Optimal Control Systems with Reduced Parametric Sensitivity Based on Particle Swarm Optimization and Simulated Annealing

Radu-Emil Precup*, Radu-Codruț David, Stefan Preitl, Emil M. Petriu, and József K. Tar

Abstract. This chapter discusses theoretical and design aspects for optimal control systems with a reduced parametric sensitivity using Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithms. Sensitivity models with respect to the parametric variations of the controlled process are derived and the optimal control problems are defined. The new objective functions in these optimization problems are integral quadratic performance indices that depend on the control error and squared output sensitivity functions. Different dynamic regimes are considered. Relatively simple PSO and SA optimization algorithms are developed for the minimization of the objective functions, which optimize the control system responses and reduce the sensitivity to parametric variations of the controlled process. Examples of optimization problems encountered in the design of optimal proportional-integral (PI) controllers for a class of second-order processes with integral component are used to validate the proposed methods.

Radu-Emil Precup · Radu-Codruț David · Stefan Preitl
Department of Automation and Applied Informatics, "Politehnica"
University of Timisoara, Bd. V. Parvan 2, RO-300223 Timisoara, Romania
e-mail: radu.precup@aut.upt.ro

Emil M. Petriu
School of Information Technology and Engineering, University of Ottawa, 800 King Edward, Ottawa, ON, K1N 6N5 Canada

József K. Tar
Institute of Intelligent Engineering Systems, Óbuda University, Bécsi út 96/B, H-1034 Budapest, Hungary

* Corresponding author.
Tel.: +40-2564032-26, -29, Fax: +40-2564032-14

M. Köppen et al. (Eds.): Intelligent Comput. Optimization in Eng., SCI 366, pp. 177–207.
springerlink.com

1 Introduction

Many control systems are tuned based on idealized linear or linearized models of the controlled processes. However, industrial processes are subjected to parametric variations of the controlled processes, which result in models that are either nonlinear or only locally linearized around several nominal operating points or trajectories. Therefore, it is necessary to do a sensitivity analysis with respect to the parametric variations of the controlled process.

As shown in [1], the uncontrollable process parametric variations lead to undesirable behavior of the control systems. As reported in the literature, the parametric sensitivity of the control systems can be studied in the frequency domain [2–7] or in the time domain [1,8].

A variety of optimal control applications employing sensitivity models in the objective functions were reported in the literature [9–12]. Objective functions as extended quadratic performance indices were used in the design of Takagi-Sugeno proportional-integral-fuzzy controllers [8,13]. An application of Bellman-Zadeh's approach to decision making in fuzzy environments for multi-criteria optimization problems is presented in [14]. The elimination of the steady-state control error by an augmented state feedback tracking guaranteed cost control is reported in [15]. In a recent review paper [16] Campos and Calado present optimal control approaches to human arm movement control. A method to estimate the minimum variance bounds and the achievable variance bounds for the assessment of the Iterative Learning Control-based batch control systems is presented in [17].

The optimal control methods with reduced parametric sensitivity presented in this chapter use time domain sensitivity models in the objective functions. Building upon Precup's and Preitl's previous work on controller optimization criteria, [8], we propose new objective functions that employ the integrals of weighted squared output sensitivity functions added to the Integral of Squared Error (ISE), the Integral of Absolute Error (IAE), the Integral of Time multiplied by Squared Error (ITSE), and the Integral of Time multiplied by Absolute Error (ITAE) to reduce the effects of the parametric disturbances.

The dynamic regimes with regard to step-type modifications of the reference input and disturbance inputs are considered resulting in additional sensitivity models and objective functions. While the fundamental deviation of a control system relative to its nominal trajectory is described by the control error, the additional deviation can be described by the output sensitivity function in the sensitivity model.

Solving the optimization problems for the usually non-convex objective functions used in many control systems is not a trivial task as it can lead to several local minima. Different solutions such as derivative-free optimization algorithms [18–21], Particle Swarm Optimization (PSO) [22,23] and Simulated Annealing (SA) [24–27] were proposed in the literature to solve these problems.

PSO algorithms have a number of advantages, which make them attractive for the control systems design:

- compact implementation programs,
- computational efficiency,

- search algorithm using objective function values instead of the gradient information,
- they are not bound by conventional deterministic methods constraints such as the linearity, differentiability, convexity, separability or non-existence of constraints,
- very little, if any, solution dependence on the initial states of particles.

In our recent paper [28] we discussed two new PSO algorithms for the optimal design of proportional-integral (PI) controllers for a class of second-order processes with integral component and variable parameters. Other examples of PSO-based designs of robust, adaptive and predictive controllers are presented in [29–32]. Optimal fuzzy controllers and combinations of PSO and fuzzy controllers are presented in [33–35]. Other PSO applications to the design of optimal controllers for power systems, transportation systems, electrical drives and artificial intelligence are presented in [36–40].

SA algorithms have the distinct capability of finding the global minimum of certain objective functions under specific conditions. Several applications of SA for the adaptive and predictive optimal control are discussed in [41–43]. Intelligent SA-based control systems are presented in [44] and [45], and applications to the control of electrical drives and chemical processes are reported in [46–48].

This chapter will discuss new PSO and SA-based algorithms for the design of optimal control systems with reduced parametric sensitivity. The design of an optimal PI controller for a class of second-order processes with integral component [49] will then be presented as a representative case study.

The chapter is structured as follows. Section 2 provides a mathematical framework for the design of optimal control systems with reduced parametric sensitivity. Models of the controlled process and controller, sensitivity models and objective functions are among the most important topics that are discussed. Section 3 focuses on PSO and SA algorithms for the optimization of the controllers with objective functions that have a single variable, β. Useful recommendations are provided for practitioners on how to set the values of the parameters for PSO and SA algorithms. Section 4 is dedicated to the representative case study of optimal designs of PI controllers for a class of second-order processes with integral component. Conclusions and further discussions are presented in Section 5.

2 Framework for Optimal Control Systems Design

The controlled process is represented by the following Single Input-Single Output (SISO) state-space model

$$\begin{aligned} &\dot{\mathbf{x}}_P = \mathbf{f}_P(\mathbf{x}_P, \boldsymbol{\alpha}, u, d), \\ &y = g_P(\mathbf{x}_P, \boldsymbol{\alpha}, d), \\ &\mathbf{x}_P(t_0) = \mathbf{x}_{P,0}, \end{aligned} \tag{1}$$

where: $t_0 \geq 0$ is the initial time moment, $\mathbf{x}_P = [x_{P,1} \quad x_{P,2} \quad \dots \quad x_{P,n}]^T \in R^n$ is the state vector of the controlled process, $\mathbf{x}_{P,0} \in R^n$ is the initial state vector of the controlled process, u is the control signal, d is the disturbance input, y is the controlled output, $\boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_m]^T \in R^m$ is the parameter vector containing the parameters of the controlled process, $\alpha_j, j = \overline{1, m}$, $\mathbf{f}_P : R^{n \times m \times 1 \times 1} \to R^n$, $g_P : R^{n \times m \times 1} \to R$ are functions that are differentiable with respect to $\boldsymbol{\alpha}$ on R^m, T indicates the matrix transposition, and the real argument of the functions, t, $t \geq t_0$, is omitted to simplify the presentation.

The state-space model (1) includes the dynamics of the measuring element(s) as well as of the actuator(s).

Fig. 1 shows the structure of a conventional control system, where: C is the controller, P is the controlled process, r is the reference input, and e is the control error,

$$e = r - y. \tag{2}$$

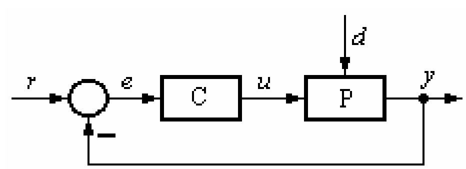

Fig. 1 Control system structure.

The controller is represented by the SISO state-space model

$$\begin{aligned} \dot{\mathbf{x}}_C &= \mathbf{f}_C(\mathbf{x}_C, \beta, e), \\ u &= g_C(\mathbf{x}_C, \beta, e), \\ \mathbf{x}_C(t_0) &= \mathbf{x}_{C,0}, \end{aligned} \tag{3}$$

where: $\mathbf{x}_C = [x_{C,1} \quad x_{C,2} \quad \dots \quad x_{C,p}]^T \in R^p$ is the state vector of the controller, $\mathbf{x}_{C,0} \in R^p$ is the initial state vector of the controller, β is the design parameter and $\mathbf{f}_C : R^{p \times q \times 1} \to R^n$, $g_C : R^{p \times q \times 1} \to R$ are continuous functions.

The majority of linear controllers [50] and many nonlinear controllers including the fuzzy controllers under certain conditions [51] can be expressed in terms of the model (3). However, the convergence of the integrals in the objective functions requires that all controllers should have an integral component in order to ensure the zero steady state of the control error for several disturbance inputs.

To express the state-space model of the control system, the elements of the two state vectors in the models (1) and (3) are grouped in the state vector $\mathbf{x}$ of the control system

$$\mathbf{x}=\begin{bmatrix}\mathbf{x}_P\\ \mathbf{x}_C\end{bmatrix}=[x_1 \quad x_2 \quad \dots \quad x_{n+p}]^T \in R^{n+p},$$
$$x_i=\begin{cases} x_{P,i} & \text{if } i=\overline{1,n} \\ x_{C,i-n} & \text{otherwise}\end{cases}, \; i=\overline{1,n+p}. \tag{4}$$

Next, the state-space models of the controlled process and controller in the models (1) and (3) are merged using equation (2) and the structure presented in Fig. 1. Therefore, the state-space model of the control system becomes

$$\dot{\mathbf{x}}=\begin{bmatrix}\mathbf{f}_P\{\mathbf{x}_P,\boldsymbol{\alpha},g_C[\mathbf{x}_C,\beta,r-g_P(\mathbf{x}_P,\boldsymbol{\alpha},d)],d\}\\ \mathbf{f}_C[\mathbf{x}_C,\beta,r-g_P(\mathbf{x}_P,\boldsymbol{\alpha},d)]\end{bmatrix}$$
$$=\mathbf{f}(\mathbf{x},\boldsymbol{\alpha},\beta,r,d)=[f_1 \quad f_2 \quad \dots \quad f_{n+p}]^T\in R^{n+p}, \tag{5}$$
$$y=g_P(\mathbf{x}_P,\boldsymbol{\alpha},d),$$
$$\mathbf{x}(t_0)=\begin{bmatrix}\mathbf{x}_{P,0}\\ \mathbf{x}_{C,0}\end{bmatrix}.$$

Considering the parameter of the controlled process $\alpha_j, j=\overline{1,m}$, the state sensitivity functions $\lambda_i^{\alpha_j}, i=\overline{1,n+p}$, and the output sensitivity function σ^{α_j} are defined according to

$$\lambda_i^{\alpha_j}=\left[\frac{\partial x_i}{\partial \alpha_j}\right]_{\alpha_j,0}, \; \sigma^{\alpha_j}=\left[\frac{\partial y}{\partial \alpha_j}\right]_{\alpha_j,0}, \; i=\overline{1,n+p}, \; j=\overline{1,m}, \tag{6}$$

where the subscript 0 indicates the nominal value of the appropriate parameter.

Using equations (6) to calculate the partial derivatives in the model (5) we obtain sensitivity models of the control system with respect to $\alpha_j, j=\overline{1,m}$, for the constant reference input r_0 and disturbance input d_0:

$$\dot{\lambda}_i^{\alpha_j}=\sum_{k=1}^{n+p}\left[\frac{\partial f_i}{\partial x_k}\right]_{\alpha_j,0}\lambda_k^{\alpha_j}+\left[\frac{\partial f_i}{\partial \alpha_j}\right]_{\alpha_j,0},$$
$$\sigma^{\alpha_j}=\sum_{k=1}^{n}\left[\frac{\partial g_P}{\partial x_k}\right]_{\alpha_j,0}\lambda_k^{\alpha_j}+\left[\frac{\partial g_P}{\partial \alpha_j}\right]_{\alpha_j,0}, \tag{7}$$
$$\lambda_i^{\alpha_j}(t_0)=0, \; i=\overline{1,n+p}, \; j=\overline{1,m}.$$

The initial state variables are important in the analysis of the sensitivity models (7).

In order to obtain good dynamics of the control systems and reduced sensitivity we define the following objective functions with the design parameter β as an independent variable:

- the extended ISE:

$$I_{ISE}^{\alpha_j}(\beta)=\int_0^\infty \{e^2(t)+(\gamma^{\alpha_j})^2[\sigma^{\alpha_j}(t)]^2\}dt,\ \ j=\overline{1,m}, \tag{8}$$

- the extended IAE:

$$I_{IAE}^{\alpha_j}(\beta)=\int_0^\infty \{|\,e(t)\,|+(\gamma^{\alpha_j})^2[\sigma^{\alpha_j}(t)]^2\}dt,\ \ j=\overline{1,m}, \tag{9}$$

- the extended ITSE:

$$I_{ITSE}^{\alpha_j}(\beta)=\int_0^\infty \{te^2(t)+(\gamma^{\alpha_j})^2[\sigma^{\alpha_j}(t)]^2\}dt,\ \ j=\overline{1,m}, \tag{10}$$

- the extended ITAE:

$$I_{ITAE}^{\alpha_j}(\beta)=\int_0^\infty \{t\,|\,e(t)\,|+(\gamma^{\alpha_j})^2[\sigma^{\alpha_j}(t)]^2\}dt,\ \ j=\overline{1,m}, \tag{11}$$

where γ^{α_j}, $j=\overline{1,m},$ are the weighting parameters. The dynamic regimes with regard to step-type modifications of the reference input and disturbance inputs are considered; therefore, the number of objective functions in (8)–(11) is doubled.

Making use of the objective functions (8)–(11) the optimization problems which ensure the optimal design of the controllers are defined as follows:

$$\begin{aligned}&\beta^* = \arg\min_{\beta\in Do} I_{ISE}^{\alpha_j}(\beta),\ \ \beta^* = \arg\min_{\beta\in Do} I_{IAE}^{\alpha_j}(\beta),\\ &\beta^* = \arg\min_{\beta\in Do} I_{ITSE}^{\alpha_j}(\beta),\ \ \beta^* = \arg\min_{\beta\in Do} I_{ITAE}^{\alpha_j}(\beta),\ \ j=\overline{1,m},\end{aligned} \tag{12}$$

where β^* is the optimal value of the variable β, and *Do* is the feasible domain of the variable β.

When setting the domain *Do*, we should first take in consideration the stability of the control system. Other inequality-type constraints can be imposed for the optimization problems defined in equation (12). For example, they can concern the actuator saturation [52], the robust stability of the control system or the controller robustness [53].

3 Particle Swarm Optimization and Simulated Annealing Algorithms

Particle Swarm Optimization

Particle Swarm Optimization (PSO) was originally designed and introduced by Eberhart and Kennedy [22,23]. PSO is a search algorithm inspired by the social behavior of birds, bees or schools of fishes. PSO uses the swarm intelligence concept, modeled by particles (agents) with specific positions and velocities, which interact locally with their environment to create coherent global functional patterns.

Social concepts like evolution, comparison and imitations of other individuals are typically associated with intelligent agents that interact in order to adapt to the environment and develop optimal patterns of behavior. Mutual learning allows individuals to behave in a similar way and to acquire adaptive patterns of behavior. The swarm intelligence is based on the following principles [54]:

- *proximity*, i.e. the population should be able to carry out simple time and space calculations,
- *quality*, i.e. the population should be able to respond to quality factors in the environment,
- *diverse response*, i.e. the population should not commit its activity to excessively long narrow channels,
- *stability*, i.e. the population should not change its behavior every time the environment changes,
- *adaptability*, i.e. the population should be able to change its behavior when it is worth the computational price.

PSO algorithm is an evolutionary algorithm, which similarly with the genetic algorithms starts with a random generation of candidate solutions and then searches for the optimal solution. In the case of PSO algorithm, the individual particles are updated in parallel, a new solutions depends on the previous one and on the solutions corresponding to its neighboring particles. The same rules apply to all updates. The particles are moving in a *D*-dimensional search space search space $\Re^D$ with randomly chosen velocities and positions, knowing their best values so far and the positions in the search space $\Re^D$. The position and velocity of each particle in the search space are updated at each step of the iteration. The velocity of each particle is adjusted according to its own previous moving history as well as to that of the other particles [28].

A swarm particle can be represented by two *D*-dimensional vectors, $X_i = [x_{i1} \quad x_{i2} \quad \dots \quad x_{iD}]^T \in \Re^D$ standing for the particle position and the particle velocity $V_i = [v_{i1} \quad v_{i2} \quad \dots \quad v_{iD}]^T$. Let $P_{i,Best} = [p_{i1} \quad p_{i2} \quad \dots \quad p_{iD}]^T$ be the best position of a specific particle and $P_{g,Best} = [p_{g1} \quad p_{g2} \quad \dots \quad p_{gD}]^T$ be the best position of the swarm.

The particle velocity and position updating rules can be expressed in terms of the state-space form [55] as follows:

$$V_i^{k+1} = wV_i^k + c_1 r_1 (P_{g,Best} - X_i^k) + c_2 r_2 (P_{i,Best} - X_i^k), \tag{13}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1}, \tag{14}$$

where: r_1, r_2 are random variables with a uniform distribution between 0 and 1, $i,\ i = \overline{1, n}$, is the index of the current particle in the swarm, n is the number of particles in the swarm, $k,\ k = \overline{1, j_{\max}}$, is the index of the current iteration, $j_{\max}$ is the maximum number of iterations. The parameter w in equation (13) stands for the inertia weight, which shows the effect of the previous velocity vector on the new one. $V_{\max}$ is the upper limit placed on the velocity in all the dimensions preventing the particle from moving too rapidly in the search space. The constants c_1 and c_2 represent the weighting factors of the stochastic accelerations pulling the particles towards their final positions. Adopting too low values of these weights will allow particles to roam far from the target regions before being tugged back. On the other hand, too high values will result in abrupt movements towards, or overshooting, the target regions.

The individual particles within the swarm communicate and learn from each other, and based on this they move to improve their previous position relative to their neighbors. Different neighborhood topologies can emerge on the basis of the communication strategy of the particles within the swarm. A star-type topology is created in the majority of cases. In that topology, each particle can communicate with every other individual forming a fully connected social network, so that each particle could access the overall best position.

PSO algorithm consists of the following steps [22,23,54,55]:

1. Initialize the swarm placing particles at random positions inside the d-dimensional search space,
2. Evaluate the fitness of each particle using its current position,
3. Compare the performance of each individual to its best performance so far,
4. Compare the performance of each particle to the best global performance,
5. Change the velocity of each particle according to equation (13),
6. Move each particle to its new position according to equation (14),
7. Go to step 2, until the maximum number of iterations is reached.

The flowchart of PSO algorithm is presented in Fig. 2 (a).

Simulated Annealing

Simulated Annealing (SA) is a random-search technique based on an analogy with the well-known annealing process used in metallurgy, consisting in a heat treatment that alters the microstructure of metal causing changes in properties such as strength and hardness and ductility. The final properties of the metal are very much dependent on the heating and cooling process. If the temperature cools too quickly, the final product will be brittle. If the temperature cools down slowly, the resulting product will have the right hardness and ductility.

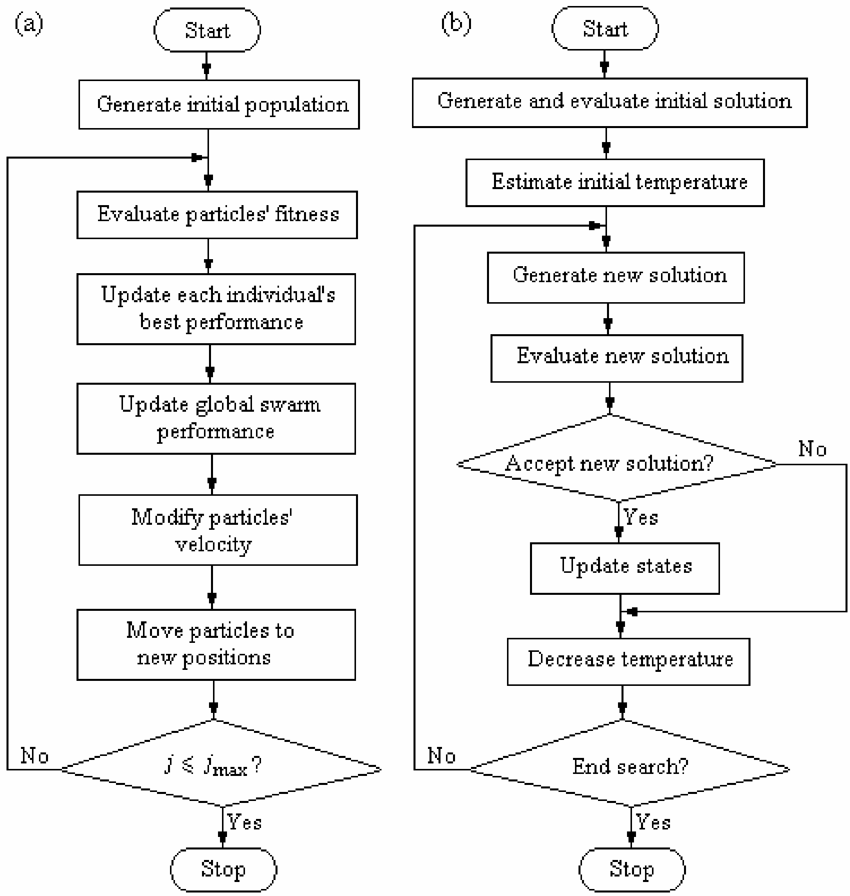

Fig. 2 Flowchart of PSO algorithm (a) and of SA algorithm (b).

SA algorithms were originally developed [24] to deal with highly nonlinear problems. It approaches the minimization problem similarly to the way a ball is rolling from valley to valley down a hill slope until it finally reaches the lowest possible (minimum altitude) location. If the ball does not have enough energy, it cannot roll high enough and it becomes trapped in a valley somewhere higher on the hill slope, above the lowest possible position [27]. The decision making on a particle staying in, or rolling off, a valley is based on a probabilistic energy framework.

SA algorithms start with a high temperature and an initial solution. Considering the initial vector solution $\boldsymbol{\varphi}$ with the corresponding fitness value $C(\boldsymbol{\varphi})$ of the fitness function C, the next probable vector solution $\boldsymbol{\psi}$ is chosen from the vicinity of $\boldsymbol{\varphi}$, and it will have the fitness value $C(\boldsymbol{\psi})$.

SA algorithms contain a probabilistic-based framework for solution acceptance. Defining

$$\Delta C_{\boldsymbol{\varphi}\boldsymbol{\psi}} = C(\boldsymbol{\varphi}) - C(\boldsymbol{\psi}), \tag{15}$$

the probability of $\boldsymbol{\varphi}$ being the next solution, referred to as p_{ψ}, is

$$p_{\psi} = \begin{cases} 1 & \text{if } \Delta C_{\varphi\psi} \le 0, \\ \exp(-\Delta C_{\varphi\psi} / \theta) & \text{otherwise.} \end{cases} \tag{16}$$

where θ is the current temperature value of the algorithm.

If $p_{\psi} > r_n$, where r_n is a randomly selected number, $0 \le r_n \le 1$, then $\boldsymbol{\psi}$ will be the new solution. Otherwise a new solution must be generated. As it can be observed in this framework, there is a valid probability of replacing the current solution with a higher cost solution.

The above process repeats for a predetermined number of steps, and the temperature is next reduced. The algorithms end when the temperature value is so low that it does not allow any modification of the fitness function, and the last value is the solution.

SA algorithms implemented here to solve the optimization problems defined in equation (12) can be formulated in terms of the following steps:

- *Step 1*. Set $m = 0$ and the minimum temperature $\theta_{\min}$. Choose the initial temperature θ_m.
- *Step 2*. Generate the initial solution $\boldsymbol{\varphi}$ and calculate its corresponding fitness value $C(\boldsymbol{\varphi})$.
- *Step 3*. Generate a probable solution $\boldsymbol{\psi}$ by perturbing $\boldsymbol{\varphi}$ and evaluate the fitness value $C(\boldsymbol{\psi})$.
- *Step 4*. Calculate $\Delta C_{\varphi\psi}$ making use of equation (15). If $\Delta C_{\varphi\psi} \le 0$ then $\boldsymbol{\psi}$ is the new solution. Else select randomly r_n, $0 \le r_n \le 1$, and calculate p_{ψ} by means of the function defined in equation (16). If $p_{\psi} > r_n$ then $\boldsymbol{\psi}$ is the new solution.
- *Step 5*. Reduce the temperature according to the temperature decrement rule

$$\theta_{m+1} = f_{cs}(\theta_m), \tag{17}$$

where f_{cs} is the cooling schedule.

- *Step 6*. If $\theta_m > \theta_{\min}$ then go to step 3, else stop.

The subscript *m* in SA algorithms stands for the iteration index.

The flowchart of an SA algorithm is illustrated in Fig. 2 (b).

The most common temperature decrement rule is the well-accepted exponential cooling schedule

$$\theta_{m+1} = \alpha_{cs}\theta_m , \tag{18}$$

where $\alpha_{cs} = \text{const}$, $\alpha_{cs} < 1$, $\alpha_{cs} \approx 1$.

The fitness functions C implemented in our SA algorithms to solve the optimization problems defined in equation (12) are the objective functions defined in equations (8)–(11). With this regard the vector arguments $\boldsymbol{\varphi}$ or $\boldsymbol{\psi}$ of these fitness functions are replaced by the scalar parameter β .

SA algorithms are very versatile as they do not depend on any restrictive properties of a model [24–27]. In addition they can be used in combination with other gradient-based algorithms due to their flexibility and ability to approach the global optimum.

4 Case Study

As shown in [50] the PI controllers can be tuned by the Extended Symmetrical Optimum (ESO) method to guarantee a good compromise to the desired / imposed control performance indices making use of a single design parameter referred to as β . This design parameter ensures the generalization of Kessler's Symmetrical Optimum (SO) method [56,57] to obtain performance enhancements.

This case study will show how PSO and SA algorithms combined with the ESO method can be used for an efficient tuning of a PI controller for a class of second-order processes with integral component [49]. The ESO method allows simplifying the implementation of PSO and SA algorithms.

The class of second-order processes with integral component considered here as case study is a particular controlled process (P) in the framework of the control system structure presented in Fig. 1. The accepted class of controlled processes can be modeled by the state-space model as that presented in equation (1) or by the transfer function $P(s)$

$$P(s) = Y(s)/U(s) = k_P /[s(1+ s\,T_{\Sigma})] , \tag{19}$$

where $Y(s)$ is the Laplace transform of y, $U(s)$ is the Laplace transform of u, zero initial conditions are assumed, s is the complex argument specific to the Laplace transform, k_P is the controlled process gain, and T_{Σ} is the small time constant or the sum of small time constants of the process. The sum of small time constants is used when the transfer function defined in equation (19) is a simplified model of higher order processes.

This process is used as a representative benchmark for many servo systems applications where equation (19) can be viewed as a convenient simplified linearized mathematical model. Thus the two parameters are variable and the sensitivity analysis and design of controllers with reduced sensitivity is of interest, and the parameter vector of the controlled process is ($m = 2$)

$$\boldsymbol{\alpha} = [\alpha_1 = k_P \quad \alpha_2 = T_\Sigma]^T \in R^2 . \tag{20}$$

The controlled process consists of the two blocks with the transfer functions presented in Fig. 3. The integral component of the process, with the transfer function $1/s$, is illustrated in equation (19) and in Fig. 3 because of the integration property of the Laplace transform.

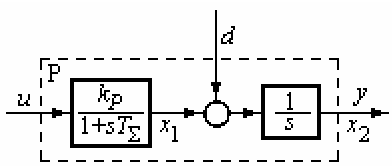

Fig. 3 Process structure.

The state-space model of this controlled process is the following particular expression of the state-space model (1):

$$\begin{aligned} &\dot{x}_1(t) = x_2(t) + d(t), \\ &\dot{x}_2(t) = -(1/T_\Sigma)x_2(t) + (k_P/T_\Sigma)u(t), \\ &y(t) = x_1(t), \end{aligned} \tag{21}$$

where x_1 and x_2 are the state variables shown in Fig. 3.

The performance specifications imposed to the control systems concern first the reference input r tracking and the regulation in the presence of disturbance inputs (d). These performance specifications are expressed in terms of maximum values of the control system performance indices; the control systems design should ensure the fulfillment of the constraints resulted from these maximum values, i.e. to ensure smaller performance indices with respect to the maximum imposed ones. It is difficult to meet generally all performance specifications.

Fig. 4 exemplifies the definitions of three performance indices with respect to the step modification of r: σ_1 – the overshoot, t_r – the 10 % to 90 % rise time, and t_s – the 2 % settling time. The following notations are used in Fig. 4: y_0 – the initial value of y, y_f – the final value of y, $y_{10} = 0.1|y_f - y_0|$, $y_{90} = 0.9|y_f - y_0|$, $y_{98} = 0.98|y_f - y_0|$, and $y_{102} = 1.02|y_f - y_0|$.

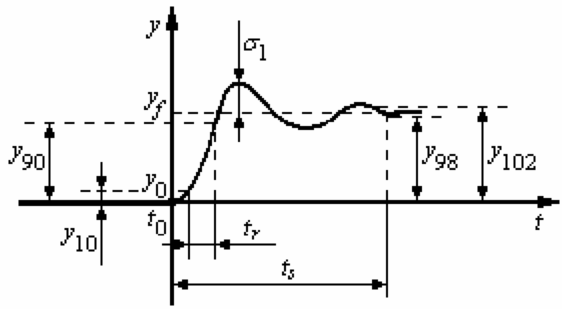

Fig. 4 Control system performance indices defined with respect to the step modification of *r*.

Good control system performance indices can be obtained if the process is controlled by means of the control system structure presented in Fig. 1, where C is a PI controller. The transfer function of the PI controller is *C*(s)

$$C(s) = U(s) / E(s) = k_c(1 + sT_i) / s = k_C[1 + 1/(sT_i)],\ k_C = T_i k_c, \tag{22}$$

where $U(s)$ is the Laplace transform of *u*, $E(s)$ is the Laplace transform of *e*, zero initial conditions are assumed, and the tuning parameters of the PI controller are k_C (k_c) – the gain, and T_i – the integral time constant.

The choice of the single design parameter β specific to the ESO method within the domain

$$Do = \{\beta \mid 1 < \beta < 20\}, \tag{23}$$

guarantees a compromise between the control system performance indices as shown in Fig. 5.

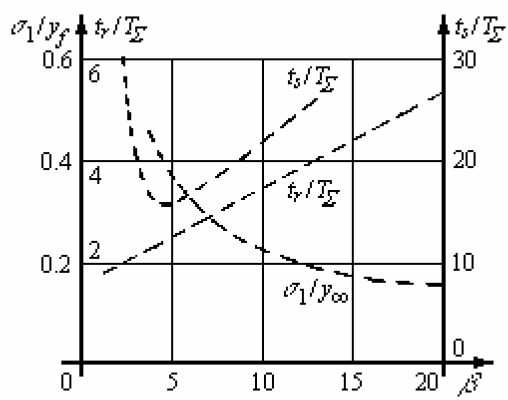

Fig. 5 Control system performance indices with respect to *r* versus β.

The ESO-based PI tuning conditions give the tuning parameters of the PI controller:

$$k_c = 1/(\beta\sqrt{\beta}\, k_P T_\Sigma^2),\ T_i = \beta T_\Sigma,\ k_C = 1/(\sqrt{\beta}\, k_P T_\Sigma), \quad (24)$$

where the parameter β is chosen such that so set the performance indices (Fig. 5) in order to fulfill the performance specifications, and $\beta = 4$ corresponds to Kessler's SO method.

The control system performance indices can be improved viz. alleviated by filtering the reference input in terms of introducing the reference filter F in the control system structure. This results in the two-degree-of-freedom control system structure presented in Fig. 6, where $\tilde{r}$ is the filtered reference input. A simple ESO-based reference filter is characterized by the transfer function $F(s)$

$$F(s) = \tilde{R}(s)/R(s) = 1/(1 + s\, T_i), \quad (25)$$

where $\tilde{R}(s)$ is the Laplace transform of $\tilde{r}$, $R(s)$ is the Laplace transform of *r*, and zero initial conditions are assumed.

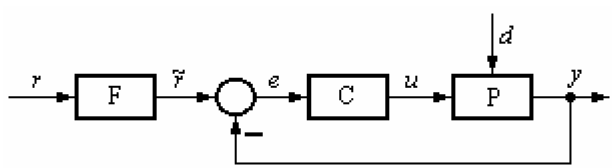

Fig. 6 Two-degree-of-freedom control system structure with reference filter.

The values of the design parameter β will be obtained as follows as solutions to the optimization problems (12).1

Accepting that x_3 is the output of the integral component in the parallel structure of the PI controller (Fig. 7), the state-space model of this controller is

$$\begin{aligned} \dot{x}_3(t) &= (1/T_i)e(t), \\ u(t) &= k_C(x_3(t) + e(t)). \end{aligned} \quad (26)$$

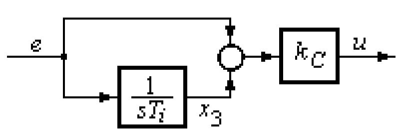

Fig. 7 Parallel structure of the PI controller.

In order to derive the sensitivity models with respect to the parametric variations of the controlled process, we have to tune the linear PI controller in terms of equations (24) for the nominal values of the process parameters k_{P0} and $T_{\Sigma 0}$. The state-space model of the PI controller becomes then

$$\begin{aligned}&\dot{x}_3(t) = [1/(\beta T_{\Sigma 0})]e(t),\\&u(t) = [1/(\sqrt{\beta}\, k_{P0} T_{\Sigma 0})](x_3(t) + e(t)).\end{aligned} \tag{27}$$

Merging the models (25) and (27), the state-space model of the control system is

$$\begin{aligned}&\dot{x}_1(t) = x_2(t) + d(t),\\&\dot{x}_2(t) = -[k_P/(\sqrt{\beta}\, k_{P0} T_{\Sigma 0} T_\Sigma)]x_1(t) - (1/T_\Sigma)x_2(t)\\&\qquad + [k_P/(\sqrt{\beta}\, k_{P0} T_{\Sigma 0} T_\Sigma)]x_3(t) + [k_P/(\sqrt{\beta}\, k_{P0} T_{\Sigma 0} T_\Sigma)]r(t),\\&\dot{x}_3(t) = -[1/(\beta T_{\Sigma 0})]x_1(t) + [1/(\beta T_{\Sigma 0})]r(t),\\&y(t) = x_1(t).\end{aligned} \tag{28}$$

Using formulas (7) in equation (28) we obtain the sensitivity model with respect to k_P

$$\begin{aligned}&\dot{\lambda}_1^{k_P}(t) = \lambda_2^{k_P}(t),\\&\dot{\lambda}_2^{k_P}(t) = -[1/(\sqrt{\beta}\, T_{\Sigma 0}^2)]\lambda_1^{k_P}(t) - (1/T_{\Sigma 0})\lambda_2^{k_P}(t)\\&\qquad + [1/(\sqrt{\beta}\, T_{\Sigma 0}^2)]\lambda_3^{k_P}(t) - [1/(\sqrt{\beta}\, k_{P0} T_{\Sigma 0}^2)]x_{10}(t)\\&\qquad + [1/(\sqrt{\beta}\, k_{P0} T_{\Sigma 0}^2)]x_{30}(t) + [1/(\sqrt{\beta}\, k_{P0} T_{\Sigma 0}^2)]r_0(t),\\&\dot{\lambda}_3^{k_P}(t) = -[1/(\beta T_{\Sigma 0})]\lambda_1^{k_P}(t),\\&\sigma^{k_P}(t) = \lambda_1^{k_P}(t),\end{aligned} \tag{29}$$

and the sensitivity model with respect to T_Σ

$$\begin{aligned}&\dot{\lambda}_1^{T_\Sigma}(t) = \lambda_2^{T_\Sigma}(t),\\&\dot{\lambda}_2^{T_\Sigma}(t) = -[1/(\sqrt{\beta}\, T_{\Sigma 0}^2)]\lambda_1^{T_\Sigma}(t) - (1/T_{\Sigma 0})\lambda_2^{T_\Sigma}(t) + [1/(\sqrt{\beta}\, T_{\Sigma 0}^2)]\lambda_3^{T_\Sigma}(t)\\&\qquad + [1/(\sqrt{\beta}\, T_{\Sigma 0}^3)]x_{10}(t) + (1/T_{\Sigma 0}^2)x_{20}(t)\\&\qquad - [1/(\sqrt{\beta}\, T_{\Sigma 0}^3)]x_{30}(t) - [1/(\sqrt{\beta}\, T_{\Sigma 0}^3)]r_0(t),\\&\dot{\lambda}_3^{T_\Sigma}(t) = -[1/(\beta T_{\Sigma 0})]\lambda_1^{T_\Sigma}(t),\\&\sigma^{T_\Sigma}(t) = \lambda_1^{T_\Sigma}(t).\end{aligned} \tag{30}$$

Eight optimization problems (12) for the given application, corresponding to the dynamic regimes characterized by the unit step modification of the reference input, are solved as follows by means of PSO and SA algorithms. The other eight-optimization problems out of the 16 possible optimization problems, corresponding to the dynamic regimes characterized by modifications of the disturbance input, are not analyzed here because of the following reasons:

- the controller designs are generally done with respect to the reference input,
- the sensitivity models (29) and (30) do not depend on the disturbance input and the disturbance input affects only the nominal behavior of the control system,
- the controllers have integral character that ensure the disturbance rejection.

However, the behavior of the optimized control systems with respect to unit step modifications of the disturbance input is analyzed in order to outline the disturbance rejection. The effects of the weighting parameters on the solutions are analyzed in all optimization problems.

The analysis of PSO algorithm involves the effects of the number of particles in the swarm, the weightings of the stochastic acceleration terms that pull each particle towards their end positions and the maximum number of iterations of the algorithm. Besides, the effects of linear cooling schedule, number of steps for each perturbation of the solution and the acceptance / rejection rates are analyzed for SA algorithm.

The PSO algorithm described in the previous section was implemented in Matlab in order to validate the proposed PI controller design method for the control of the process with the transfer function (19) and the parameters $k_P = 1$ and $T_\Sigma = 1\,\text{s}$.

For the sake of simplicity the fitness functions corresponding to each objective function (12) are represented by the generic names I^{k_P} and I^{T_Σ}. They were used to evaluate the population and calculate the local best $P_{i,Best}$ and global best $P_{g,Best}$. The fitness functions were calculated by repeated simulations of the control systems' behavior with respect to the unit step modification of the reference input accepting the presence of the reference filter described by formula (25).

The optimization problems derived from equations (12) are reduced to finding the optimal value β^* of the design parameter β, thus reducing the solution search space to $D = 1$. The values of the weighting parameters used in the minimization of I^{k_P} were chosen to belong to the set $(\gamma^{k_P})^2 \in \{0, 0.1, 1, 10\}$. The values of the parameters in the corresponding PSO algorithm were set to $n = 10, c_1 = c_2 = 1.2$. The analysis of the effect of the maximum number of iterations on the optimal values of the controller tuning parameters and minimum values of the objective functions is illustrated in Tables 1 to 4.

Table 1 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{ISE}^{k_P}$

$(\gamma^{k_P})^2$	$j_{\max}$	β^*	k_C^*	T_i^*	$I_{ISE}^{k_P}$
0	30	31.7219	0.1776	31.7219	0.2466
0	50	48.7477	0.1432	48.7477	0.2438
0	100	53.0471	0.1373	53.0471	0.2432
0	200	106.7760	0.0968	106.7760	0.2241
0	500	92.8131	0.1038	92.8131	0.2303
0.1	30	5.1401	0.4411	5.1401	0.5507
0.1	50	5.1388	0.4411	5.1388	0.5507
0.1	100	5.1388	0.4411	5.1388	0.5507
1	30	2.6509	0.6142	2.6509	2.0253
1	50	2.6511	0.6142	2.6511	2.0253
1	100	2.6512	0.6142	2.6512	2.0253
10	30	2.1286	0.6854	2.1286	15.0580
10	50	2.1322	0.6848	2.1322	15.0579
10	100	2.1322	0.6848	2.1322	15.0579

Table 2 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{IAE}^{k_P}$

$(\gamma^{k_P})^2$	$j_{\max}$	β^*	k_C^*	T_i^*	$I_{IAE}^{k_P}$
0	30	3.8982	0.5065	3.8982	2.1032
0	50	3.8981	0.5065	3.8981	2.1032
0	100	3.8981	0.5065	3.8981	2.1032
0	200	3.9027	0.5062	3.9027	2.1031
0	500	3.9027	0.5062	3.9027	2.1031
0.1	30	3.8231	0.5114	3.8231	2.2828
0.1	50	3.8252	0.5113	3.8252	2.2828
0.1	100	3.8252	0.5113	3.8252	2.2828
1	30	3.2637	0.5535	3.2637	3.8144
1	50	3.2309	0.5563	3.2309	3.8139
1	100	3.2272	0.5567	3.2272	3.8138
10	30	2.2880	0.6611	2.2880	17.2018
10	50	2.2861	0.6614	2.2861	17.2018
10	100	2.2861	0.6614	2.2861	17.2018

Table 3 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{ITSE}^{k_P}$

$(\gamma^{k_P})^2$	$j_{\max}$	β^*	k_C^*	T_i^*	$I_{ITSE}^{k_P}$
0	30	4.5597	0.4683	4.5597	1.8650
0	50	4.5596	0.4683	4.5596	1.8650
0	100	4.5595	0.4683	4.5595	1.8650
0	200	4.5595	0.4683	4.5595	1.8650
0.1	30	4.3409	0.4800	4.3409	2.0606
0.1	50	4.3409	0.4800	4.3409	2.0606
1	30	3.4282	0.5401	3.4282	3.6606
1	50	3.4281	0.5401	3.4281	3.6606
1	100	3.4281	0.5401	3.4281	3.6606
10	30	2.4135	0.6437	2.4135	17.3087
10	50	2.4135	0.6437	2.4135	17.3087

Table 4 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{ITAE}^{k_P}$

$(\gamma^{k_P})^2$	$j_{\max}$	β^*	k_C^*	T_i^*	$I_{ITAE}^{k_P}$
0	30	3.5658	0.5296	3.5658	13.0250
0	50	3.5658	0.5296	3.5658	13.0250
0.1	30	3.5630	0.5298	3.5630	13.1970
0.1	50	3.5658	0.5296	3.5658	13.1968
0.1	100	3.5658	0.5296	3.5658	13.1968
1	30	3.5732	0.5290	3.5732	14.7451
1	50	3.5158	0.5333	3.5158	14.7382
1	100	3.5158	0.5333	3.5158	14.7382
10	30	3.1316	0.5651	3.1316	29.6065
10	50	3.3519	0.5462	3.3519	29.7806
10	100	3.1148	0.5666	3.1148	29.6057

A similar analysis was done for the family objective functions I^{T_Σ} accepting the weighting parameter $(\gamma^{T_\Sigma})^2 \in \{0,0.1,1,10\}$ and the same parameters in PSO algorithm. The results are illustrated in Tables 5 to 8.

Table 5 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{ISE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	j_{max}	β^*	k_C^*	T_i^*	$I_{ISE}^{T_\Sigma}$
0	30	32.0494	0.1766	32.0494	0.2465
0	50	57.1700	0.1323	57.1700	0.2426
0	100	45.2117	0.1487	45.2117	0.2443
0	200	72.4913	0.1175	72.4913	0.2386
0	500	144.4070	0.0832	144.4070	0.2118
0.1	30	3.6023	0.5269	3.6023	0.6885
0.1	50	3.6036	0.5268	3.6036	0.6885
0.1	100	3.6036	0.5268	3.6036	0.6885
1	30	2.9529	0.5819	2.9529	2.9854
1	50	2.9531	0.5819	2.9531	2.9854
1	100	2.9530	0.5819	2.9530	2.9854
10	30	2.8696	0.5903	2.8696	25.5593
10	50	2.8689	0.5904	2.8689	25.5593
10	100	2.8689	0.5904	2.8689	25.5593

Table 6 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{IAE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	j_{max}	β^*	k_C^*	T_i^*	$I_{IAE}^{T_\Sigma}$
0	30	3.9031	0.5062	3.9031	2.1031
0	50	3.9027	0.5062	3.9027	2.1031
0	100	3.9027	0.5062	3.9027	2.1031
0.1	30	3.6591	0.5228	3.6591	2.3995
0.1	50	3.6607	0.5227	3.6607	2.3995
0.1	100	3.6607	0.5227	3.6607	2.3995
1	30	3.0690	0.5708	3.0690	4.7623
1	50	3.0887	0.5690	3.0887	4.7619
1	100	3.0887	0.5690	3.0887	4.7619
10	30	2.8860	0.5886	2.8860	27.3707
10	50	2.8860	0.5886	2.8860	27.3707

Table 7 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{ITSE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	j_{max}	β^*	k_C^*	T_i^*	$I_{ITSE}^{T_\Sigma}$
0	30	4.5602	0.4683	4.5602	1.8650
0	50	4.5595	0.4683	4.5595	1.8650
0.1	30	4.0134	0.4992	4.0134	2.2035
0.1	50	3.9491	0.5032	3.9491	2.2029
0.1	100	3.9480	0.5033	3.9480	2.2029
1	30	3.1716	0.5615	3.1716	4.6304
1	50	3.1703	0.5616	3.1703	4.6304
1	100	3.1704	0.5616	3.1704	4.6304
10	30	2.8994	0.5873	2.8994	27.2770
10	50	2.8994	0.5873	2.8994	27.2770

Table 8 Results of the analysis of the effects of the maximum number of iterations on the optimal controller parameters and minimum value of objective function in case of $I_{ITAE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	j_{max}	β^*	k_C^*	T_i^*	$I_{ITAE}^{T_\Sigma}$
0	30	3.6162	0.5259	3.6162	13.0291
0	50	3.5658	0.5296	3.5658	13.0250
0	100	3.5658	0.5296	3.5658	13.0250
0.1	30	3.5631	0.5298	3.5631	13.3043
0.1	50	3.5641	0.5297	3.5641	13.3043
0.1	100	3.5641	0.5297	3.5641	13.3043
1	30	3.4401	0.5392	3.4401	15.7773
1	50	3.4450	0.5388	3.4450	15.7773
1	100	3.4389	0.5392	3.4389	15.7772
10	30	3.0613	0.5715	3.0613	39.0402
10	50	3.0620	0.5715	3.0620	39.0401
10	100	3.0620	0.5715	3.0620	39.0401

Similar results were obtained for different combinations of the parameters regarding PSO algorithm for example $n \in \{10, 11, ..., 30\}$, $0.3 \le c_1 < 1.2$, and $0.3 \le c_2 < 1.2$. Therefore, the analysis of the effects of the parameters n, c_1, c_2 proves that starting with a small value of j_{max} the same value of the optimal solution is obtained no matter the values of n, c_1, c_2 taken into consideration.

Numerous experiments done here with the implemented PSO algorithms prove that their sensitivity with respect to the initial conditions associated to equations (13) and (14) is relatively small. The considered case study assisted by PSO algorithms shows that the results exhibit reduced sensitivity with respect to the initial conditions of the sensitivity models.

The same generic families of fitness functions I^{k_P} and I^{T_Σ} highlighting the objective functions (12) were used for SA algorithm implemented in Matlab as well.

In order to get a higher rate of convergence we used the following linear cooling schedule

$$T^{K+1} = 0.9T^{K}\,, \tag{31}$$

where T is the temperature and K is the current iteration index.

A more abrupt cooling schedule could generate similar results, but at the expense of a higher probability of being trapped into a local minimum. A maximum number of 300 iterations at each temperature level, with a success rate of 50 and a rejection rate of 1000, allowed obtaining a maximum convergence rate and a low probability to be trapped in a local minimum.

An identical set of weighting parameters, $(\gamma^{k_P})^2 \in \{0,0.1,1,10\}$ and $(\gamma^{T_\Sigma})^2 \in \{0,0.1,1,10\}$, was used for both families of objective functions I^{k_P} and I^{T_Σ}.

The optimal controller parameters and minimum objective functions for the family of objective functions I^{k_P} are presented in Tables 9 to 12.

Table 9 Optimal controller parameters that minimize $I^{k_P}_{ISE}$

$(\gamma^{k_P})^2$	β^*	k_C^*	T_i^*	$I^{k_P}_{ISE}$
0	36.8668	0.1647	36.8668	0.2455
0.1	5.1357	0.4413	5.1357	0.5507
1	2.6519	0.6141	2.6519	2.0253
10	2.1319	0.6849	2.1319	15.0579

Table 10 Optimal controller parameters that minimize $I^{k_P}_{IAE}$

$(\gamma^{k_P})^2$	β^*	k_C^*	T_i^*	$I^{k_P}_{IAE}$
0	3.9030	0.5062	3.9030	2.1031
0.1	3.8254	0.5113	3.8254	2.2828
1	3.2271	0.5567	3.2271	3.8138
10	2.2862	0.6614	2.2862	17.2018

Table 11 Optimal controller parameters that minimize $I^{k_P}_{ITSE}$

$(\gamma^{k_P})^2$	β^*	k_C^*	T_i^*	$I^{k_P}_{ITSE}$
0	4.5606	0.4683	4.5606	1.8650
0.1	4.3406	0.4800	4.3406	2.0606
1	3.4273	0.5402	3.4273	3.6606
10	2.4134	0.6437	2.4134	17.3087

Table 12 Optimal controller parameters that minimize $I_{ITAE}^{k_P}$

$(\gamma^{k_P})^2$	β^*	k_C^*	T_i^*	$I_{ITAE}^{k_P}$
0	3.5727	0.5291	3.5727	13.0251
0.1	3.5658	0.5296	3.5658	13.1968
1	3.5024	0.5343	3.5024	14.7393
10	3.1148	0.5666	3.1148	29.6057

A similar analysis was performed for the family of objective functions I^{T_Σ} leading to the results shown in Tables 13 to 16.

Table 13 Optimal controller parameters that minimize $I_{ISE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	β^*	k_C^*	T_i^*	$I_{ISE}^{T_\Sigma}$
0	37.0825	0.1642	37.0825	0.2454
0.1	3.6026	0.5269	3.6026	0.6885
1	2.9536	0.5819	2.9536	2.9854
10	2.8687	0.5904	2.8687	25.5593

Table 14 Optimal controller parameters that minimize $I_{IAE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	β^*	k_C^*	T_i^*	$I_{IAE}^{T_\Sigma}$
0	3.9027	0.5062	3.9027	2.1031
0.1	3.6620	0.5226	3.6620	2.3995
1	3.0887	0.5690	3.0887	4.7619
10	2.8864	0.5886	2.8864	27.3707

Table 15 Optimal controller parameters that minimize $I_{ITSE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	β^*	k_C^*	T_i^*	$I_{ITSE}^{T_\Sigma}$
0	4.5607	0.4683	4.5607	1.8650
0.1	3.9478	0.5033	3.9478	2.2029
1	3.1709	0.5616	3.1709	4.6304
10	2.8995	0.5873	2.8995	27.2770

Table 16 Optimal controller parameters that minimize $I_{ITAE}^{T_\Sigma}$

$(\gamma^{T_\Sigma})^2$	β^*	k_C^*	T_i^*	$I_{ITAE}^{T_\Sigma}$
0	3.5531	0.5305	3.5531	13.0262
0.1	3.5440	0.5312	3.5440	13.3051
1	3.4389	0.5392	3.4389	15.7772
10	3.0619	0.5715	3.0619	39.0401

This case study shows the reduced effects of the initial conditions associated to the sensitivity models (29) and (30) of SA algorithm on the optimal values of the controller tuning parameters.

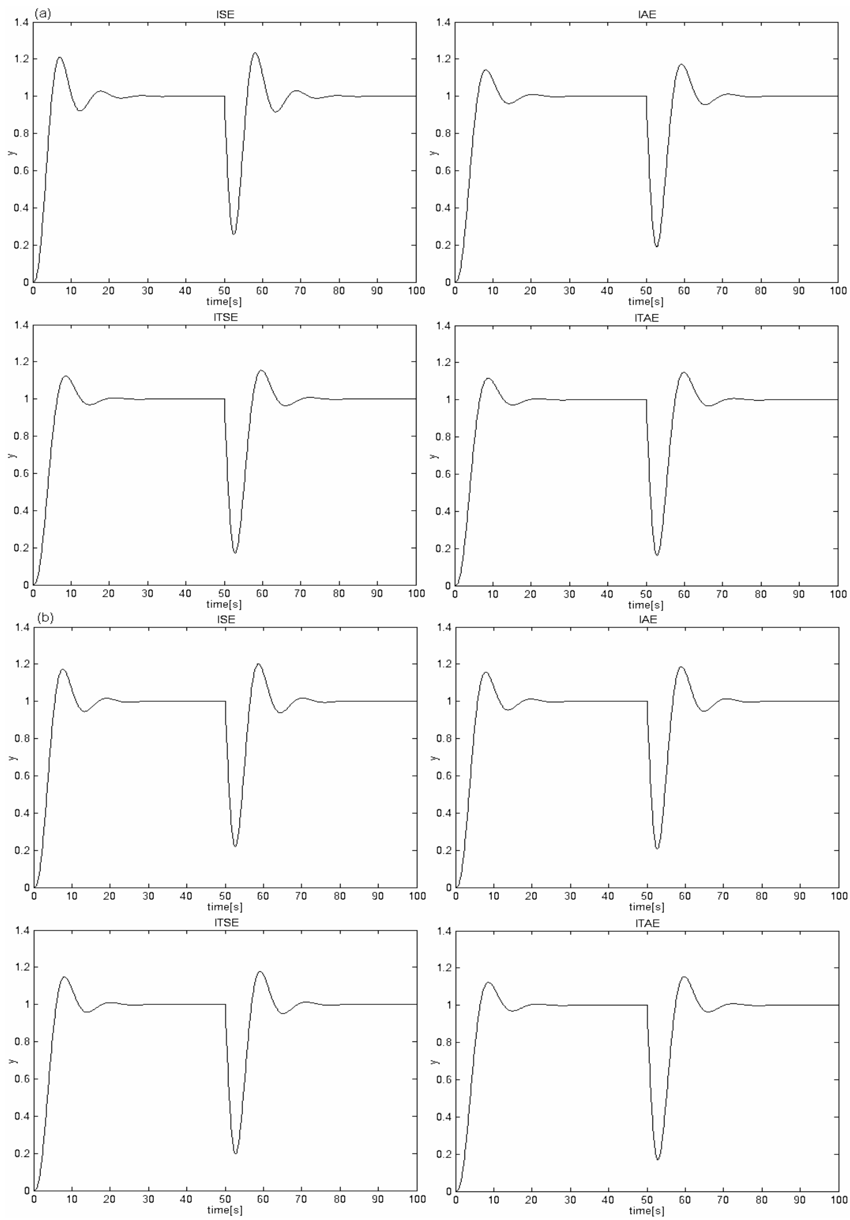

Fig. 8 Simulation results of optimal control systems obtained by PSO algorithm for the sensitivity model (29) (a) and sensitivity model (30) (b).

The controlled output (y) versus time diagrams given in Fig. 8 (a) and Fig. 8 (b) illustrate the optimization of the control system behavior by the minimization of the families of objective functions I^{k_P} and I^{T_Σ}, for $(\gamma^{k_P})^2 = 1$ and $(\gamma^{T_\Sigma})^2 = 1$, respectively, for the parameters of the optimal PI controller obtained by PSO algorithm with $j_{\max} = 100$. These simulations employed a unit step reference input, followed by a –0.5 step disturbance input applied at 50 s.

Considering the same simulation scenario, Fig. 9 (a) and Fig. 9 (b) illustrate the behaviors of the control systems optimized by the minimization of the objective functions I^{k_P} and I^{T_Σ}, for $(\gamma^{k_P})^2 = 1$ and $(\gamma^{T_\Sigma})^2 = 1$, respectively, and the parameters of the optimal PI controller obtained by SA algorithm.

The simulation results presented in Figs. 8 and 9 convincingly show that the controller exhibits good setting time and overshoot performance indices. The rapid decay of the system responses in Figs. 8 and 9 is explained by the application of the –0.5 step disturbance input (*d* in Fig. 1 and in Fig. 3) at 50 s.

Tables 1 to 16 show close values of the optimal parameters of the controllers for all the eight optimization problems solved. Furthermore, PSO and SA algorithms applied in solving the same optimization problems lead to very close solutions to those problems.

Figs. 10 and 11 illustrate the evolution of the design parameter β and the corresponding values of the objective function $I_{ISE}^{k_P}(\beta)$ versus the iterations of PSO and SA algorithm, respectively. The weighting parameter values used in these cases were $(\gamma^{k_P})^2 = 1$ and $(\gamma^{T_\Sigma})^2 = 1$ for both objective functions, and $j_{\max} = 100$ was considered in PSO algorithm.

Using the same weighting parameters and maximum number of iterations in PSO algorithm, Figs. 12 and 13 illustrate the evolution of the design parameter β and the corresponding values of another objective function, $I_{ISE}^{T_\Sigma}(\beta)$, versus the iterations of PSO and SA algorithm, respectively.

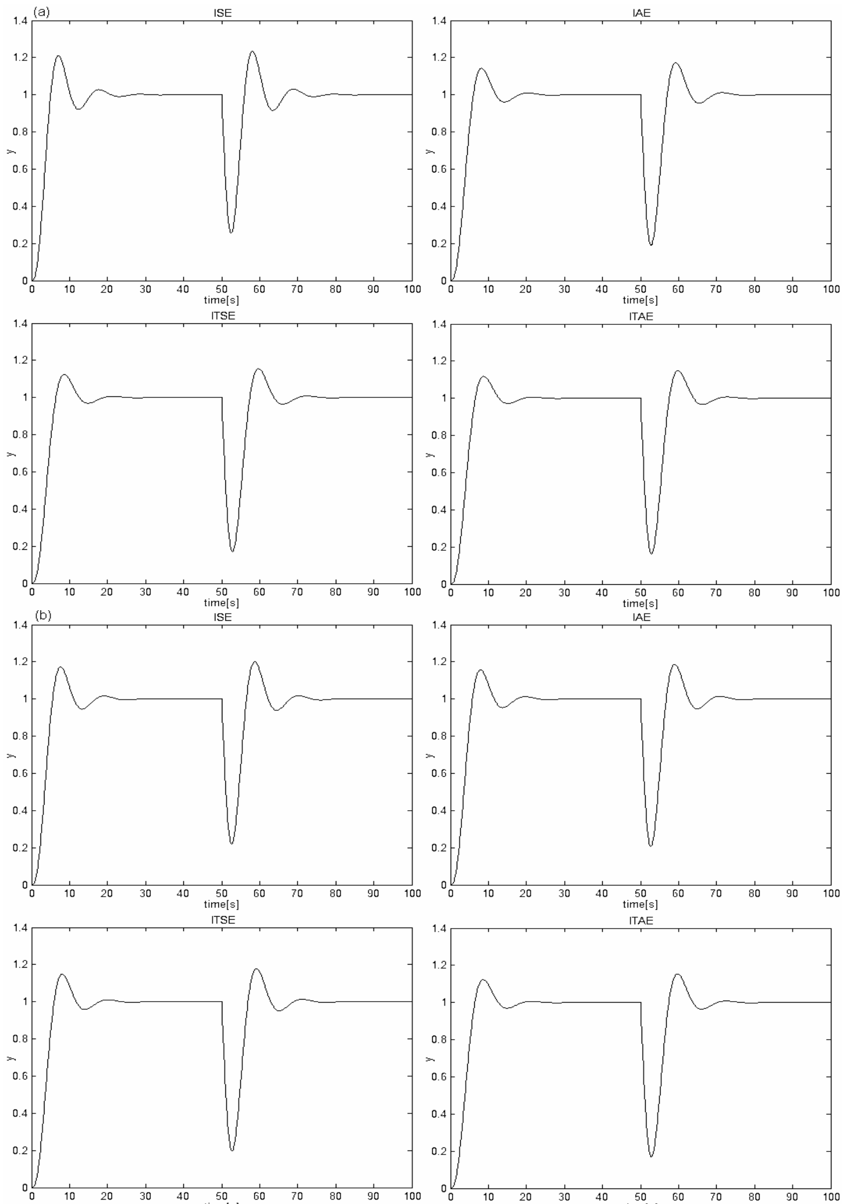

Fig. 9 Simulation results of optimal control systems obtained by SA algorithm for the sensitivity model (29) (a) and sensitivity model (30) (b).

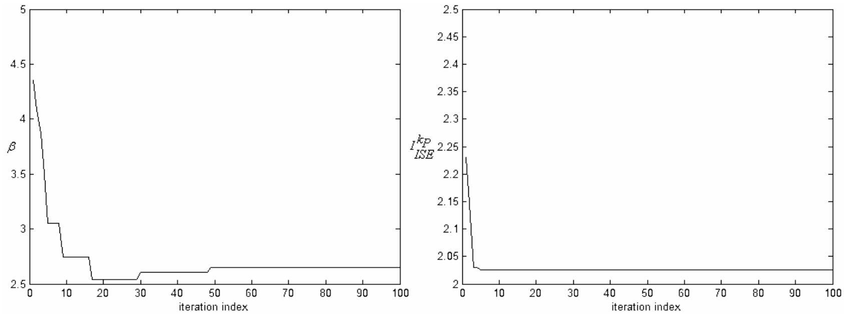

Fig. 10 Evolution of design parameter β and objective function $I_{ISE}^{k_P}$ versus the iteration index in PSO algorithm.

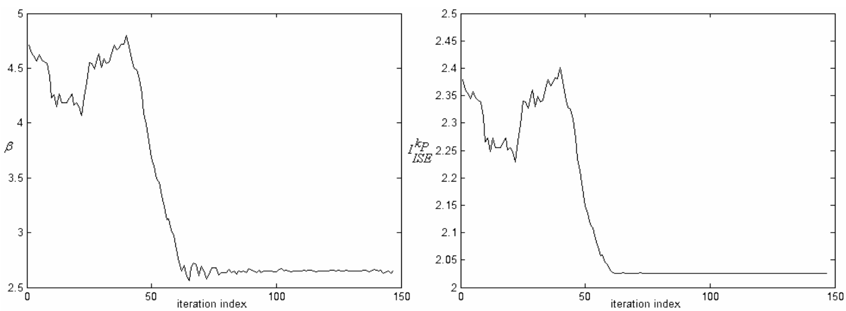

Fig. 11 Evolution of design parameter β and objective function $I_{ISE}^{k_P}$ versus the iteration index in SA algorithm.

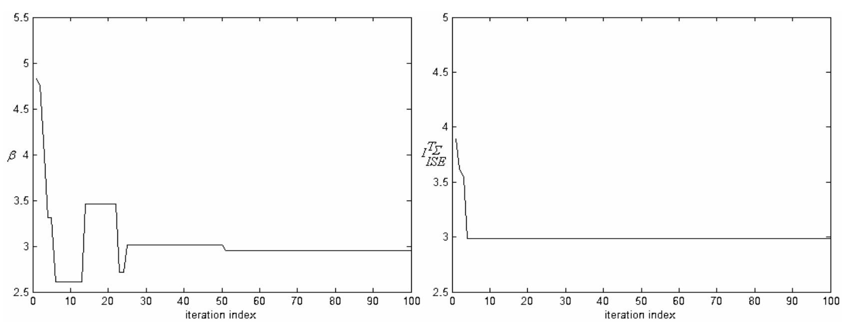

Fig. 12 Evolution of design parameter β and objective function $I_{ISE}^{T_\Sigma}$ versus the iteration index in PSO algorithm.

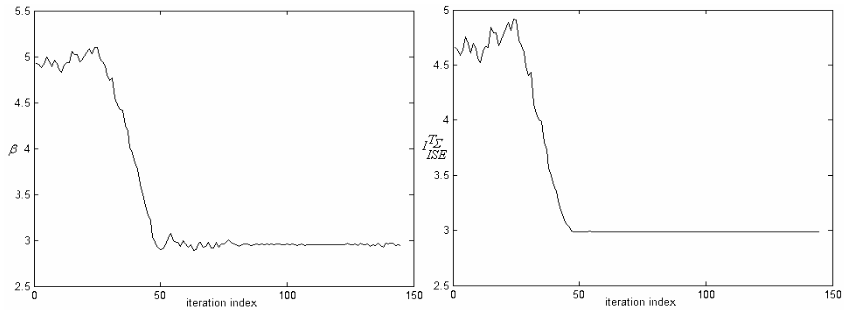

Fig. 13 Evolution of design parameter β and objective function $I_{ISE}^{T_\Sigma}$ versus the iteration index in SA algorithm.

The close values produced by the two PSO and SA optimization algorithms in similar situations demonstrate that the global minimum has been reached and thus the correct solutions to the optimization problems defined in equations (12) were obtained. Both algorithms have also avoided being trapped in local minima.

5 Conclusions

In this chapter, an application of PSO and SA optimization algorithms to the optimal design of PI controllers with a reduced sensitivity with respect to the parametric variations of the controlled process was presented. The proposed optimization problems ensure the weighted minimization of several integrals of the control error eventually weighted by time, and output sensitivity functions.

The advantages of using these algorithms include an expedient implementation, good computational efficiency and convergence.

Simulations were conducted to illustrate the good behavior of the optimal control systems in the dynamic regimes characterized by the step modifications of reference and disturbance inputs. Both optimization algorithms have a reduced sensitivity to the initial conditions of the sensitivity models.

The main limitation for these algorithms regards the degrees of freedom represented by some of their parameters. SA algorithm also presents a more computational-intensive disadvantage compared to PSO algorithm.

Future research will study the applicability of these optimization algorithms to other sensitivity-based optimization problems in the frequency domain, the extension of their area of application to other processes and controller structures [58–63]. All applications should be accompanied by the systematic analysis and guarantee of the convergence of the algorithms such that to enable the implementation of low cost and hybrid algorithms.

Acknowledgments. This work was supported by the CNCSIS and CNMP of Romania. The support from the cooperation between the "Politehnica" University of Timisoara, Romania, the Óbuda University, Budapest, Hungary, and the University of Ljubljana, Slovenia, in the framework of the Hungarian-Romanian and Slovenian-Romanian Intergovernmental Science & Technology Cooperation Programs is dully acknowledged.

References

[1] Rosenwasser, E., Yusupov, R.: Sensitivity of automatic control systems. CRC Press, Boca Raton (2000)

[2] Yaniv, O.: Design of low-order controllers satisfying sensitivity constraints for unstructured uncertain plants. Int. J. Robust Nonlinear Control 14, 1359–1370 (2004)

[3] Oded, T., Arkady, L.: Design of low order controllers satisfying sensitivity constraints for unstructured uncertain plants. In: Proceedings of 23rd IEEE Convention of Electrical and Electronics Engineers in Israel, Herzlia, Israel, pp. 33–36 (2004)

[4] Shirao, J., Imai, J., Konishi, M.: Structure design with sensitivity control performance limitation for electromechanical systems. In: Proceedings of SICE-ICASE International Joint Conference SICE-ICCAS 2006, Busan, Korea, pp. 2362–2367 (2006)

[5] Precup, R.E., Preitl, S., Korondi, P.: Fuzzy controllers with maximum sensitivity for servosystems. IEEE Trans. Ind. Electron 54, 1298–1310 (2007)

[6] Marchetti, G., Barolo, M., Jovanovic, L., Zisser, H., Seborg, D.E.: An improved PID switching control strategy for type 1 diabetes. IEEE Trans. Biomed. Eng. 55, 857–865 (2008)

[7] Wang, Y.G., Xu, X.M.: PID tuning for unstable processes with sensitivity specification. In: Proceedings of Chinese Control and Decision Conference CCDC 2009, Guilin, China, pp. 3460–3464 (2009)

[8] Precup, R.E., Preitl, S.: Optimisation criteria in development of fuzzy controllers with dynamics. Eng. Appl. Artif. Intell. 17, 661–674 (2004)

[9] Köppen, M.: Light-weight evolutionary computation for complex image-processing applications. In: Proceedings of 6th International Conference on Hybrid Intelligent Systems HIS 2006, Auckland, New Zealand, pp. 3–3 (2006)

[10] Zhou, H., Schaefer, G., Shi, C.: A mean shift based fuzzy c-means algorithm for image segmentation. In: Proceedings of 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC 2008, Vancouver, BC, Canada, pp. 3091–3094 (2008)

[11] Schaefer, G., Nakashima, T., Zavisek, M.: Analysis of breast thermograms based on statistical image features and hybrid fuzzy classification. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 753–762. Springer, Heidelberg (2008)

[12] Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72, 2729–2730 (2009)

[13] Precup, R.E., Preitl, S.: On the stability and sensitivity analysis of fuzzy control systems for servo-systems. In: Nedjah, N., de Macedo Mourelle, L. (eds.) Fuzzy Systems Engineering, Theory and Practice, pp. 131–161. Springer, Heidelberg (2005)

[14] Ekel, P.Y., Menezes, M., Schuffner Neto, F.H.: Decision making in a fuzzy environment and its application to multicriteria power engineering problems. Nonlinear Analysis: Hybrid Syst. 1, 527–536 (2007)

[15] Xu, J., Wu, H., Wang, Y.: Unpower aerocraft augmented state feedback tracking guaranteed cost control. J. Syst. Eng. Electron 19, 125–130 (2008)

[16] Fmmo, C., Calado, JMF.: Approaches to human arm movement control - A review. Annu Rev Control 33, 69–77 (2009)
[17] Chen, J., Kong, C.K.: Performance assessment for iterative learning control of batch units. J. Process Control 19, 1043–1053 (2009)
[18] Kim, D.H., Abraham, A., Cho, J.H.: A hybrid genetic algorithm and bacterial foraging approach for global optimization. Inf. Sci. 177, 3918–3937 (2007)
[19] Nolle, L.: On a novel ACO-estimator and its application to the target motion analysis problem. Knowl-Based Syst. 21, 225–231 (2008)
[20] Köppen, M., Kinoshita, Y., Yoshida, K.: Auxiliary objectives for the evolutionary multi-objective principal color extraction from logo images. In: Proceedings of IEEE Congress on Evolutionary Computation CEC 2008, Hong Kong, China, pp. 3537–3544 (2008)
[21] Plant, W.R., Schaefer, G., Nakashima, T.: An overview of genetic algorithms in simulation soccer. In: Proceedings of IEEE Congress on Evolutionary Computation CEC 2008, Hong Kong, China, pp. 3897–3904 (2008)
[22] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks ICNN 1995, Perth, Australia, pp. 1942–1948 (1995)
[23] Kennedy, J., Eberhart, R.C.: A new optimizer using particle swarm theory. In: Proceedings of 6th International Symposium on Micro Machine and Human Science MHS 1995, Nagoya, Japan, pp. 39–43 (1995)
[24] Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 20, 671–680 (1983)
[25] Geman, S., Geman, D.: Stochastic relaxation, Gibbs distribution and the Bayesian restoration in images. IEEE Trans. Pattern Anal. Mach. Intell. 6, 721–741 (1984)
[26] Kalai, A.T., Vempala, S.: Simulated annealing for convex optimization. Math. Oper. Res. 31, 253–266 (2006)
[27] Ledesma, S., Torres, M., Hernández, D., Aviña, G., García, G.: Temperature cycling on simulated annealing for neural network learning. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 161–171. Springer, Heidelberg (2007)
[28] David, R.C., Rădac, M.B., Preitl, S., Tar, J.K.: Particle swarm optimization-based design of control systems with reduced sensitivity. In: Proceedings of 5th International Symposium on Applied Computational Intelligence and Informatics SACI 2009, Timisoara, Romania, pp. 491–496 (2009)
[29] Kim, T.H., Maruta, I., Sugie, T.: Robust PID controller tuning based on the constrained particle swarm optimization. Automatica 44, 1104–1110 (2008)
[30] Lin, C.M., Lin, M.H., Chen, C.H., Yeung, D.S.: Robust PID control system design for chaotic systems using particle swarm optimization algorithm. In: Proceedings of 2009 International Conference on Machine Learning and Cybernetics ICMLC, Baoding, China, vol. 6, pp. 3294–3299 (2009)
[31] Fang, H., Chen, L.: Application of an enhanced PSO algorithm to optimal tuning of PID gains. In: Proceedings of Chinese Control and Decision Conference CCDC 2009, Guilin, China, pp. 35–39 (2009)
[32] Su, C., Wu, Y.: Adaptive neural network predictive control based on PSO algorithm. In: Proceedings of Chinese Control and Decision Conference CCDC 2009, Guilin, China, pp. 5829–5833 (2009)
[33] Mukherjee, V., Ghoshal, S.P.: Intelligent particle swarm optimized fuzzy PID controller for AVR system. Electr. Power Syst. Res. 77, 1689–1698 (2007)

[34] Mohan Rao, A.R., Sivasubramanian, K.: Multi-objective optimal design of fuzzy logic controller using a self configurable swarm intelligence algorithm. Comput. Struct. 86, 2141–2154 (2008)
[35] Tang, H.Y., Ding, B., Qi, W.G.: Research on traffic mode of elevator applied fuzzy C-mean clustering algorithm based on PSO. In: Proceedings of International Conference on Measuring Technology and Mechatronics Automation ICMTMA 2009, Zhangjiajie, China, vol. 2, pp. 582–585 (2009)
[36] Veenhuis, C., Köppen, M., Vicente-Garcia, R.: Evolutionary multi-objective optimization of particle swarm optimizers. In: Proceedings of IEEE Congress on Evolutionary Computation CEC 2007, Singapore, pp. 2273–2280 (2007)
[37] Das, S., Abraham, A., Konar, A.: Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. Pattern Recognit. Lett. 29, 688–699 (2008)
[38] Shayeghi, H., Shayanfar, H.A., Jalili, A.: Load frequency control strategies: a state-of-the-art survey for the researcher. Energy Convers Manag. 50, 344–353 (2009)
[39] Sabat, S.L., dos Santos Coelho, L., Abraham, A.: MESFET DC model parameter extraction using quantum particle swarm optimization. Microelectron Reliab. 49, 660–666 (2009)
[40] Liang, X.R., Fan, Y.K., Jiang, T.: Application of PSO algorithm to coordinated ramp control. In: Proceedings of 2009 International Conference on Machine Learning and Cybernetics ICMLC, Baoding, China, vol. 3, pp. 1712–1716 (2009)
[41] Zhang, Y., Hu, Y.: On PID controllers based on simulated annealing algorithm. In: Proceedings of 27th Chinese Control Conference CCC 2008, Kunming, China, pp. 225–228 (2008)
[42] Cao, X.R.: Stochastic learning and optimization - A sensitivity-based approach. Annu. Rev. Control 33, 11–24 (2009)
[43] Qiu, X.Z., Xu, Z.G., Zhang, L.M., Zhou, J.X., Si, F.Q.: Nonlinear predictive control on the load system of a thermal power unit based on AOSVR and SAPSO. In: Proceedings of Asia-Pacific Power and Energy Engineering Conference APPEEC 2009, Wuhan, China, p. 4 (2009)
[44] Wu, M., Xu, C.H., She, J.H., Yokoyama, R.: Intelligent integrated optimization and control system for lead-zinc sintering process. Control Eng. Pract. 17, 280–290 (2009)
[45] Haber, R.E., Haber-Haber, R., Jiménez, A., Galán, R.: An optimal fuzzy control system in a network environment based on simulated annealing. An application to a drilling process, Appl. Soft Comput. 9, 889–895 (2009)
[46] Buyamin, S., Finch, J.W.: Comparative study on optimising the EKF for speed estimation of an induction motor using simulated annealing and genetic algorithm. In: Proceedings of IEEE International Electric Machines & Drives Conference IEMDC 2007, Antalya, Turkey, vol. 2, pp. 1689–1694 (2007)
[47] Sayol, J., Nolle, L., Schaefer, G., Nakashima, T.: Comparison of simulated annealing and SASS for parameter estimation of biochemical networks. In: Proceedings of IEEE Congress on Evolutionary Computation CEC 2008, Hong Kong, China, pp. 3568–3571 (2008)
[48] Qin, X.S., Huang, G.H., He, L.: Simulation and optimization technologies for petroleum waste management and remediation process control. J. Environ. Manag. 90, 54–76 (2009)
[49] Åström, K.J., Hägglund, T.: PID controllers theory: design and tuning. Instrument Society of America, Research Triangle Park, NC (1995)

[50] Preitl, S., Precup, R.E.: An extension of tuning relations after symmetrical optimum method for PI and PID controllers. Automatica 35, 1731–1736 (1999)
[51] Abonyi, J.: Fuzzy model identification for control. Birkhäuser, Boston (2003)
[52] Itagaki, N., Nishimura, H., Takagi, K.: Two-degree-of-freedom control system design in consideration of actuator saturation. IEEE/ASME Trans. Mechatronics 13, 470–475 (2008)
[53] Precup, R.E., Preitl, S.: PI and PID controllers tuning for integral-type servo systems to ensure robust stability and controller robustness. Electrical Eng (Archiv für Elektrotechnik) 88, 149–156 (2006)
[54] del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C., Harley, R.G.: Particle swarm optimization: Basic concepts, variants and applications in power systems. IEEE Trans. Evol. Comput. 12, 171–195 (2008)
[55] Khanesar, M.A., Tavakoli, H., Teshnehlab, M., Shoorehdeli, M.A.: A novel binary particle swarm optimization. In: Proceedings of Mediterranean Conference on Control & Automation MED 2007, Athens, Greece, p. 6 (2007)
[56] Kessler, C.: Das symmetrische Optimum. Teil I. Regelungstechnik 6, 395–400 (1955)
[57] Kessler, C.: Das symmetrische Optimum. Teil II. Regelungstechnik 6, 432–436 (1955)
[58] Horváth, L., Rudas, I.J.: Modeling and problem solving methods for engineers. Academic Press, Elsevier, Burlington, MA (2004)
[59] Johanyák, Z.C., Kovács, S.: Sparse fuzzy system generation by rule base extension. In: Proceedings of 11th International Conference on Intelligent Engineering Systems INES 2007, Budapest, Hungary, pp. 99–104 (2007)
[60] Vaščák, J.: Fuzzy cognitive maps in path planning. Acta Tech. Jaurinensis Ser. Intell. Comput. 1, 467–479 (2008)
[61] Klančar, G., Matko, D., Blažič, S.: Wheeled mobile robots control in a linear platoon. J. Intell. Robotic Syst. 54, 709–731 (2009)
[62] Chiou, J.S., Liu, M.T.: Numerical simulation for fuzzy-PID controllers and helping EP reproduction with PSO hybrid algorithm. Simul Modell Pract. Theory 17, 1555–1565 (2009)
[63] Liu, K., Tan, Y., He, X.: An adaptive staged PSO based on particles' search capabilities. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010. LNCS, vol. 6145, pp. 52–59. Springer, Heidelberg (2010)