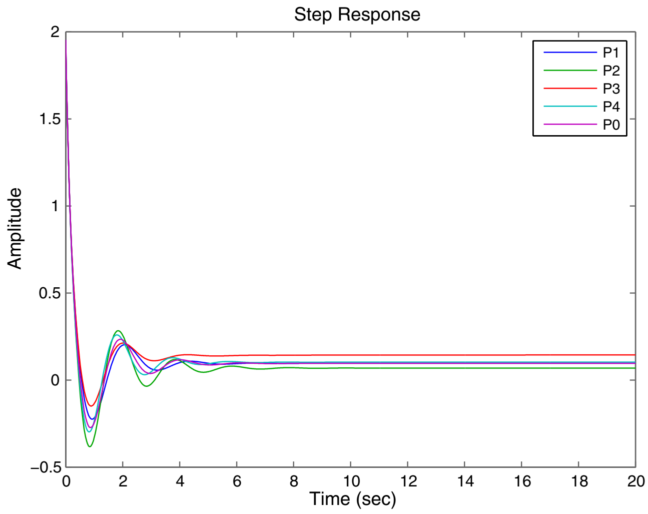

Fig. 3. Closed-loop step response $u(t)$ for G_{P1}, G_{P2}, G_{P3}, and G_{P4}.

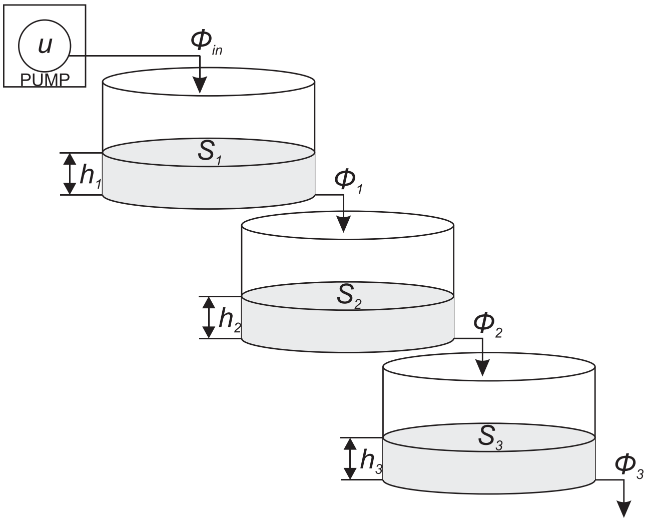

Fig. 4. Three tanks hydraulic system.

4. Case study 1: Three tanks hydraulic system

4.1. Three tanks hydraulic system

In order to illustrate the proposed approach for controller design on a realistic system, a three tanks hydraulic system is considered.

The system is depicted in Fig. 4. It consists of three containers for liquids. The input of the system represents the voltage u applied to the pump, which induces flow ϕ_{in} that fills the first tank according to Eq. (36), where K_u is a parameter of the pump. The liquid flows from the first tank into the second one with flow ϕ_1 according to Eq. (37). Here, K_1 denotes the constant of the output pipe in the first tank, and h_1 stands for the level of liquid in the first tank. Similarly, the liquid flows from the second tank into the third one with flow ϕ_2 according to Eq. (38) and finally out of the third tank according to Eq. (39). Note that the installation involves considerable length of piping between the pump and the first tank and between the tanks, therefore, the actual flow at one point is subject to a considerable time delay with regard to the flow at another point. K_2 and K_2 denote the constant of the output pipe, and h_2 and h_3 stand for the level of liquid in the second tank and the third tank, respectively. The output of the system is the level of liquid in the third tank h_3.

$$\phi_{in}(t) = K_u \cdot u(t-\tau) \tag{36}$$

$$\phi_1 = K_1\sqrt{h_1} \tag{37}$$

$$\phi_2 = K_2\sqrt{h_2} \tag{38}$$

$$\phi_3 = K_3\sqrt{h_3} \tag{39}$$

Note that the time delay is introduced at the input in Eq. (36). The dynamics of the system are given in Eqs. (40)–(42).

$$\frac{dh_1}{dt} = \frac{1}{S_1}(\phi_{in} - \phi_1) \tag{40}$$

$$\frac{dh_2}{dt} = \frac{1}{S_2}(\phi_1 - \phi_2) \tag{41}$$

$$\frac{dh_3}{dt} = \frac{1}{S_3}(\phi_2 - \phi_3) \tag{42}$$

In Eqs. (40)–(42), S_1, S_2 and S_3 represent the surface area of the first, second and third tank, respectively.

4.2. Linearization at the operating point

The proposed system can be linearized at an operating point. Assume it is linearized around the input value of u_0. In this case, the variables of the deviation model can be rewritten as in the following equations.

$$u = u_0 + \Delta u \tag{43}$$

$$\phi_{in} = \phi_{in0} + \Delta\phi_{in}$$
$$\Delta\phi_{in}(t) = \frac{\partial \phi_{in}(t)}{\partial u(t-\tau)}|_{u=u_0} \cdot \Delta u(t-\tau) = K_u \Delta u(t-\tau) \tag{44}$$

$$\phi_1 = \phi_{10} + \Delta\phi_1$$
$$\Delta\phi_1 = \frac{\partial \phi_1}{\partial h_1}|_{h_1=h_{10}} \cdot \Delta h_1 = \frac{K_1}{2\sqrt{h_{10}}} \cdot \Delta h_1 \tag{45}$$

$$\phi_2 = \phi_{20} + \Delta\phi_2$$
$$\Delta\phi_2 = \frac{\partial \phi_2}{\partial h_2}|_{h_2=h_{20}} \cdot \Delta h_2 = \frac{K_2}{2\sqrt{h_{20}}} \cdot \Delta h_2 \tag{46}$$

$$\phi_3 = \phi_{30} + \Delta\phi_3$$
$$\Delta\phi_3 = \frac{\partial \phi_3}{\partial h_3}|_{h_3=h_{30}} \cdot \Delta h_3 = \frac{K_3}{2\sqrt{h_{30}}} \cdot \Delta h_3 \tag{47}$$

A steady state at the chosen operating point is reached when the inflows and the outflows of all the tanks are equal. Therefore, the following constants for a particular operating point can be derived.

$$\phi_{in0} = \phi_{10} = \phi_{20} = \phi_{30}$$
$$K_u \cdot u_0 = K_1\sqrt{h_{10}} = K_2\sqrt{h_{20}} = K_3\sqrt{h_{30}}$$
$$h_{10} = \left(\frac{K_u \cdot u_0}{K_1}\right)^2$$
$$h_{20} = \left(\frac{K_u \cdot u_0}{K_2}\right)^2 \tag{48}$$
$$h_{30} = \left(\frac{K_u \cdot u_0}{K_3}\right)^2$$

By taking into account the aforementioned equations and rewriting the linearized system in the state-space form, Eqs. (49) and (50) are obtained.

$$\begin{bmatrix} \Delta\dot{h}_1(t) \\ \Delta\dot{h}_2(t) \\ \Delta\dot{h}_3(t) \end{bmatrix} = \begin{bmatrix} -\frac{K_1^2}{2\cdot S_1\cdot K_u\cdot u_0} & , & 0 & , & 0 \\ \frac{K_1^2}{2\cdot S_2\cdot K_u\cdot u_0} & , & -\frac{K_2^2}{2\cdot S_2\cdot K_u\cdot u_0} & , & 0 \\ 0 & , & \frac{K_2^2}{2\cdot S_3\cdot K_u\cdot u_0} & , & -\frac{K_3^2}{2\cdot S_3\cdot K_u\cdot u_0} \end{bmatrix} \begin{bmatrix} \Delta h_1(t) \\ \Delta h_2(t) \\ \Delta h_3(t) \end{bmatrix} + \begin{bmatrix} \frac{K_u}{S_1} \\ 0 \\ 0 \end{bmatrix} \cdot \Delta u(t-\tau) \tag{49}$$

$$\Delta h_3(t) = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta h_1(t) \\ \Delta h_2(t) \\ \Delta h_3(t) \end{bmatrix} \tag{50}$$

By converting Eqs. (49) and (50), the transfer function in Eq. (51) and Eqs. (52)–(55) is derived.

$$G_P(s) = \frac{K}{(s+a)(s+b)(s+c)} \cdot e^{-s\tau} \tag{51}$$

$$K = \frac{K_u \cdot K_1^2 \cdot K_2^2}{4\cdot S_1\cdot S_2\cdot S_3\cdot (K_u\cdot u_0)^2} \tag{52}$$

$$a = \frac{K_1^2}{2\cdot S_1\cdot K_u\cdot u_0} \tag{53}$$

$$b = \frac{K_2^2}{2\cdot S_2\cdot K_u\cdot u_0} \tag{54}$$

$$c = \frac{K_3^2}{2\cdot S_3\cdot K_u\cdot u_0} \tag{55}$$

4.3. System parameters

The basic parameters of the exemplary system are as follows.

$$\begin{aligned} S_1 &= S_2 = S_3 = \pi \cdot 0.07^2 = 15.4 \cdot 10^{-3}\,\mathrm{m}^2 \\ K_1 &= 1.22 \cdot 10^{-3}\,\mathrm{m}^3/\mathrm{s}\sqrt{\mathrm{m}} \\ K_2 &= 1.41 \cdot 10^{-3}\,\mathrm{m}^3/\mathrm{s}\sqrt{\mathrm{m}} \\ K_3 &= 1.00 \cdot 10^{-3}\,\mathrm{m}^3/\mathrm{s}\sqrt{\mathrm{m}} \\ K_u &= 0.10 \cdot 10^{-3}\,\mathrm{m}^3/\mathrm{V\,s} \end{aligned} \tag{56}$$

The operating point is assumed to vary between the output value $h_{30,1} = 7\,\mathrm{m} \le h_{30} \le h_{30,2} = 12\,\mathrm{m}$. The constant of the output pipe of the third tank varies between $K_{32} = 0.0009\,\mathrm{m}^3/\mathrm{s}\sqrt{\mathrm{m}} \le K_3 \le K_{31} = 0.001\,\mathrm{m}^3/\mathrm{s}\sqrt{\mathrm{m}}$. Furthermore, the delay is limited to $\tau \le 50\,\mathrm{s}$. Therefore, gain, poles and delay of the transfer function in Eq. (51) fall within the following boundaries. Hence, 32 bounding systems G_{Pi}, where $i \in \{1, \ldots, 32\}$, must be considered.

$$\begin{aligned} 1.71 \cdot 10^{-6} &\le K \le 2.94 \cdot 10^{-6} \\ 1.41 \cdot 10^{-2} &\le a \le 1.84 \cdot 10^{-2} \\ 1.88 \cdot 10^{-2} &\le b \le 2.46 \cdot 10^{-2} \\ 0.94 \cdot 10^{-2} &\le c \le 1.23 \cdot 10^{-2} \end{aligned} \tag{57}$$

4.4. The representing transfer function

By taking into account Eqs. (7)–(8), (9), (15) and (18), the representing transfer function given in Eq. (58) is derived.

$$G_{P0}(s) = \frac{2.33 \cdot 10^{-6}}{(s+0.0160)(s+0.0213)(s+0.0106)} \tag{58}$$

The bode plot of the representing system G_{P0} is shown in Fig. 5. The Nyquist diagram of the representing system G_{P0} and the family of Nyquist diagrams of the extreme cases G_{Pi}, where $i \in \{1, \ldots, 32\}$, are shown in Fig. 6.

Furthermore, Fig. 7 depicts the maximum relative system perturbations of the representing system G_{P0} with regard to G_{Pi}, where $i \in \{1, \ldots, 32\}$, as defined in Eq. (59).

$$\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right| = \max_i \left|\frac{G_{Pi}(j\omega) - G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|, \quad \text{where} \quad i \in \{1, \ldots, 32\} \tag{59}$$

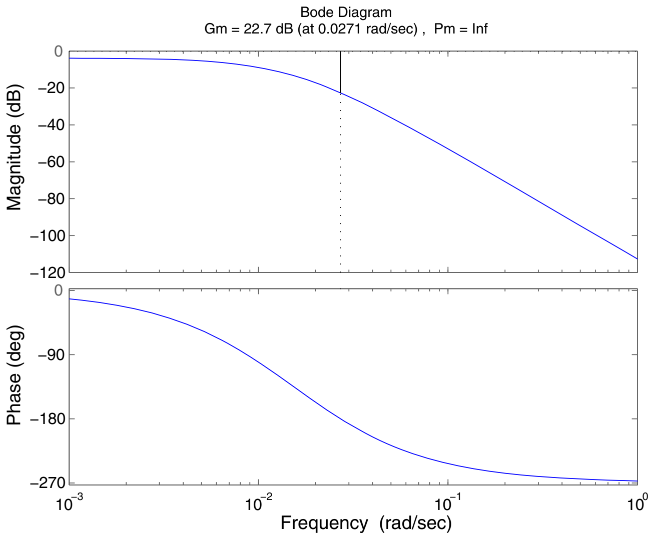

Fig. 5. Bode diagram for G_{P0}.

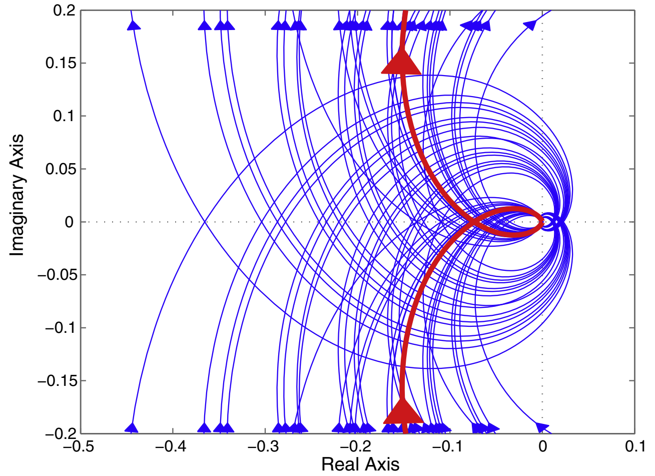

Fig. 6. Nyquist diagrams for G_{Pi}, where $i \in \{0, \ldots, 32\}$.

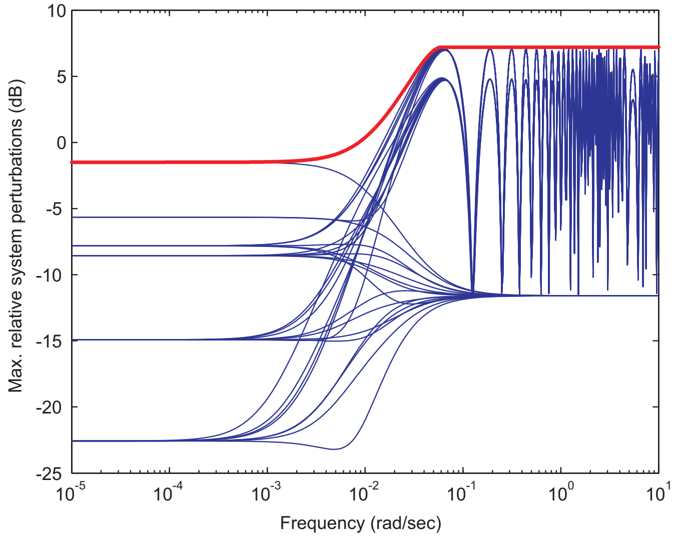

Fig. 7. Maximum relative system perturbations $\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|$.

4.5. Optimization of controller parameters

The particle swarm optimization of controller parameters is carried out as described in Section 3. A particle represent one of the possible solutions, in this case as defined in Eq. (60).

$$x = \begin{bmatrix} T_I \\ T_D \\ K_P \end{bmatrix} \tag{60}$$

The algorithm is initiated using the parameters in Eq. (61).

$$\begin{aligned} x_{minmax} &= \begin{bmatrix} 1 & 300 \\ 1 & 300 \\ 1 & 300 \end{bmatrix} \\ v_{minmax} &= \begin{bmatrix} -3 & 3 \\ -3 & 3 \\ -3 & 3 \end{bmatrix} \\ n_{iter} &= 40 \\ n_{part} &= 100 \\ \omega &= 0.7 \\ c_1 &= 1.5 \\ c_2 &= 1.5 \end{aligned} \tag{61}$$

The fitness function is defined in Eqs. (62)–(64).

$$F(G_{ZZ}) = -J_1(G_{ZZ}) - J_c(G_{ZZ}) \tag{62}$$

$$J_1(G_{ZZ}) = -f_b(G_{ZZ}) \tag{63}$$

$$J_c(G_{ZZ}) = \left\{ \begin{array}{ll} \infty; & G_{ZZ} \text{ is unstable} \bigvee Ms(G_{ZZ}) > M_{s,\max} \bigvee |T(G_{ZZ}, j\omega)| > T_{\max}(\omega) \\ 0; & \text{otherwise} \end{array} \right\} \tag{64}$$

The optimized parameters are given in Eq. (65).

$$\begin{aligned} K_P &= 2.82 \\ T_I &= 141 \\ T_D &= 61.11 \\ T_f &= 12.22 \end{aligned} \tag{65}$$

4.6. Closed-loop behavior

The closed-loop step response for the representing system and all the bounding systems using the optimized controller are shown in Figs. 8 and 9. Furthermore, Fig. 10 shows the optimized complementary sensitivity function and the constraint imposed by the relative system perturbations. The constraint is not violated, therefore, the closed loop is stable for all bounding systems, although some cases are very close to the stability limit.

4.7. Practical control performance

In order to prove the practical applicability of the proposed approach an experiment on a real nonlinear three tank hydraulic system simulation example was conducted. The reference signal covered the operating range from $h_3 = 7$ to $h_3 = 12$. Furthermore, there is an additive input disturbance $\Delta u_{dist} = 2V$ applied between $t = 500$ s and $t = 1750$ s.

The closed-loop control performance is analyzed for two cases: first, the expected poor results are shown for the case where the relative system perturbations are not considered in the design phase;

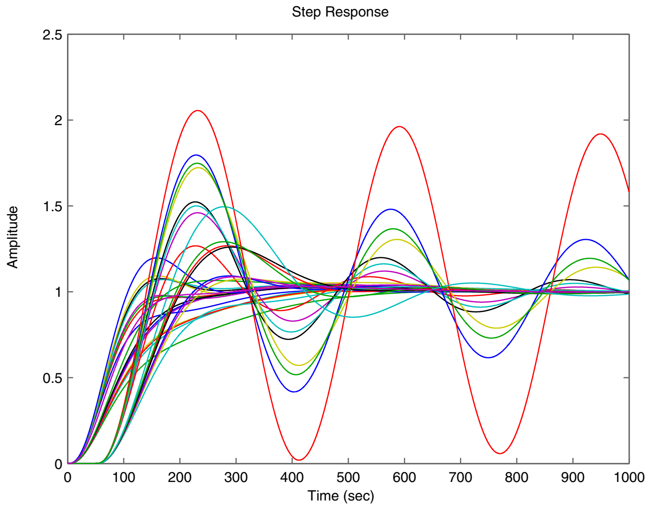

Fig. 8. Closed-loop step response $\Delta h_3(t)$.

next, the case where the relative system perturbations are taken into account as proposed in the paper is shown.

4.7.1. *Poor control performance: relative system perturbations have not been considered*

In case relative system perturbations are not considered in the design phase, the optimized solution yields a controller that is adequate only for the linearized case of the representing system in one operating point. However, critical closed-loop properties, such as stability, cannot be guaranteed in the whole operating range. The closed-loop response is shown in Fig. 11. The results show that the overall controller performance is not adequate. This was expected, because relative system perturbations have not been considered in the design phase.

4.7.2. *Good control performance: relative system perturbations have been considered*

If the proposed approach is carried out as proposed in the paper, the relative system perturbations are taken into account by constraining the complementary sensitivity function as shown in Fig. 10. In this case, the closed-loop response is shown in Figs. 12 and 13.

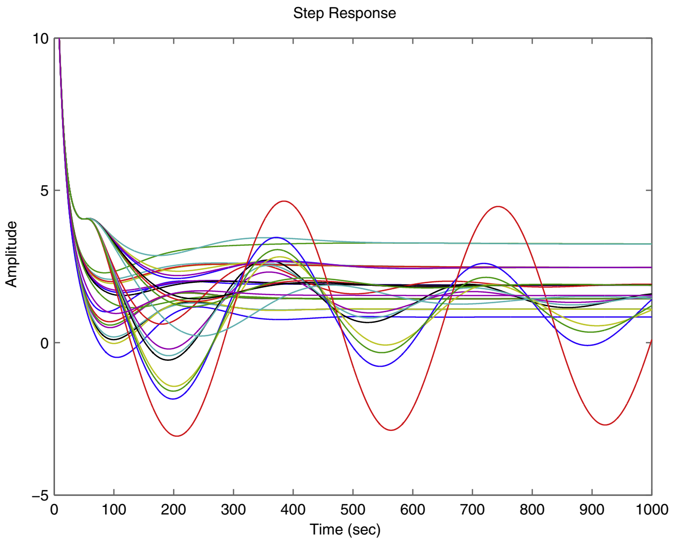

Fig. 9. Closed-loop step response $\Delta u(t)$.

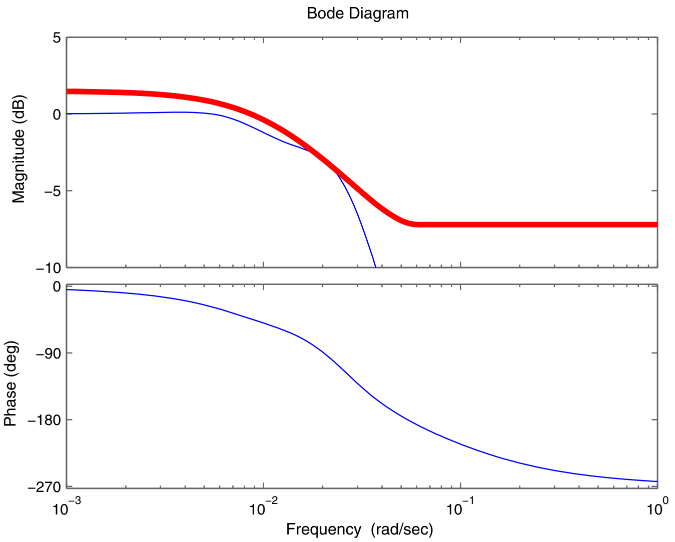

Fig. 10. Complementary sensitivity functions for the representing system G_{P0} (blue line). The red line represents the constraint imposed on the complementary sensitivity function that guarantees robust stability of the closed loop system with regard to the boundaries of the interval models parameters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

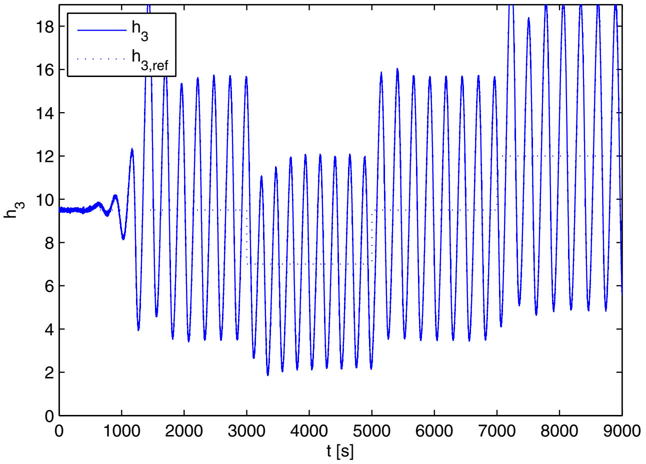

Fig. 11. Poor control performance: closed-loop response $h_3(t)$ – non-robust solution.

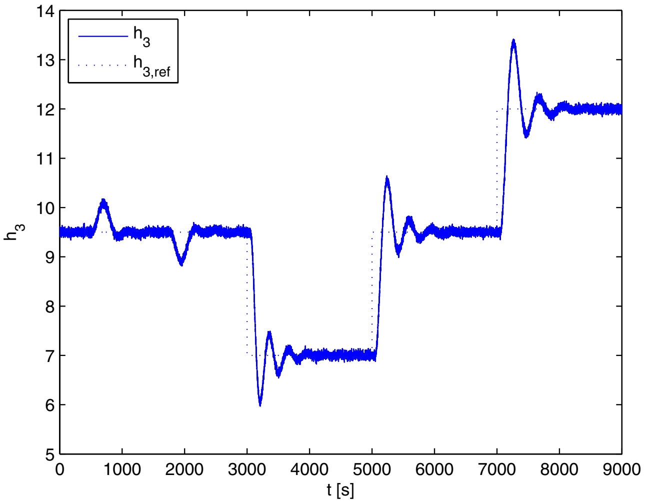

Fig. 12. Good control performance: closed-loop response $h_3(t)$ – robust solution.

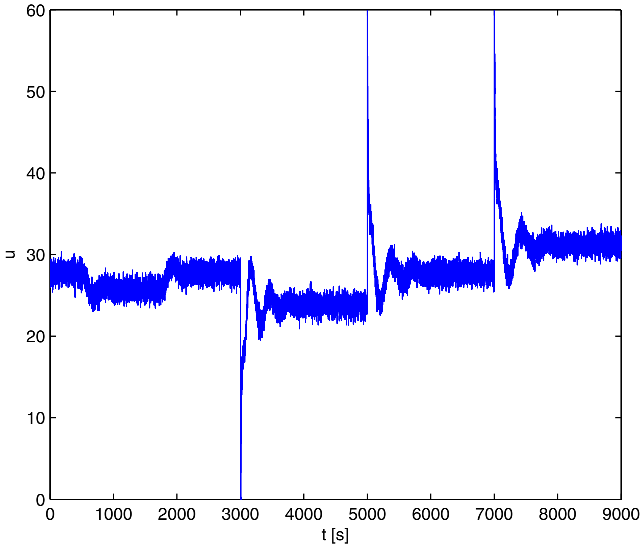

Fig. 13. Good control performance: closed-loop response $u(t)$ – robust solution.

The results indicate that the reference-trajectory tracking is satisfactory within the whole operating range that defines the robustness requirements.

4.8. Discussion on the proposed framework

The results suggest that the presented approach can be a useful framework for obtaining controller parameters, even when dealing with relatively complex system dynamics. However, in order to further analyze the proposed approach, especially with regard to its repeatability and convergence properties, the algorithm has been used to solve the task (described in Section 5) 30 times consecutively.

For the 30 consecutive solutions, the globally best value of the fitness function F_{gBest} with regard to the iteration step *iter* is depicted in Fig. 14. The set of 30 closed-loop step responses is depicted in Figs. 15 and 16.

As the optimality of the solution is linked to the formulation of the fitness function, the solution itself – i.e. the controller parameters K_P, T_I and T_D – tend to be slightly dispersed. Hence, the closed-loop step responses of the representing system using the set 30 consecutive solutions are similar, but not exactly equal, as can be seen in Figs. 15 and 16. This has been expected as the solution values are interdependent, yielding similar control performance with respect to the fitness function with different parameter values, therefore, they must be interpreted within the context of the optimization task. That said, the final value of the fitness function $F_{gBest,fin}$, which actually indicates the convergence and repeatability of the algorithm, seems to reach almost the same value every time (see Fig. 14), with the average value $F_{gBest,fin} = 0.02029$ and very small standard deviation $\sigma(F_{gBest,fin}) = 7.7 \cdot 10^{-6}$.

The statistics of the consecutive algorithm runs are given in Table 1.

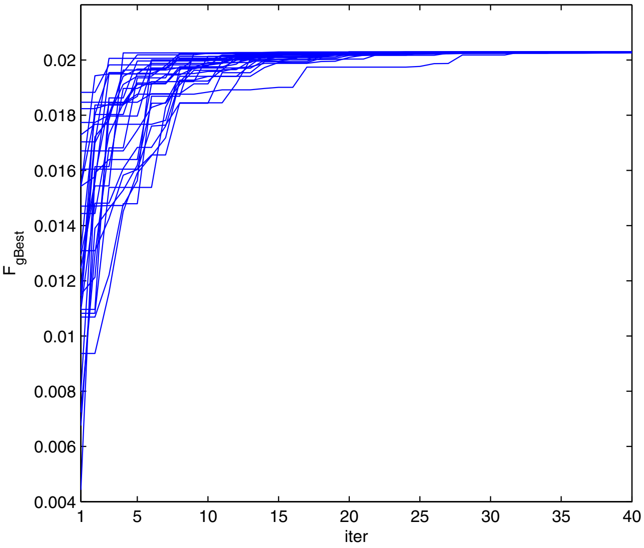

Fig. 14. Globally best value of the fitness function with regard to the iteration step for 30 consecutive algorithm runs.

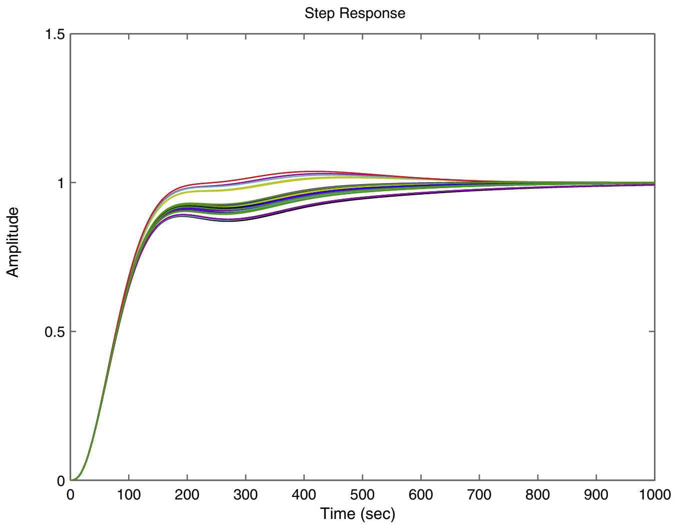

Fig. 15. Set of closed-loop step responses for 30 consecutive solutions.

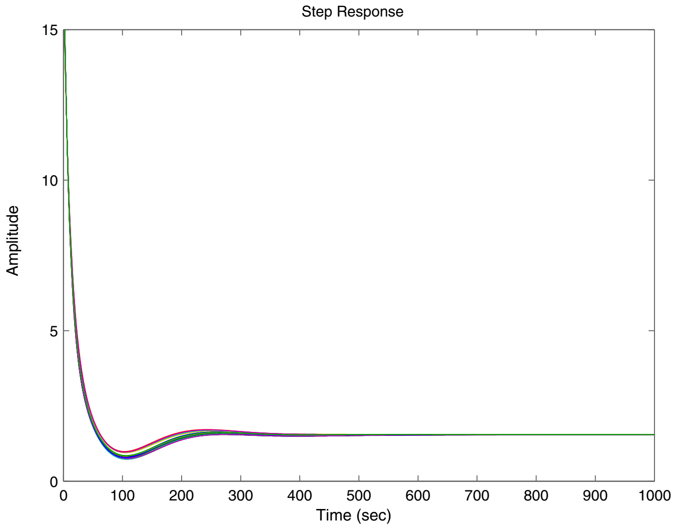

Fig. 16. Set of closed-loop step responses for 30 consecutive solutions.

Table 1
Mean value, standard deviation, normalized standard deviation of the results of the consecutive algorithm runs.

Parameter (x)	$\bar{x}$	$\sigma(x)$	$\sigma(x)/\bar{x}$
K_P	2.9	0.052	0.018
T_I	204	33.1	0.16
T_D	55.2	2.60	0.047
$F_{gBest,fin}$	0.02029	$7.7 \cdot 10^{-6}$	$3.8 \cdot 10^{-4}$

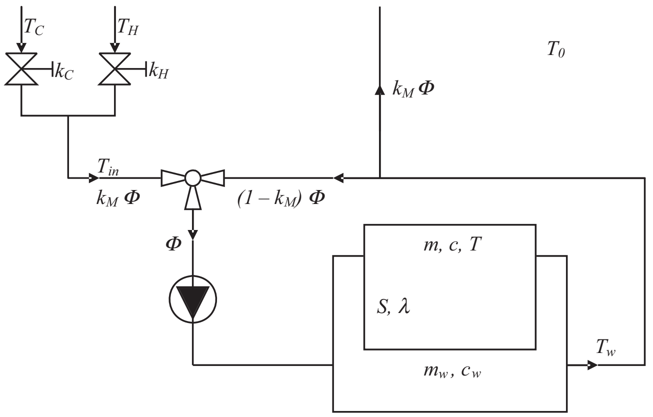

Fig. 17. Scheme of the batch reactor.

The results of the repeated runs lead to the conclusion that the algorithm is capable of reaching the optimal solution with regard to the predefined fitness function.

Furthermore, the computational complexity of the approach has been estimated. On a computer running Matlab, the solutions have been obtained in 213 s on average. In the worst case, 219 s were required, while in the best case, the solution was obtained in 210 s. The time required to consecutively carry out the optimization indicates that the computational complexity of the approach is quite predictable and does not vary from one run to another.

5. Case study 2: Batch reactor

5.1. Batch reactor

The proposed approach has been applied on a batch rector example. The case study includes the results of the experiments carried out on model of a real batch reactor. The considered batch reactor is situated in a pharmaceutical company in Slovenia and is used in production of medicines [43].

The aim is to control the temperature of the ingredients that are stirred in the core of the batch reactor, where the final product is made. In order to achieve a satisfactory quality of the final product it is vital to ensure that the trajectory of the reference temperature, which is defined in the production recipe, is followed as accurately as possible.

The control problem of the batch reactor has been approached in various ways. Several advanced control methods have been considered, such as predictive control based on a reachability analysis [43], predictive control based on a genetic algorithm [44], self-adaptive predictive control with an online local-linear-model identification [45–47] and control using an inverse hybrid fuzzy model [48]. That said, the industrial partners in the company that actually use such batch reactors have been traditionally inclined towards PID controllers and tend to prefer PID over more advanced control methods. In addition, PID controllers can easily be implemented using the existing hardware. However, due to the parameter variability of the plant, the parameter intervals need to be considered, which turns out to be problematic using classic design approaches. In this regard, the proposed framework proves to be very useful.

A scheme of the batch reactor is shown in Fig. 17. The reactor's core (temperature T) is heated or cooled through the reactor's water jacket (temperature T_w). The heating medium in the water jacket is a mixture of fresh input water, which enters the reactor through on/off valves, and reflux water. The water is pumped into the water jacket with a constant flow ϕ.

Table 2
Inputs of the batch reactor.

Label	Meaning
k_M	Position of the mixing valve
k_C	Position of the cold-water on/off valve
k_H	Position of the hot-water on/off valve

Table 3
Outputs of the batch reactor.

Label	Meaning
T	Temperature in the core of the reactor
T_w	Temperature in the water jacket of the reactor

The dynamics of the system depend on the physical properties of the batch reactor, i.e. the mass m and the specific heat capacity c of the ingredients in the reactor's core and in the reactor's water jacket (here, the index w denotes the water jacket). λ is the thermal conductivity, S is the contact area and T_0 is the temperature of the surroundings.

T_{in} stands for the temperature of the fresh input water. T_C and T_H denote the temperature of the cool and hot fresh input water, respectively. The position of the on/off valves for cool and hot fresh input water are labeled k_H and k_C, respectively. The position of the mixing valve is denoted by k_M.

The temperature of the fresh input water T_{in} depends on two inputs: the position of the on/off valves k_H and k_C. There are two possible operating modes of the on/off valves. In case $k_C = 1$ and $k_H = 0$, the input water is cool ($T_{in} = T_C = 12\,^\circ\mathrm{C}$), whereas if $k_C = 0$ and $k_H = 1$, the input water is hot ($T_{in} = T_H = 75\,^\circ\mathrm{C}$).

The ratio of fresh input water to reflux water is controlled by the third input, i.e. by the position of the mixing valve k_M.

Table 2 contains the input signals, Table 3 contains the input signals, and Table 4 the other physical parameters and quantities related to the batch reactor.

Thus, the considered process can be regarded as a multivariable system with two discrete inputs (k_C, k_H), a continuous input (k_M) and two measurable outputs (T and T_w). Due to the nature of the system, the time constant of the temperature in the water jacket is obviously much shorter than the time constant of the temperature in the reactor's core. Therefore, the batch reactor is considered as a stiff system.

5.2. Basic mathematical model of the batch reactor using differential equations

The dynamics of the batch reactor can be described using differential equations, which can be derived from the law of conservation of energy. The heat flows in the batch reactor can be divided as follows.

Table 4
Other physical parameters and quantities related to the batch reactor.

Label	Meaning
T_0	Temperature of the surroundings
T_{in}	Temperature of the fresh input water
T_C	Temperature of cold water
T_H	Temperature of hot water
ϕ	Water flow in the jacket
m	Mass of ingredients in the core
c	Specific heat of the ingredients in the core
m_w	Mass of water in the jacket
c_w	Specific heat of water in the jacket
λ	Thermal conductivity between the core and the jacket
S	Contact surface between the core and the jacket
λ_0	Thermal conductivity between the jacket and the surroundings
S_0	Contact surface between the jacket and the surroundings

- heat conduction between the reactor's core and the reactor's water jacket;
- heat conduction between the reactor's water jacket and the surroundings;
- heat convection due to fresh water inflow in the reactor's water jacket;
- heat convection due the outflow of water from the reactor's water jacket.

For the sake of simplicity, the mathematical submodels for the core and for the water jacket of the batch reactor are treated separately. In this manner, it is possible to split the complex multivariable system representing the batch reactor into two simpler multiple-input-single-output subsystems.

The temperature in the reactor's core T is affected only by the heat conduction between the reactor's core and the reactor's water jacket. Furthermore, it is surmised that the heat conduction is proportional to the temperature difference between the reactor's core T and the reactor's water jacket T_w. It is therefore possible to derive a differential equation from the law of conservation of energy for the core of the batch reactor as given in Eq. (66).

$$mc\frac{dT}{dt} = \lambda S(T_w - T) \tag{66}$$

The temperature in the reactor water jacket T_w is affected by all of the aforementioned heat flows taking place in the batch reactor. A differential equation is derived from the law of conservation of energy for the water jacket of the batch reactor as given in Eq. (67).

$$\begin{aligned} m_w c_w \frac{dT_w}{dt} &= k_M \phi c_w T_{in} + (1 - k_M)\phi c_w T_w \\ &\quad - \phi c_w T_w - \lambda S(T_w - T) - \lambda_0 S_0 (T_w - T_0) \\ &= k_M \phi c_w T_{in} - k_M \phi c_w T_w - \lambda S(T_w - T) - \lambda_0 S_0 (T_w - T_0) \end{aligned} \tag{67}$$

The valves are rather large and thus cannot be moved instantaneously. Due to slow movement of the valves in the reactor an input delay τ is introduced, which represents the time delay between the change of the input signal and the actual resulting change of the valve position.

Without going too much into detail, the resulting model can be expressed by defining $y = T - T_0$ and $u = (T_{in} - T_w) \cdot u_{k_M}$ as a transfer function given in Eq. (68).

$$G_P(s) = \frac{Y(s)}{U(s)} = \frac{\frac{\phi \lambda S}{m m_w c} \cdot e^{-s\tau}}{s^2 + \left(\frac{\lambda S}{mc} + \frac{\lambda S}{m_w c_w} + \frac{\lambda_0 S_0}{m_w c_w}\right) s + \frac{\lambda S \lambda_0 S_0}{m c m_w c_w}} \tag{68}$$

Eq. (68) can be rewritten as Eq. (69), with K, a, and b in (70)–(72), respectively.

$$G_P(s) = \frac{Y(s)}{U(s)} = \frac{K}{(s+a)(s+b)} \cdot e^{-s\tau} \tag{69}$$

$$K = \frac{\phi \lambda S}{m m_w c} \tag{70}$$

The measured parameters of the real batch reactor are as follows.

$$\begin{aligned} m_w &= 40\,\text{kg} \\ c_w &= 4200\,\text{J/kg K} \\ c &= 4200\,\text{J/kg K} \\ T0 &= 18\,{}^\circ\text{C} \\ TC &= 12\,{}^\circ\text{C} \\ TH &= 75\,{}^\circ\text{C} \\ \phi &= 1.6\,\text{kg/s} \\ \lambda &= 420\,\text{W/m}^2\,\text{K} \\ \lambda_0 &= 84\,\text{W/m}^2\,\text{K} \\ S &= 2\,\text{m}^2 \\ S0 &= 2\,\text{m}^2 \end{aligned} \tag{73}$$

The quantity of the raw material being processed in the batch reactor is not constant. During the production process, additional material is injected into the core of the batch reactor, according to the reference recipe. Thus, the quantity varies between $m_1 = 300\,\text{kg} \leq m \leq m_2 = 600\,\text{kg}$. Furthermore, the delay due to valve movement is not greater than $\tau \leq 12\,\text{s}$. Hence, the parameters of the transfer function in Eq. (69) fall within the following boundaries.

$$\begin{aligned} 1.33 \cdot 10^{-5} &\leq K \leq 2.67 \cdot 10^{-5} \\ 5.31 \cdot 10^{-5} &\leq a \leq 1.02 \cdot 10^{-4} \\ 6.28 \cdot 10^{-3} &\leq b \leq 6.57 \cdot 10^{-3} \end{aligned} \tag{74}$$

By taking into account Eqs. (7)–(9), (15) and (18), the representing transfer function given in Eq. (75) is derived.

$$G_{P0}(s) = \frac{2.00 \cdot 10^{-5}}{(s + 6.97 \cdot 10^{-5})(s + 6.42 \cdot 10^{-3})} \tag{75}$$

From here on, the subsequent steps to achieving suitable controller parameters are exactly the same as in the previous case study, which is presented in detail in Section 5. The only difference is that the PSO algorithm uses more appropriate parameters, according to the expected solution (see Eq. (76)).

$$\begin{aligned} x_{minmax} &= \begin{bmatrix} 1 & 20,000 \\ 0.1 & 100 \\ 0.1 & 100 \end{bmatrix} \\ v_{minmax} &= \begin{bmatrix} -500 & 500 \\ -3 & 3 \\ -3 & 3 \end{bmatrix} \end{aligned} \tag{76}$$

The resulting optimized parameters are given in Eq. (77).

$$\begin{aligned} K_P &= 72.34 \\ T_I &= 10,548 \\ T_D &= 33.00 \\ T_f &= 6.60 \end{aligned} \tag{77}$$

$$a = \frac{-\left(\frac{\lambda S}{mc} + \frac{\lambda S}{m_w c_w} + \frac{\lambda_0 S_0}{m_w c_w}\right)}{2} + \frac{\sqrt{\left(\frac{\lambda S}{mc}\right)^2 + \left(\frac{\lambda S}{m_w c_w}\right)^2 + \left(\frac{\lambda_0 S_0}{m_w c_w}\right)^2 + 2\left(\frac{\lambda^2 S^2}{m c m_w c_w} - \frac{\lambda S \lambda_0 S_0}{m c m_w c_w} + \frac{\lambda S \lambda_0 S_0}{m_w^2 c_w^2}\right)}}{2} \tag{71}$$

$$b = \frac{-\left(\frac{\lambda S}{mc} + \frac{\lambda S}{m_w c_w} + \frac{\lambda_0 S_0}{m_w c_w}\right)}{2} - \frac{\sqrt{\left(\frac{\lambda S}{mc}\right)^2 + \left(\frac{\lambda S}{m_w c_w}\right)^2 + \left(\frac{\lambda_0 S_0}{m_w c_w}\right)^2 + 2\left(\frac{\lambda^2 S^2}{m c m_w c_w} - \frac{\lambda S \lambda_0 S_0}{m c m_w c_w} + \frac{\lambda S \lambda_0 S_0}{m_w^2 c_w^2}\right)}}{2} \tag{72}$$

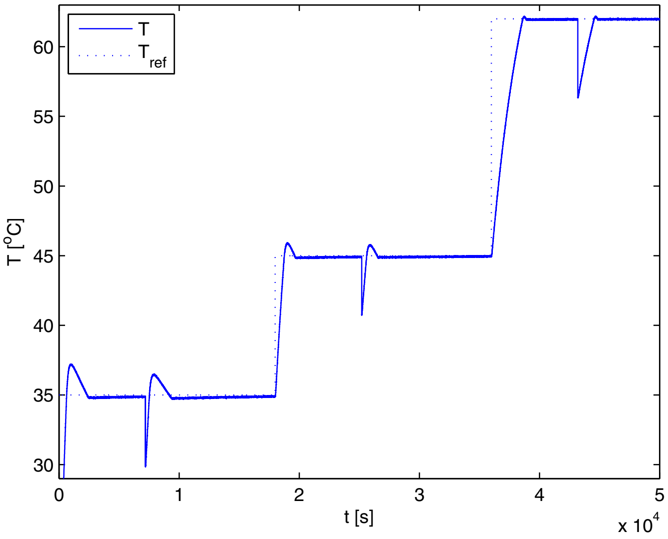

Fig. 18. Closed-loop control of the batch reactor (core temperature *T*). The abrupt drops of temperature are due to ingredients that are added during the process.

The obtained controller was tested on the batch reactor. The results are shown in Figs. 18 and 19. The reference recipe prescribes that the reactor core should first be heated to 35 °C. 130 kg of fresh ingredients are injected into the core. Next, the core is heated to 45 °C and 80 kg of ingredients are added. Finally, the core temperature is brought up to 62 °C and the final batch of ingredients (90 kg) is injected into the reactor. The abrupt drops of temperature are due to freshly injected ingredients that are added during the process, as they have roughly the same temperature as the surroundings. Due to measurement noise the depicted signals are not smooth.

The results indicate that the reference-trajectory tracking is satisfactory within the whole operating range and that the robustness requirements are satisfied as well. As in the previous section, in order to further analyze the proposed approach with regard to its repeatability and convergence properties, the algorithm has been run 30 times consecutively. The globally best value of the fitness function F_{gBest} with regard to the iteration step *iter* is depicted in Fig. (20). The statistics of the consecutive algorithm runs are given in Table 5.

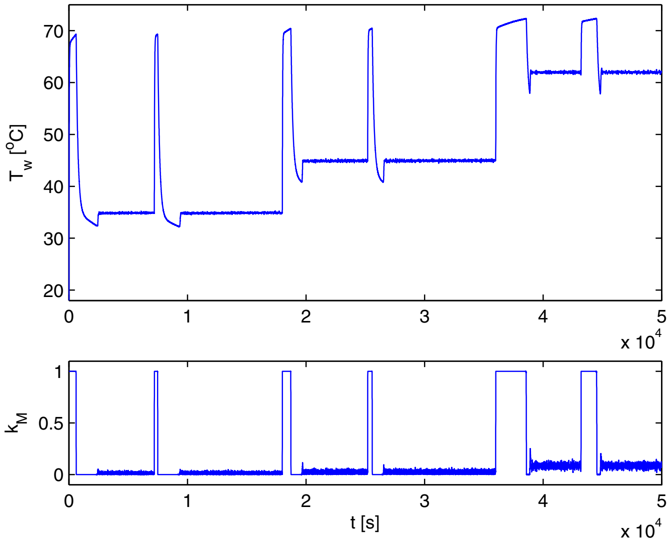

Fig. 19. Closed-loop control of the batch reactor (water jacket temperature T_w, mixing valve position k_M).

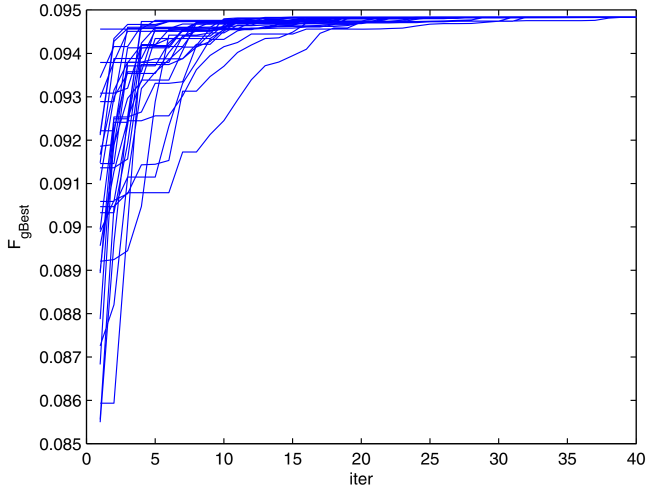

Fig. 20. Globally best value of the fitness function with regard to the iteration step for 30 consecutive algorithm runs.

Table 5
Mean value, standard deviation, normalized standard deviation of the results of the consecutive algorithm runs.

Parameter (x)	$\bar{x}$	$\sigma(x)$	$\sigma(x)/\bar{x}$
K_P	72.0	0.0275	0.037
T_I	11,029	2003	0.18
T_D	33.2	0.159	0.0048
$F_{gBest,fin}$	0.0948	$3.5 \cdot 10^{-6}$	$3.7 \cdot 10^{-5}$

The results of the repeated runs once again lead to the conclusion that the proposed framework is useful for reaching the optimal solution with regard to the predefined fitness function. In this case, the solutions were obtained on average in 184 s, with the worst case requiring 186 s.

6. Conclusion

The paper proposes a framework, which involves using an interval model for describing the uncertain or variable dynamics of the process. The framework employs a particle swarm optimization algorithm for obtaining the best performing PID controller with regard to several possible control-performance criteria defined in the fitness function, but at the same time taking into account the complementary sensitivity function constraints that ensure robustness within the bounds of the uncertain parameters' intervals. Hence, the presented approach enables a simple, computationally tractable and efficient constrained optimization solution for tuning the parameters of the robust controller, while considering the eventual gain, pole, zero and time-delay uncertainties defined using an interval model of the controlled process.

When implementing various optimization approaches one cannot be absolutely positive that the optimum solution will always be reached. However, in case of the proposed approach, the obtained results from the consecutive algorithm runs stand in favor of such conclusion quite strongly. On the other hand, a more advanced global optimization algorithm could also be implemented in order to possibly improve the convergence and computational complexity aspects. However, the advantages would be questionable whereas the approach could become significantly more complicated. Nonetheless, further research is needed to support the potential improvements in this regard.

To sum up, the results of various experiments indicate that the proposed framework yields favorable control performance while assuring robust stability and safe closed-loop operation within the prescribed uncertainty constraints, even when dealing with relatively complex system dynamics and when considerable process uncertainties are expected. It is shown that the controller performance is adequate only when the relative system perturbations are taken into account in the design phase, as proposed in the paper.

Appendix A. PSO pseudo code

The PSO algorithm implemented in the approach can be comprehensively described using pseudo code for easier understanding. The notation used in the pseudo code is given below:

- x_i: position of ith particle;
- v_i: velocity of ith particle;
- F_{gBest}: global best fitness function;
- $F_{pBest,i}$: best fitness function of ith particle.

The PSO algorithm can be divided into the following steps.

```
Set the algorithm parameters $x_{minmax}$, $v_{minmax}$, $n_{iter}$, $n_{part}$, $\omega$, $c_1$ and $c_2$.
Randomly initialize particle positions $x_i$ ; $\forall i \in \{1, \ldots, n_{part}\}$.
Randomly initialize particle velocities $v_i$; $\forall i \in \{1, \ldots, n_{part}\}$.
Initialize global best fitness function $F_{gBest} = -\infty$.
Initialize best fitness function for each particle $F_{pBest,i} = -\infty$ ; $\forall i \in \{1, \ldots, n_{part}\}$.
for $iter = 1$ to $n_{iter}$
  for $i = 1$ to $n_{part}$
    Calculate controller for the current particle $x_i$.
    if Closed-loop system is stable then
      Calculate the considered performance parameters – e. g., sensitivity function $S(j\omega)$, complementary sensitivity function $T(j\omega)$, bandwidth $f_b$ of the closed-loop system, settling time $T_{ST,2\%}$ of the closed-loop system.
      Calculate the value of fitness function $F$.
      if $F > F_{pBest,i}$ then
        $F_{pBest,i} = F$
        $x_{pBest,i} = x_i$
        if $F > F_{gBest}$ then
          $F_{gBest} = F$
          $x_{gBest} = x_i$
        end if
      end if
    end if
    Calculate new particle velocity $v_i = \omega \cdot v_i + c_1 \cdot rand \cdot (x_{pBest,i} - x_i) + c_2 \cdot rand \cdot (x_{gBest} - x_i)$.
    Calculate new particle position $x_i = x_i + v_i$.
  end for
end for
return $x_{gBest}$
```

Appendix B. Derivation of the representing system parameters

Assume the system is given in the factorized form, as shown in Eq. (78).

$$G_P(s) = K_{DC} \cdot \frac{\prod_{j=1}^{m}\left(1+\frac{s}{b_j}\right)}{\prod_{i=1}^{n}\left(1+\frac{s}{a_i}\right)} \cdot e^{-\tau s} \tag{78}$$

In the following subsections the parameters of the representing system when dealing with gain, delay, zero and pole uncertainty are derived.

B.1. Gain uncertainty

In this case, the gain of the system is assumed to be uncertain (parameter K_{DC} of the system $G_P(s)$), as given in Eq. (79). Here, $G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected by any uncertainties.

$$\begin{aligned} &G_P(s) = K_{DC} \cdot G_{Pfix}(s);\\ &G_{Pfix}(s) = \frac{\prod_{j=1}^{m}\left(1+\dfrac{s}{b_{fix,j}}\right)}{\prod_{i=1}^{n}\left(1+\dfrac{s}{a_{fix,i}}\right)} \cdot e^{-\tau s},\\ &K_{DC1} < K_{DC} < K_{DC2} \end{aligned} \tag{79}$$

In order to ensure that the relative system perturbations due to the uncertain parameter are as small as possible, the representing system $G_{P0}(s)$ should be defined as given in Eq. (80).

$$G_{P0}(s) = K_{DC0} \cdot G_{Pfix}(s) \tag{80}$$

The perturbation of the representing system can be calculated as in Eq. (81).

$$\begin{aligned} \left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{K_{DC}\in[K_{DC1},K_{DC2}]} &= \left|\frac{G_P(j\omega)-G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|\\ \left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{K_{DC}\in[K_{DC1},K_{DC2}]} &= \left|\frac{K_{DC}\cdot G_{Pfix}(j\omega)-K_{DC0}\cdot G_{Pfix}(j\omega)}{K_{DC0}\cdot G_{Pfix}(j\omega)}\right|\\ \left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{K_{DC}\in[K_{DC1},K_{DC2}]} &= \left|\frac{K_{DC}-K_{DC0}}{K_{DC0}}\right| \end{aligned} \tag{81}$$

Assume the unknown x defined in Eq. (82) is to be established.

$$K_{DC0} = K_{DC1} + x, \text{ where } x \in [0, K_{DC2} - K_{DC1}] \tag{82}$$

There are two cases that need to be analyzed separately.

- $K_{DC} \to K_{DC1}$:

$$\begin{aligned} \left|\frac{K_{DC}-K_{DC0}}{K_{DC0}}\right| &= \left|\frac{K_{DC1}-(K_{DC1}+x)}{K_{DC1}+x}\right|\\ \left|\frac{K_{DC}-K_{DC0}}{K_{DC0}}\right| &= \left|\frac{-x}{K_{DC1}+x}\right|\\ \left|\frac{K_{DC}-K_{DC0}}{K_{DC0}}\right| &= \frac{x}{K_{DC1}+x} \end{aligned} \tag{83}$$

$$\begin{aligned} \frac{d\dfrac{x}{K_{DC1}+x}}{dx} &= \frac{K_{DC1}}{(K_{DC1}+x)^2}\\ \frac{d\dfrac{x}{K_{DC1}+x}}{dx} &> 0 \end{aligned} \tag{84}$$

Eq. (84) proves that the function in Eq. (83) is monotonically increasing with regard to parameter x.

- $K_{DC} \to K_{DC2}$:

$$\begin{aligned} \left|\frac{K_{DC}-K_{DC0}}{K_{DC0}}\right| &= \left|\frac{K_{DC2}-(K_{DC1}+x)}{K_{DC1}+x}\right|\\ \left|\frac{K_{DC}-K_{DC0}}{K_{DC0}}\right| &= \left|\frac{K_{DC2}-K_{DC1}-x}{K_{DC1}+x}\right|\\ \left|\frac{K_{DC}-K_{DC0}}{K_{DC0}}\right| &= \frac{K_{DC2}-K_{DC1}-x}{K_{DC1}+x} \end{aligned} \tag{85}$$

$$\begin{aligned} &\frac{d\dfrac{K_{DC2}-K_{DC1}-x}{K_{DC1}+x}}{dx} = \frac{-K_{DC2}}{(K_{DC1}+x)^2} \\ &\frac{d\dfrac{K_{DC2}-K_{DC1}-x}{K_{DC1}+x}}{dx} < 0 \end{aligned} \tag{86}$$

Eq. (86) proves that the function in Eq. (85) is monotonically decreasing with regard to parameter x.

Taking into account the equations above, the maximum possible system perturbation is limited to the smallest value when Eq. (87) holds.

$$\begin{aligned} &\frac{x}{K_{DC1}+x} = \frac{K_{DC2}-K_{DC1}-x}{K_{DC1}+x} \\ &x = \frac{K_{DC2}-K_{DC1}}{2} \end{aligned} \tag{87}$$

Therefore, the resulting parameter K_{DC0} can be derived as defined in Eq. (88).

$$\begin{aligned} &K_{DC0} = K_{DC1}+x \\ &K_{DC0} = \frac{K_{DC1}+K_{DC2}}{2} \end{aligned} \tag{88}$$

B.2. Delay uncertainty

In this case, the time-delay of the system is assumed to be uncertain (parameter τ of the system $G_P(s)$), as given in Eq. (89). Here, $G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected by any uncertainties.

$$\begin{aligned} &G_P(s) = e^{-\tau s}\cdot G_{Pfix}(s); \\ &G_{Pfix}(s) = K_{DC,fix}\cdot\frac{\prod_{j=1}^{m}\left(1+\dfrac{s}{b_{fix,j}}\right)}{\prod_{i=1}^{n}\left(1+\dfrac{s}{a_{fix,i}}\right)} \\ &\tau_0 \le \tau \le \tau_{\max} \end{aligned} \tag{89}$$

Assume the representing system $G_{P0}(s)$ is defined as given in Eq. (90).

$$G_{P0}(s) = e^{-\tau_0 s}\cdot G_{Pfix}(s) \tag{90}$$

The perturbation of the representing system can be calculated as in Eq. (91).

$$\begin{aligned} &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{\tau\in[\tau_0,\tau_{\max}]} = \left|\frac{G_P(j\omega)-G_{P0}(j\omega)}{G_{P0}(j\omega)}\right| \\ &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{\tau\in[\tau_0,\tau_{\max}]} = \left|\frac{e^{-j\omega\tau}\cdot G_{Pfix}(j\omega)-e^{-j\omega\tau_0}\cdot G_{Pfix}(j\omega)}{e^{-j\omega\tau_0}\cdot G_{Pfix}(j\omega)}\right| \\ &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{\tau\in[\tau_0,\tau_{\max}]} = |e^{-j\omega(\tau-\tau_0)}-1| \\ &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{\tau\in[\tau_0,\tau_{\max}]} = |e^{-j\omega\Delta\tau}-1| \end{aligned} \tag{91}$$

B.3. Zero uncertainty

In this case, the a zero of the system is assumed to be uncertain (parameter b of the system $G_P(s)$), as given in Eq. (92). Here, $G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected by any uncertainties.

$$\begin{aligned} &G_P(s) = (1+\frac{s}{b})\cdot G_{Pfix}(s); \\ &G_{Pfix}(s) = K_{DC,fix}\cdot\frac{\prod_{j=2}^{m}\left(1+\dfrac{s}{b_{fix,j}}\right)}{\prod_{i=1}^{n}\left(1+\dfrac{s}{a_{fix,i}}\right)}\cdot e^{-\tau s}, \\ &b_1 < b < b_2 \end{aligned} \tag{92}$$

In order to ensure that the relative system perturbations due to the uncertain parameter are as small as possible, the representing system $G_{P0}(s)$ should be defined as given in Eq. (93).

$$G_{P0}(s) = \left(1+\frac{s}{b_0}\right)\cdot G_{Pfix}(s) \tag{93}$$

The perturbation of the representing system can be calculated as in Eq. (94).

$$\begin{aligned} &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{b\in[b_1,b_2]} = \left|\frac{G_P(j\omega)-G_{P0}(j\omega)}{G_{P0}(j\omega)}\right| \\ &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{b\in[b_1,b_2]} = \left|\frac{\left(1+\dfrac{j\omega}{b}\right)\cdot G_{Pfix}(j\omega)-\left(1+\dfrac{j\omega}{b_0}\right)\cdot G_{Pfix}(j\omega)}{\left(1+\dfrac{j\omega}{b_0}\right)\cdot G_{Pfix}(j\omega)}\right| \\ &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{b\in[b_1,b_2]} = \left|\frac{\left(1+\dfrac{j\omega}{b}\right)-\left(1+\dfrac{j\omega}{b_0}\right)}{\left(1+\dfrac{j\omega}{b_0}\right)}\right| \\ &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{b\in[b_1,b_2]} = \left|\frac{\dfrac{j\omega}{b}-\dfrac{j\omega}{b_0}}{1+\dfrac{j\omega}{b_0}}\right| \\ &\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{b\in[b_1,b_2]} = \left|\frac{j\omega\left(\dfrac{b_0}{b}-1\right)}{b_0+j\omega}\right| \end{aligned} \tag{94}$$

Assume the unknown x defined in Eq. (95) is to be established.

$$b_0 = b_1+x, \quad \text{where} \quad x\in[0, b_2-b_1] \tag{95}$$

There are two cases that need to be analyzed separately.

- $b\to b_1$:

$$\begin{aligned} &\left|\frac{j\omega(\dfrac{b_0}{b}-1)}{b_0+j\omega}\right| = \left|\frac{j\omega(\dfrac{b_1+x}{b_1}-1)}{b_1+x+j\omega}\right| \\ &\left|\frac{j\omega(\dfrac{b_0}{b}-1)}{b_0+j\omega}\right| = \left|\frac{j\omega x}{b_1\cdot(b_1+x+j\omega)}\right| \\ &\left|\frac{j\omega(\dfrac{b_0}{b}-1)}{b_0+j\omega}\right| = \frac{\omega x}{b_1\cdot\sqrt{(b_1+x)^2+\omega^2}} \end{aligned} \tag{96}$$

$$
\begin{aligned}
\frac{d\dfrac{\omega x}{b_1\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{\omega\cdot b_1\cdot\sqrt{(b_1+x)^2+\omega^2}-\dfrac{\omega x\cdot b_1\cdot(2(b_1+x))}{2\cdot\sqrt{(b_1+x)^2+\omega^2}}}{\left(b_1\cdot\sqrt{(b_1+x)^2+\omega^2}\right)^2}\\
\frac{d\dfrac{\omega x}{b_1\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{\omega\cdot((b_1+x)^2+\omega^2)-\omega x\cdot(b_1+x)}{b_1\cdot((b_1+x)^2+\omega^2)^{\frac{3}{2}}}\\
\frac{d\dfrac{\omega x}{b_1\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{\omega\cdot(b_1^2+2b_1x+x^2+\omega^2-b_1x-x^2)}{b_1\cdot((b_1+x)^2+\omega^2)^{\frac{3}{2}}}\\
\frac{d\dfrac{\omega x}{b_1\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{\omega\cdot(b_1\cdot(b_1+x)+\omega^2)}{b_1\cdot((b_1+x)^2+\omega^2)^{\frac{3}{2}}}\\
\frac{d\dfrac{\omega x}{b_1\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &\geq 0
\end{aligned}
\tag{97}
$$

Eq. (97) proves that the function in Eq. (96) is monotonically increasing with regard to parameter x.

- $b\rightarrow b_2$:

$$
\begin{aligned}
\left|\frac{j\omega(\frac{b_0}{b}-1)}{b_0+j\omega}\right| &= \left|\frac{j\omega(\frac{b_1+x}{b_2}-1)}{b_1+x+j\omega}\right|\\
\left|\frac{j\omega(\frac{b_0}{b}-1)}{b_0+j\omega}\right| &= \left|\frac{-j\omega(b_2-b_1-x)}{b_2\cdot(b_1+x+j\omega)}\right|\\
\left|\frac{j\omega(\frac{b_0}{b}-1)}{b_0+j\omega}\right| &= \frac{\omega(b_2-b_1-x)}{b_2\cdot\sqrt{(b_1+x)^2+\omega^2}}
\end{aligned}
\tag{98}
$$

$$
\begin{aligned}
\frac{d\dfrac{\omega(b_2-b_1-x)}{b_2\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{-\omega\cdot b_2\cdot\sqrt{(b_1+x)^2+\omega^2}-\dfrac{\omega(b_2-b_1-x)\cdot b_2\cdot(2(b_1+x))}{2\cdot\sqrt{(b_1+x)^2+\omega^2}}}{\left(b_2\cdot\sqrt{(b_1+x)^2+\omega^2}\right)^2}\\
\frac{d\dfrac{\omega(b_2-b_1-x)}{b_2\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{-\omega\cdot((b_1+x)^2+\omega^2)-\omega\cdot(b_2-b_1-x)\cdot(b_1+x)}{b_2\cdot((b_1+x)^2+\omega^2)^{\frac{3}{2}}}\\
\frac{d\dfrac{\omega(b_2-b_1-x)}{b_2\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{-\omega\cdot(b_1^2+2b_1x+x^2+\omega^2+b_1b_2-b_1^2-b_1x+b_2x-b_1x-x^2)}{b_2\cdot((b_1+x)^2+\omega^2)^{\frac{3}{2}}}\\
\frac{d\dfrac{\omega(b_2-b_1-x)}{b_2\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &= \frac{-\omega\cdot(b_2\cdot(b_1+x)+\omega^2)}{b_2\cdot((b_1+x)^2+\omega^2)^{\frac{3}{2}}}\\
\frac{d\dfrac{\omega(b_2-b_1-x)}{b_2\cdot\sqrt{(b_1+x)^2+\omega^2}}}{dx} &\leq 0
\end{aligned}
\tag{99}
$$

Eq. (99) proves that the function in Eq. (98) is monotonically decreasing with regard to parameter x.

Taking into account the equations above, the maximum possible system perturbation can be limited to the smallest value when Eq. (100) holds.[3]

$$
\begin{aligned}
\frac{\omega x}{b_1\cdot\sqrt{(b_1+x)^2+\omega^2}} &= \frac{\omega(b_2-b_1-x)}{b_2\cdot\sqrt{(b_1+x)^2+\omega^2}}\\
x &= \frac{b_1\cdot(b_2-b_1)}{b_1+b_2}
\end{aligned}
\tag{100}
$$

Therefore, the resulting parameter b_0 can be derived as defined in Eq. (101).

$$
\begin{aligned}
b_0 &= b_1+x\\
b_0 &= \frac{2b_1b_2}{b_1+b_2}
\end{aligned}
\tag{101}
$$

B.4. Pole uncertainty

In this case, the a pole of the system is assumed to be uncertain (parameter a of the system $G_P(s)$), as given in Eq. (102). Here, $G_{Pfix}(s)$ represents an arbitrary multiplicatory system with fixed parameters that are assumed not to be affected by any uncertainties.

$$
\begin{aligned}
G_P(s) &= \frac{1}{1+\dfrac{s}{a}}\cdot G_{Pfix}(s);\\
G_{Pfix}(s) &= K_{DC,fix}\cdot\frac{\prod_{j=1}^{m}\left(1+\dfrac{s}{b_{fix,j}}\right)}{\prod_{i=2}^{n}\left(1+\dfrac{s}{a_{fix,i}}\right)}\cdot e^{-\tau s},\\
a_1 &< a < a_2
\end{aligned}
\tag{102}
$$

[3] Note that Eq. (100) always holds for $\omega=0$.

In order to ensure that the relative system perturbations due to the uncertain parameter are as small as possible, the representing system $G_{P0}(s)$ should be defined as given in Eq. (103).

$$G_{P0}(s)=\frac{1}{1+\dfrac{s}{a_0}}\cdot G_{Pfix}(s) \tag{103}$$

The perturbation of the representing system can be calculated as in Eq. (104).

$$\begin{aligned}
\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{a\in[a_1,a_2]} &= \left|\frac{G_P(j\omega)-G_{P0}(j\omega)}{G_{P0}(j\omega)}\right| \\
\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{a\in[a_1,a_2]} &= \left|\frac{\dfrac{1}{1+\dfrac{j\omega}{a}}\cdot G_{Pfix}(j\omega)-\dfrac{1}{1+\dfrac{j\omega}{a_0}}\cdot G_{Pfix}(j\omega)}{\dfrac{1}{1+\dfrac{j\omega}{a_0}}\cdot G_{Pfix}(j\omega)}\right| \\
\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{a\in[a_1,a_2]} &= \left|\frac{\dfrac{1}{1+\dfrac{j\omega}{a}}-\dfrac{1}{1+\dfrac{j\omega}{a_0}}}{\dfrac{1}{1+\dfrac{j\omega}{a_0}}}\right| \\
\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{a\in[a_1,a_2]} &= \left|\frac{a_0a+aj\omega-a_0a-a_0j\omega}{a_0\cdot(a+j\omega)}\right| \\
\left|\frac{\Delta G_{P0}(j\omega)}{G_{P0}(j\omega)}\right|_{a\in[a_1,a_2]} &= \left|\frac{j\omega\cdot(a-a_0)}{a_0\cdot(a+j\omega)}\right|
\end{aligned} \tag{104}$$

Assume the unknown x defined in Eq. (105) is to be established.

$$a_0=a_1+x,\quad \text{where}\quad x\in[0,a_2-a_1] \tag{105}$$

There are two cases that need to be analyzed separately.

- $a\to a_1$:

$$\begin{aligned}
\left|\frac{j\omega\cdot(a-a_0)}{a_0\cdot(a+j\omega)}\right| &= \left|\frac{j\omega\cdot(a_1-(a_1+x))}{(a_1+x)\cdot(a_1+j\omega)}\right| \\
\left|\frac{j\omega\cdot(a-a_0)}{a_0\cdot(a+j\omega)}\right| &= \left|\frac{-j\omega x}{a_1\cdot(a_1+x)+j\omega\cdot(a_1+x)}\right| \\
\left|\frac{j\omega\cdot(a-a_0)}{a_0\cdot(a+j\omega)}\right| &= \frac{\omega x}{(a_1+x)\cdot\sqrt{a_1^2+\omega^2}}
\end{aligned} \tag{106}$$

$$\begin{aligned}
\frac{d\dfrac{\omega x}{(a_1+x)\cdot\sqrt{a_1^2+\omega^2}}}{dx} &= \frac{\omega\cdot(a_1+x)\cdot\sqrt{a_1^2+\omega^2}-\omega x\cdot\sqrt{a_1^2+\omega^2}}{((a_1+x)\cdot\sqrt{a_1^2+\omega^2})^2} \\
\frac{d\dfrac{\omega x}{(a_1+x)\cdot\sqrt{a_1^2+\omega^2}}}{dx} &= \frac{\omega\cdot\sqrt{a_1^2+\omega^2}\cdot((a_1+x)-x)}{((a_1+x)\cdot\sqrt{a_1^2+\omega^2})^2} \\
\frac{d\dfrac{\omega x}{(a_1+x)\cdot\sqrt{a_1^2+\omega^2}}}{dx} &= \frac{\omega a_1}{(a_1+x)^2\cdot\sqrt{a_1^2+\omega^2}} \\
\frac{d\dfrac{\omega x}{(a_1+x)\cdot\sqrt{a_1^2+\omega^2}}}{dx} &\geq 0
\end{aligned} \tag{107}$$

Eq. (107) proves that the function in Eq. (106) is monotonically increasing with regard to parameter x.

- $a\to a_2$:

$$\begin{aligned}
\left|\frac{j\omega\cdot(a-a_0)}{a_0\cdot(a+j\omega)}\right| &= \left|\frac{j\omega\cdot(a_2-(a_1+x))}{(a_1+x)\cdot(a_2+j\omega)}\right| \\
\left|\frac{j\omega\cdot(a-a_0)}{a_0\cdot(a+j\omega)}\right| &= \left|\frac{j\omega\cdot(a_2-a_1-x)}{a_2\cdot(a_1+x)+j\omega\cdot(a_1+x)}\right| \\
\left|\frac{j\omega\cdot(a-a_0)}{a_0\cdot(a+j\omega)}\right| &= \frac{\omega\cdot(a_2-a_1-x)}{(a_1+x)\cdot\sqrt{a_2^2+\omega^2}}
\end{aligned} \tag{108}$$

$$\begin{aligned}
\frac{d\dfrac{\omega\cdot(a_2-a_1-x)}{(a_1+x)\cdot\sqrt{a_2^2+\omega^2}}}{dx} &= \frac{-\omega\cdot(a_1+x)\cdot\sqrt{a_2^2+\omega^2}-\omega\cdot(a_2-a_1-x)\cdot\sqrt{a_2^2+\omega^2}}{((a_1+x)\cdot\sqrt{a_2^2+\omega^2})^2} \\
\frac{d\dfrac{\omega\cdot(a_2-a_1-x)}{(a_1+x)\cdot\sqrt{a_2^2+\omega^2}}}{dx} &= \frac{-\omega\cdot\sqrt{a_2^2+\omega^2}\cdot((a_1+x)+(a_2-a_1-x))}{((a_1+x)\cdot\sqrt{a_1^2+\omega^2})^2} \\
\frac{d\dfrac{\omega\cdot(a_2-a_1-x)}{(a_1+x)\cdot\sqrt{a_2^2+\omega^2}}}{dx} &= \frac{-\omega a_2}{(a_1+x)^2\cdot\sqrt{a_1^2+\omega^2}} \\
\frac{d\dfrac{\omega x}{(a_1+x)\cdot\sqrt{a_1^2+\omega^2}}}{dx} &\leq 0
\end{aligned} \tag{109}$$

Eq. (109) proves that the function in Eq. (108) is monotonically decreasing with regard to parameter x.

Taking into account the equations above, the maximum possible system perturbation can be limited to the smallest value when Eq. (110) holds.[4]

$$\begin{aligned}
&\frac{\omega x}{(a_1+x)\cdot\sqrt{a_1^2+\omega^2}}=\frac{\omega\cdot(a_2-a_1-x)}{(a_1+x)\cdot\sqrt{a_2^2+\omega^2}} \\
&x\left(\sqrt{a_1^2+\omega^2}+\sqrt{a_2^2+\omega^2}\right)=(a_2-a_1)\sqrt{a_1^2+\omega^2} \\
&x=\frac{(a_2-a_1)\sqrt{a_1^2+\omega^2}}{\sqrt{a_1^2+\omega^2}+\sqrt{a_2^2+\omega^2}} \\
&x=\frac{(a_2-a_1)}{1+\sqrt{\dfrac{a_2^2+\omega^2}{a_1^2+\omega^2}}}
\end{aligned} \tag{110}$$

Therefore, the resulting parameter a_0 can be derived as defined in Eq. (111).

$$\begin{aligned}
a_0&=a_1+x \\
a_0&=a_1+\frac{a_2-a_1}{1+\sqrt{\dfrac{a_2^2+\omega^2}{a_1^2+\omega^2}}}
\end{aligned} \tag{111}$$

This means that a_0 lies in the interval bounded by the low-frequency solution ($\omega\to 0$) and the high-frequency solution ($\omega\to\infty$) as defined in Eq. (112).

$$a_0\in\left[\frac{2a_1a_2}{a_1+a_2},\frac{a_1+a_2}{2}\right] \tag{112}$$

References

[1] J.G. Ziegler, N.B. Nichols, Optimum settings for automatic controllers, Trans. ASME 64 (1942) 759–768.

[2] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micromachine and Human Science, vol. 1, 1995, pp. 39–43.

[3] K.E. Parsopoulos, M.N. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, Nat. Comput. 1 (2–3) (2002) 235–306, http://dx.doi.org/10.1023/A:1016568309421.

[4] J. Moore, R. Chapman, Application of particle swarm to multiobjective optimization, Technical report, Department of Computer Science and Software Engineering, Auburn University, 1999.

[5] C.A.C. Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation, 2002, pp. 1051–1056.

[4] Note that Eq. (110) always holds for $\omega=0$.

[6] M. Curtis, A. Lewis, Reduction of computational load for MOPSO, Proc. Comput. Sci. 51 (2015) 2789–2793.
[7] C.C.A. Coello, M. Reyes-Sierra, Multi-objective particle swarm optimizers: a survey of the state-of-the-art, Int. J. Comput. Intell. Res. 2 (3) (2006) 287–308.
[8] Y. Gong, W. Chen, Z. Zhan, J. Zhang, Y. Li, Q. Zhang, J. Li, Distributed evolutionary algorithms and their models: a survey of the state-of-the-art, Appl. Soft Comput. 34 (2015) 286–300.
[9] S. Dehuri, S.-B. Cho, Multi-criterion pareto based particle swarm optimized polynomial neural network for classification: a review and state-of-the-art, Comput. Sci. Rev. 3 (1) (2009) 19–40.
[10] M.Q. Raza, A. Khosravi, A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings, Renew. Sustain. Energy Rev. 50 (2015) 1352–1372.
[11] H. Borhanazad, S. Mekhilef, V.G. Ganapathy, M. Modiri-Delshad, A. Mirtaheri, Optimization of micro-grid system using MOPSO, Renew. Energy 71 (2014) 295–306.
[12] J. Evora, J.J. Hernandez, M. Hernandez, A MOPSO method for direct load control in smart grid, Expert Syst. Appl. 42 (21) (2015) 7456–7465.
[13] S. Lalwani, S. Singhal, R. Kumar, N. Gupta, A comprehensive survey: applications of multi-objective particle swarm optimization (MOPSO) algorithm, Trans. Combinatorics 2 (1) (2013) 39–101.
[14] K.E. Parsopoulos, D.K. Tasoulis, M.N. Vrahatis, Multiobjective optimization using parallel vector evaluated particle swarm optimization, in: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, ACTA Press, 2004, pp. 823–828.
[15] S. Omkar, D. Mudigere, G.N. Naik, S. Gopalakrishnan, Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures, Comput. Struct. 86 (1–2) (2008) 1–14.
[16] Y. Shi, R.C. Eberhart, Fuzzy adaptive particle swarm optimization, in: Proceedings of the Congress on Evolutionary Computation, 2001, pp. 101–106.
[17] Z. Liu, C. Mao, J. Luo, Y. Zhang, C.P. Chen, A three-domain fuzzy wavelet network filter using fuzzy {PSO} for robotic assisted minimally invasive surgery, Knowl. Based Syst. 66 (2014) 13–27.
[18] S.H. Ling, K.Y. Chan, F.H.F. Leung, F. Jiang, H. Nguyen, Quality and robustness improvement for real world industrial systems using a fuzzy particle swarm optimization, Eng. Appl. Artif. Intell. 47 (2016) 68–80.
[19] I. Kaya, Tuning PI controllers for stable processes with specifications on gain and phase margins, ISA Trans. 43 (2) (2004) 297–304.
[20] P. Airikka, Robust predictive PI controller tuning, in: Preprints of the 19th World Congress the International Federation of Automatic Control, Cape Town, South Africa, 2014, pp. 9301–9306.
[21] Y.-J. Wang, Graphical computation of gain and phase margin specifications-oriented robust PID controllers for uncertain systems with time-varying delay, J. Process Control 21 (4) (2011) 475–488.
[22] Y.-J. Wang, Determination of all feasible robust PID controllers for open-loop unstable plus time delay processes with gain margin and phase margin specifications, ISA Trans. 53 (2) (2014) 628–646.
[23] D.-J. Wang, A PID controller set of guaranteeing stability and gain and phase margins for time-delay systems, J. Process Control 22 (7) (2012) 1298–1306.
[24] K. Li, Robust IMC-PID design for optimal closed-loop response with specified gain and phase margins for soptd systems, in: I.A. Karimi, R. Srinivasan (Eds.), 11th International Symposium on Process Systems Engineering, vol. 31 of Computer Aided Chemical Engineering, Elsevier, 2012, pp. 1657–1661.
[25] D.H. Kim, Hybrid gabf based intelligent {PID} controller tuning for {AVR} system, Appl. Soft Comput. 11 (1) (2011) 11–22.
[26] R. Precup, S. Preitl, G. Faur, PI predictive fuzzy controllers for electrical drive speed control: methods and software for stable development, Comput. Ind. 52 (3) (2003) 253–270.
[27] R. Precup, M.L. Tomescu, S. Preitl, Lorenz system stabilization using fuzzy controllers, Int. J. Comput. Commun. Control 2 (3) (2007) 279–287.
[28] M. Radac, R. Precup, E.M.E.M. Petriu, S. Preitl, Application of IFT and SPSA to Servo system control, IEEE Trans. Neural Netw. 22 (12, 2) (2011) 2363–2375.
[29] P. Puri, S. Gosh, A hybrid optimization approach for PI controller tuning based on gain and phase margin specifications, Swarm Evol. Comput. 8 (2013) 69–78.
[30] T.-H. Kim, I. Maruta, T. Sugie, Robust PID controller tuning based on the constrained particle swarm optimization, Automatica 44 (4) (2008) 1104–1110.
[31] K. Åström, H. Panagopoulos, T. Hägglund, Design of PI controllers based on non-convex optimization, Automatica 34 (5) (1998) 585–601.
[32] P. Hušek, Robust PI controller design with respect to fuzzy sensitivity margins, Appl. Soft Comput. 13 (4) (2013) 2037–2044.
[33] M.I. Solihin, L.F. Tack, M.L. Kean, Tuning of PID controller using particle swarm optimization (PSO), in: Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, 2011, pp. 458–461.
[34] S.J. Bassi, M.K. Mishra, E.E. Omizegba, Automatic tuning of proportional integral derivative (PID) controller using particle swarm optimization (PSO) algorithm, Int. J. Artif. Intell. Appl. 2 (4) (2011) 25–32.
[35] T. Azuma, S. Watanbe, A design of PID controllers using FRIT-PSO, in: Proceeding 8th International Conference on Sensing Technology, 2014, pp. 459–464.
[36] S. Bouallgue, J. Haggge, M. Ayadi, M. Benrejeb, Pid-type fuzzy logic controller tuning based on particle swarm optimization, Eng. Appl. Artif. Intell. 25 (3) (2012) 484–493.
[37] K.J. Åström, Control System Design: Lecture Notes for ME 155A, 2002 http://www.cds.caltech.edu/murray/courses/cds101/fa02/caltech/astrom.html.
[38] K.J. Åström, R.M. Murray, Feedback Systems, Princeton University Press, 2012.
[39] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Inf. Process. Lett. 85 (6) (2003) 317–325.
[40] K.E. Parsopoulos, M.N. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, Nat. Comput. Int. J. 1 (2–3) (2002) 235–306.
[41] M. Clerc, J. Kennedy, The particle swarm – explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (1) (2002) 58–73.
[42] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, 1998, pp. 69–73.
[43] G. Karer, G. Mušič, I. Škrjanc, B. Zupančič, Hybrid fuzzy model-based predictive control of temperature in a batch reactor, Comput. Chem. Eng. 31 (2007) 1552–1564.
[44] J. Causa, G. Karer, A. Nú nez, D. Sáez, I. Škrjanc, Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor, Comput. Chem. Eng. 32 (12) (2008) 3254–3263.
[45] I. Škrjanc, Adaptive supervisory predictive control of a hybrid fed-batch reactor with slow actuator, Ind. Eng. Chem. Res. 46 (24) (2007) 8050–8057.
[46] G. Karer, I. Škrjanc, B. Zupančič, Self-adaptive predictive functional control of the temperature in an exothermic batch reactor, Chem. Eng. Process. 47 (2008) 2379–2385.
[47] I. Škrjanc, Self-adaptive supervisory predictive functional control of a hybrid semi-batch reactor with constraints, Chem. Eng. J. 136 (2/3) (2007) 312–319.
[48] G. Karer, G. Mušič, I. Škrjanc, B. Zupančič, Feedforward control of a class of hybrid systems using an inverse model, Math. Comput. Simul. 82 (3) (2011) 414–427.