Expert Systems with Applications 38 (2011) 13694–13701

Contents lists available at ScienceDirect

Expert Systems with Applications

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

A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map

Leandro dos Santos Coelho *, Marcelo Wicthoff Pessôa

Industrial and Systems Engineering Graduate Program, PPGEPS, Pontifical Catholic University of Paraná, Imaculada Conceição, 1155, Zip code 80215-901, Curitiba, Paraná, Brazil

ARTICLE INFO

Keywords:
Differential evolution
Chaotic sequences
PID tuning
Optimization
Evolutionary algorithms

ABSTRACT

A technique for tuning of decoupled proportional-integral (PI) and proportional-integral-derivative (PID) multivariable controllers based on a chaotic differential evolution (DE) approach is presented in this paper. Due to the simple concept, easy implementation and quick convergence, nowadays DE has gained much attention and wide application in solving continuous non-linear optimization problems. However, the performance of DE greatly depends on its control parameters and it often suffers from being trapped in local optimum. The application of chaotic sequences based on chaotic Zaslavskii map instead of random sequences in DE is a powerful strategy to diversify the population and improve the DE's performance in preventing premature convergence to local optima. The optimized PD and PID controllers shows good closed-loop responses in control of the binary Wood–Berry distillation column, a multivariable process with strong interactions between input and output pairs. Some comparison results of PD and PID tuning using chaotic DE, classical DE and genetic algorithm are presented and discussed.

1. Introduction

The proportional-integral-derivative (PID) controller has remained, by far, as the most commonly used controller in practically all industrial control applications. More than 90% of all control loops are PID, with a wide range of applications: process control, motor drives, magnetic memories, automotive, flight control, among others (Åström & Hägglund, 2001). The reason is that it has a simple structure which is easy to be understood by the engineers. In the past decades, different tuning methodologies of PI and PID controllers have been proposed in the literature such as auto-tuning, self-tuning, pattern recognition, and computational intelligence (Åström & Hägglund, 1995, 2001; Cominos & Munro, 2002).

Unfortunately, it has been quite difficult to tune properly the gains of PID controllers because many industrial plants are often burdened with problems such as high order, time delays, poorly damped, nonlinearities, and time-varying dynamics.

Over the years, several authors have proposed the tuning of PID to control monovariable processes by optimization methods, such as genetic algorithms (Altınten, Ketevanlioğlu, Erdoğan, Hapoğlu, & Alpbaz, 2008; Huang & Chen, 1997; Hwang & Thompson, 1993; Jan, Tseng, & Liu, 2008; Krohling & Rey, 2001; Li, Jan, & Shieh, 2003; Takahashi, Peres, & Ferreira, 1997; Wang & Kwok, 1993; Zhang, Zhuang, Du, & Wang, 2009), particle swarm optimization (Chang, 2009; Kim, Maruta, & Sugie, 2008), tribes algorithm (Coelho & Bernert, 2009a), harmony search (Coelho & Bernert, 2009b), evolution strategy (Coelho & Coelho, 1999), ant colony (Duan, Wang, & Yu, 2006), among others (Bianchi, Mantz, & Christiansen, 2008; Toscano, 2005; Toscano & Lyonnet, 2009; Xu, Li, Qi, & Cai, 2005). Moreover, there is literature about the tuning of multivariable PI and PID controllers using genetic algorithms (see details in Zuo (2005), Vlachos, Williams, and Gomm (2002), Chang (2007), Herreros, Baeyens, and Perán (2002)).

Recently, a new evolutionary computation technique, called differential evolution (DE) algorithm, has been proposed in Storn and Price (2005, 1997). The DE has three main advantages: it can find near optimal solution regardless the initial parameter values, its convergence is fast and it uses few number of control parameters. In addition, DE is simple in coding, easy to use and it can handle integer and discrete optimization (Storn & Price, 2005). Due to the good features of DE algorithm, nowadays it has been emerged as a new and attractive optimizer and applied in variety of research fields.

This paper presents a hybrid optimization approach given by combination of DE (Storn & Price, 1997, 2005) with chaotic Zaslavskii map (Zaslavskii, 1978) (DECZ) to determine PID control gains in a multiloop control scheme. The main benefits of DECZ over traditional DE approaches are allowing the diversity maintenance and aids in slowing premature convergence and exploring the search space. A lack of diversity in a population corresponds to sample solutions being very similar with respect to the distance metric. Conversely, when samples are not very similar then the degree of

* Corresponding author. Tel.: +55 41 3271 13 33; fax: +55 41 3271 13 45.
E-mail addresses: leandro.coelho@pucpr.br (L.S. Coelho), celowp@gmail.com (M.W. Pessôa).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.156

diversity is high. In this context, the chaotic Zaslavskii map can be useful in mutation factor tuning in DECZ approach.

The feasibility of the proposed PI and PID schemes based on DECZ tuning is demonstrated on a simulated binary distillation column. Additionally, the simulation results are compared to those obtained using classical tuning based on DE and genetic algorithm with floating point representation.

The remainder of this paper is structured as follows. In Section 2, a description of binary distillation column is provided. Section 3 presents the fundamentals of PI and PID controllers. Section 4 explains the DE algorithm combined with chaotic Zaslavskii map. Simulation results are presented and discussed in Section 5. Finally, Section 6 outlines a brief conclusion about this study.

2. Case study: a distillation column model

Distillation columns are very commonly used separation equipment in chemical and process industries. The Wood–Berry model is a 2×2 (two inputs and two outputs) transfer function model of a pilot plant distillation column that separates methanol and water (Wood & Berry, 1973). The Wood–Berry model is highly coupled and attracts much attention in the literature. This methanol-water column model has been used for several controller studies, including advanced control algorithms (Aceves-López & Aguilar-Martin, 2006; Deshpande & Ash, 1988; Edgar, Postlethwaite, & Gormandy, 2000; Jain & Lakshminarayanan, 2007; Lee & Yu, 1994; Luyben, 1986; Mantz & De Battista, 2002; Shridhar & Cooper, 1997; Wang, Zou, Lee, & Qiang, 1997; Zeng, Chen, & Gao, 2009). The composition of the top and bottom products expressed in weight% of methanol are the controlled variables. The reflux and the reboiler steam flow rates are the manipulated inputs expressed in l b/min. The distillation column is the binary with eight plates, reported by Wood and Berry (1973) that is shown in Fig. 1. The transfer function of distillation column has first order dynamics and significant time delays and it has a strong interaction between inputs and outputs. The linear Wood–Berry model is given by

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21.0s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \cdot \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} + \begin{bmatrix} \frac{3.8e^{-8.1s}}{10.9s+1} \\ \frac{4.9e^{-3.4s}}{13.2s+1} \end{bmatrix} \cdot D(s) \tag{1}$$

where input signals are the reflux flow rate u_1 and the steam flow rate u_2, the output signals are the top product composition y_1 in mole fraction and the bottom product composition y_2 also in mole fraction. The influence of the feed flow rate (in mole fraction) D was taken from Deshpande and Ash (1988). The unmeasured feed flow rate, D, acts as a process disturbance. This linear model is valid around the set point $y_1 = 0.96$ and $y_2 = 0.02$ (Aceves-López & Aguilar-Martin, 2006). The time sampling is 1 min.

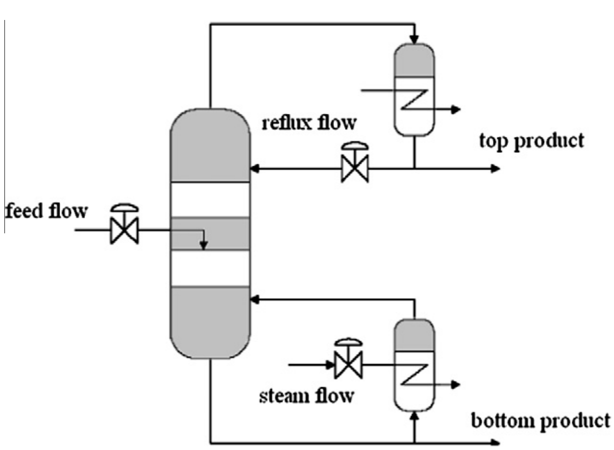

Fig. 1. Simplified schematic diagram of the distillation column (Shridhar & Cooper, 1997).

3. PI and PID control

Multiloop single-input single-output (SISO) controllers are often used to control chemical plants which have multi-input multi-output (MIMO) dynamics. The simple controller structure and the easiness to handle loop failure are the most attractive advantages of such systems. But, inevitably, interactions exist between loops, design of such controllers to meet specifications would then encounter more difficulties than that for a single loop and becomes an open research topic for years.

In this work, consider a process with n inputs and n outputs represented by Chang (2007), where

$$G(s) = \begin{bmatrix} g_{11}(s) & \cdots & g_{1n}(s) \\ \vdots & \ddots & \vdots \\ g_{n1}(s) & \cdots & g_{nn}(s) \end{bmatrix}. \tag{2}$$

A multivariable controller $K(s)$ with $n \times n$ structure is adopted, where

$$K(s) = \begin{bmatrix} k_{11}(s) & \cdots & k_{1n}(s) \\ \vdots & \ddots & \vdots \\ k_{n1}(s) & \cdots & k_{nn}(s) \end{bmatrix}. \tag{3}$$

The form for $k_{ij}(s)$, for $i, j \in \underline{n} = \{1, 2, \ldots, n\}$ is given by

$$k_{ij}(s) = Kp_{ij}\left(1 + \frac{1}{Ti_{ij} \cdot s} + Td_{ij} \cdot s\right), \tag{4}$$

where Kp_{ij} is the proportional gain, Ti_{ij} is the integral time constant, and Td_{ij} is the derivative time constant. The control law of Eq. (4) can be rewritten as

$$k_{ij}(s) = Kp_{ij} + Ki_{ij} \cdot \frac{1}{s} + Kd_{ij} \cdot s, \tag{5}$$

where $Ki_{ij} = Kp_{ij}/Ti_{ij}$ is the integral gain and $Kd_{ij} = Kp_{ij} \cdot Td_{ij}$ is the derivative gain. For convenience, let PI and PID gains for optimization, where

(i) $\theta_{ij} = [Kp_{ij}, Ki_{ij}]^T$ represents the gains vector of ith row and jth column of sub-PI controller in $K(s)$;
(ii) $\theta_{ij} = [Kp_{ij}, Ki_{ij}, Kd_{ij}]^T$ represents the gains vector of ith row and jth column of sub-PID controller in $K(s)$.

In Wood–Berry distillation column with $n = 2$, as shown in Fig. 2, the following objective function is adopted:

$$f = \sum_{k=1}^{N} k \cdot |e_1(k)| + k \cdot |e_2(k)|, \tag{6}$$

where k is the number of sample in the time domain, N is the number of samplings; $e_i(k)$ is the error signal given by difference between the output signal and the setpoint signal. The optimization problem involves finding the PI and PID such that the f performance index is minimized.

The PID controller is able to reduce response error via proportional control, and thus the process can track the input and respond properly; it can lower the output overshoot and reduce the response time through derivative control; and it can eliminate steady-state offset through integral control.

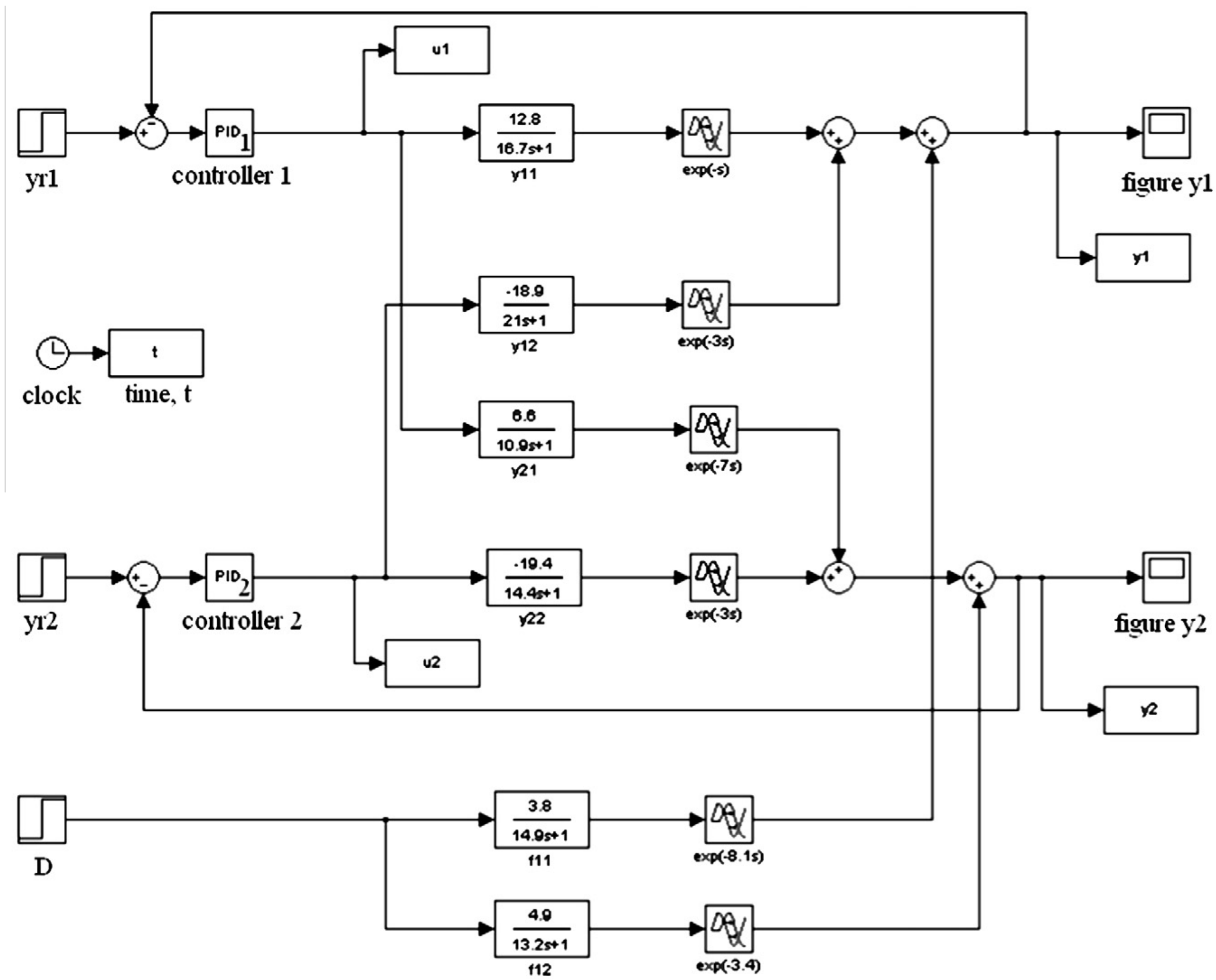

Fig. 2. Representation of adopted multivariable PID control scheme using Matlab/Simulink environment.

4. Fundamentals of differential evolution approaches

This section describes the proposed DECZ algorithm. First, a brief overview of the DE is provided, and finally the proposed DECZ algorithm is presented.

4.1. Differential evolution algorithm

DE is a population-based stochastic function minimizer (or maximizer) relating to evolutionary algorithms (EAs), whose simple yet powerful and straightforward features make it attractive for numerical optimization. Classical DE algorithms use a floating-point encoding EA for global optimization over continuous spaces. Like other EAs, DE is a global optimizer capable of working reliably in non-linear and multimodal environments. Using a few parameters, DE exhibits an overall excellent performance for a wide range of benchmark functions. The advantages of DE, such as a simple, compact structure, ease of use, high convergence characteristics, and robustness, make it a high-class technique for real-valued parameter optimization.

DE combines simple arithmetical operators with the operators of recombination, mutation and selection to evolve from a randomly generated starting population to a final solution. DE uses mutation which is based on the distribution of solutions in the current population. In this way, search directions and possible step sizes depend on the location of the individuals selected to calculate the mutation values. It evolutes generation by generation until the termination conditions have been met.

The different variants of DE are classified using the following notation: DE/α/β/δ, where α indicates the method for selecting the parent chromosome that will form the base of the mutated vector, β indicates the number of difference vectors used to perturb the base chromosome, and δ indicates the recombination mechanism used to create the offspring population. The *bin* acronym indicates that the recombination is controlled by a series of independent binomial experiments.

The fundamental idea behind DE is a scheme whereby it generates the trial parameter vectors. In each step, the DE mutates vectors by adding weighted, random vector differentials to them. If the cost of the trial vector is better than that of the target, the target vector is replaced by the trial vector in the next generation. The variant implemented here was DE/*rand*/1/*bin*, which involved the following steps and procedures:

Step 1: *Initialization of the parameter setup*: The user must choose the key parameters that control the algorithm, i.e., population size (N), boundary constraints of optimization variables, mutation factor (MF), crossover rate (CR), and the stopping criterion (t_{max}).

Step 2: *Initialize the initial population of individuals*: Initialize the generation's counter $t = 0$ and also initialize a population of individuals (solution vectors) $x(t)$ in upper and lower bounds of each decision variable with random values generated according to a uniform probability distribution in the n-dimensional problem space.

Step 3: *Evaluate the objective function value*: For each individual, evaluate its objective function value. The objective function will also be referred to as the *fitness function*. We are minimizing, rather than maximizing, the fitness function given by Eq. (6) in this paper.

Step 4: *Mutation operation (or differential operation)*: Mutate individuals according to the following equation:

$$z_i(t+1) = x_{i_1}(t) + MF \cdot [x_{i_2}(t) - x_{i_3}(t)]. \qquad (7)$$

In Eq. (1), $i = 1, 2, \ldots, N$ is the individual's index of population; t is the time (generation); $x_i(t) = [x_{i_1}(t), x_{i_2}(t), \ldots, x_{i_n}(t)]^T$ stands for the position of the ith individual of population of N real-valued n-dimensional vectors; $z_i(t) = [z_{i_1}(t), z_{i_2}(t), \ldots z_{i_n}(t)]^T$ stands for the position of the ith individual of a *mutant vector*; $MF > 0$ is a real parameter, called *mutation factor*, which controls the amplification of the difference between two individuals so as to avoid search stagnation. The mutation operation randomly select the target vector $x_{i_1}(t)$, with $i \neq i_1$. Then, two individuals $x_{i_2}(t)$ and $x_{i_3}(t)$ are randomly selected with $i_1 \neq i_2 \neq i_3 \neq i$, and the difference vector $x_{i_2} - x_{i_3}$ is calculated.

Step 5: *Crossover (recombination) operation*: Following the mutation operation, crossover is applied in the population. For each mutant vector, $z_i(t+1)$, an index $rnbr(i) \in \{1, 2, \ldots, n\}$ is randomly chosen using a uniform distribution, and a *trial vector*, $u_i(t+1) = [u_{i_1}(t+1), u_{i_2}(t+1), \ldots, u_{i_n}(t+1)]^T$, is generated via

$$u_{i_j}(t+1) = \begin{cases} z_{i_j}(t+1) & \text{if } randb(j) \leqslant CR \text{ or } j = rnbr(i), \\ x_{i_j}(t) & \text{otherwise}, \end{cases} \qquad (8)$$

where $j = 1, 2, \ldots, n$ is the parameter index; $x_{ij}(t)$ stands for the ith individual of jth real-valued vector; $z_{ij}(t)$ stands for the ith individual of jth real-valued vector of a *mutant vector*; $u_{ij}(t)$ stands for the ith individual of jth real-valued vector after crossover operation; $randb(j)$ is the jth evaluation of a uniform random number generation with $[0, 1]$; CR is a *crossover rate* in the range $[0, 1]$.

To decide whether or not the vector $u_i(t+1)$ should be a member of the population comprising the next generation, it is compared to the corresponding vector $x_i(t)$. Thus, if f denotes the objective function under minimization, then

$$x_i(t+1) = \begin{cases} u_i(t+1) & \text{if } f(u_i(t+1)) < f(x_i(t)), \\ x_i(t) & \text{otherwise} \end{cases} \qquad (9)$$

Step 6: *Update the generation's counter*: $t = t + 1$;

Step 7: *Verification of the stopping criterion*: Loop to Step 3 until a stopping criterion is met. In this paper, a maximum number of iterations (generations), $t_{\max}$, is adopted.

4.2. Proposed DECZ algorithm

The behavior and performance of DE is influenced both by the mutation and crossover operators and by the values of the involved control parameters (e.g. MF and CR). Wrong choices of values for these parameters may result in divergent or cyclic individual trajectories. A number of variations based on empirical and theoretical studies have been developed in the past decade to address the problem of finding insights concerning the behavior of DE algorithms (see Angira & Santosh, 2007; Babu & Angira, 2006; Brest, Greiner, Boskovic, Mernik, & Zumer, 2006; Coelho & Mariani, 2007; Coelho, Souza, & Mariani, 2009; Omran, Engelbrecht, & Salman, 2009; Ponsich & Coello Coello, 2011; Zhang, Luo, & Wang, 2008). The adaptive and self-adaptive variants of DE (Brest et al., 2006; Omran et al., 2009) are based on exploring the control parameters space using deterministic or random strategies.

Recently, the idea of using chaotic systems instead of random strategies has been noticed in several fields (Cai, Ma, Li, Li, & Peng, 2009; Coelho & Lee, 2008; Modares, Alfi, & Fateh, 2010; Unsihuay-Vila, Souza, Marangon-Lima, & Balestrassi, 2010; Wu, 2010). Chaos is a kind of characteristics of non-linear system that demonstrates sensitive dependence on initial conditions and also includes infinite unstable periodic motions. Due to non-repetitive nature of

Table 1
Results of objective function in 30 runs.

Optimization method	Type of multivariable controller	Best objective function after 100 generations in 30 runs			
		Minimum (best)	Maximum (worst)	Mean	Standard deviation
DE	PI	31.316	31.511	32.323	0.036
DECZ	PI	31.310	31.316	31.312	$8.3526 \cdot 10^{-4}$
GA	PI	31.342	107.150	91.71	91.061
DE	PID	13.836	18.060	14.706	1.022
DECZ	PID	13.810	13.829	13.857	0.069
GA	PID	14.112	148.317	23.691	23.601

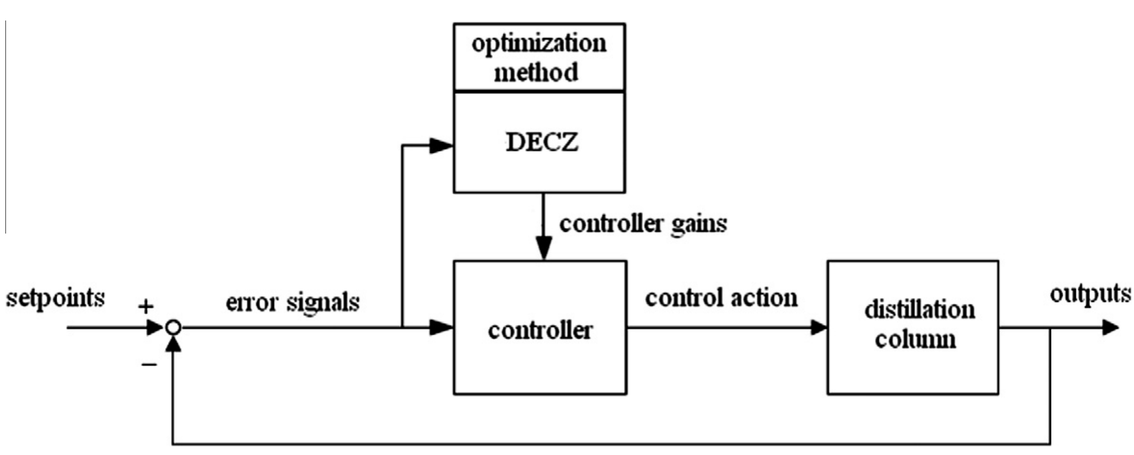

Fig. 3. Diagram of multivariable controller design based on DECZ.

Table 2
Best results in multivariable PI controller design using DE, DECZ, and GA.

Parameters	DE	DECZ	GA
$K_{p,1}$	1.8934	1.8933	1.9153
$K_{i,1}$	0.2941	0.2939	0.2755
$K_{p,2}$	−0.1319	−0.1319	−0.1384
$K_{i,2}$	−0.0205	−0.0205	−0.0210
F	31.316	31.310	31.342
IAE(y_1)	0.0649	0.0649	0.0664
IAE(y_2)	0.8263	0.8262	0.8176
Mean (u_1)	0.0059	0.0058	0.0059
Mean (u_2)	0.0132	0.0132	0.0132
Variance (u_1)	$1.1823 \cdot 10^{-5}$	$1.1821 \cdot 10^{-5}$	$1.1656 \cdot 10^{-5}$
Variance (u_2)	$2.5641 \cdot 10^{-5}$	$2.5639 \cdot 10^{-5}$	$2.5625 \cdot 10^{-5}$

Table 3
Best results in multivariable PID controller design using DE, DECZ, and GA.

Parameters	DE	DECZ	GA
$K_{p,1}$	1.9866	1.7840	1.9248
$K_{i,1}$	0.4643	0.5036	0.5285
$K_{d,1}$	1.0242	1.0243	1.0755
$K_{p,2}$	−0.2254	−0.2229	−0.2214
$K_{i,2}$	−0.1008	−0.1017	−0.0882
$K_{d,2}$	−0.4123	−0.4172	−0.3690
F	13.836	13.810	14.112
IAE(y_1)	0.0476	0.0469	0.0434
IAE(y_2)	0.3866	0.3875	0.3997
Mean (u_1)	0.0063	0.0063	0.0063
Mean (u_2)	0.0135	0.0135	0.0135
Variance (u_1)	$1.1522 \cdot 10^{-5}$	$1.0961 \cdot 10^{-5}$	$1.0794 \cdot 10^{-5}$
Variance (u_2)	$2.7249 \cdot 10^{-5}$	$2.7248 \cdot 10^{-5}$	$2.6881 \cdot 10^{-5}$

chaos, it can carry out overall searches at higher speeds than stochastic ergodic searches that is probabilistic in nature.

With the easy implementation and special ability to avoid being trapped in local optimum, chaos and chaos-based searching algorithms can be a good alternative to maintain the search diversity in stochastic optimization procedures.

In the proposed DECZ, we opt for Eq. (12) with an adaptive mutation factor based on Zaslavskii map (Zaslavskii, 1978) instead of Eq. (7) with *MF* fixed. In DECZ approach, the generation of new individuals (Step 4 of DE described in Section 4.1) in mutation operation is adjusted by following equations:

$$w(t) = \mathrm{mod}[w(t-1) + v + a \cdot ZF(t), 1], \tag{10}$$

$$ZF(t) = \cos(2\pi \cdot w(t-1)) + e^{-r} \cdot ZF(t-1), \tag{11}$$

$$z_i(t+1) = x_{i_1}(t) + NZF(t) \cdot [x_{i_2}(t) - x_{i_3}(t)], \tag{12}$$

where $NZF(t)$ normalized chaotic Zaslavskii map represented by variable $ZF(t)$ with values in range [0.3, 0.7], mod is the modulus after division. The Zaslavskii map shows a strange attractor with the largest Lyapunov exponent for $v = 400$, $r = 3$, and $a = 12.6695$

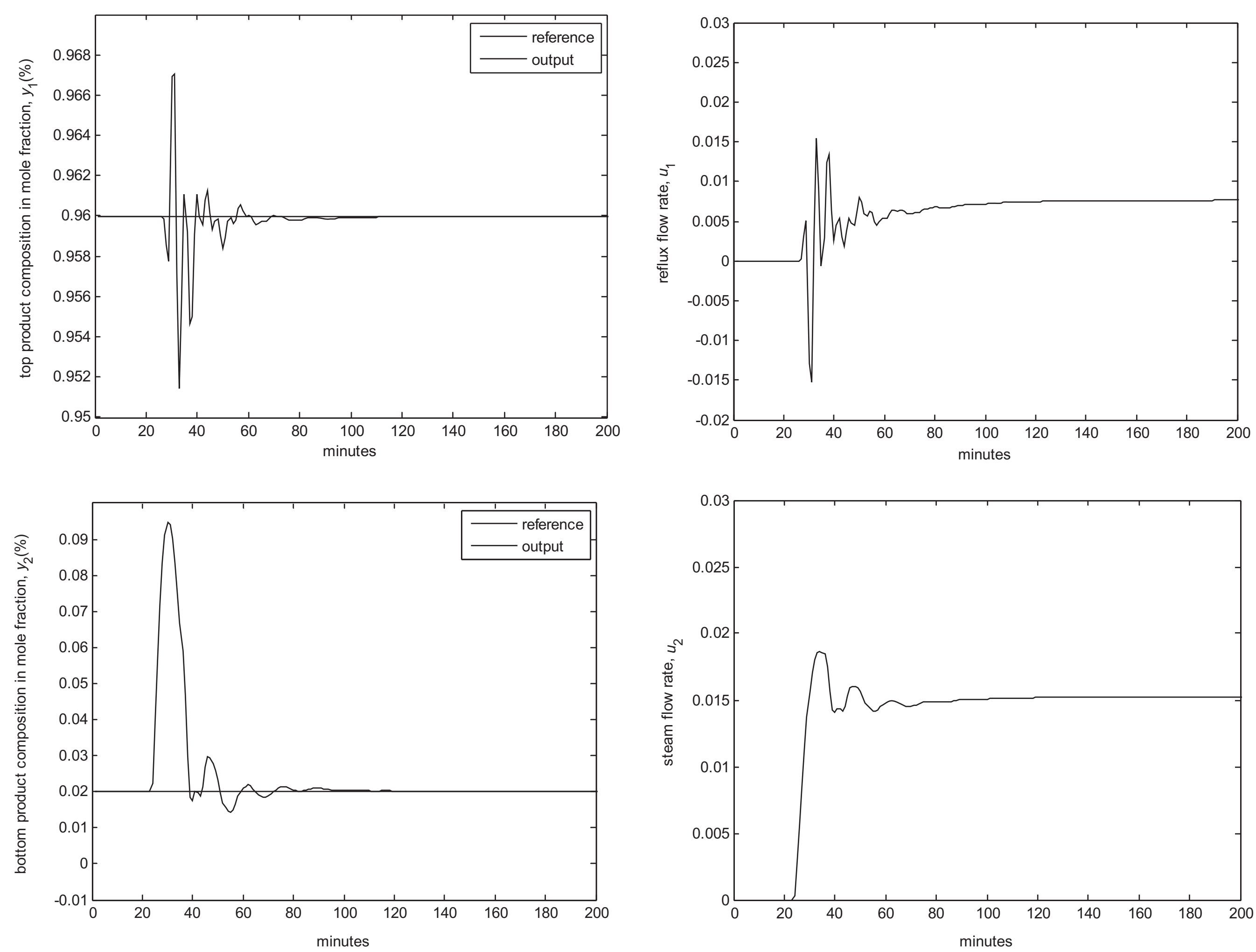

Fig. 4. Result in closed-loop of multivariable PI controller tuned by DECZ.

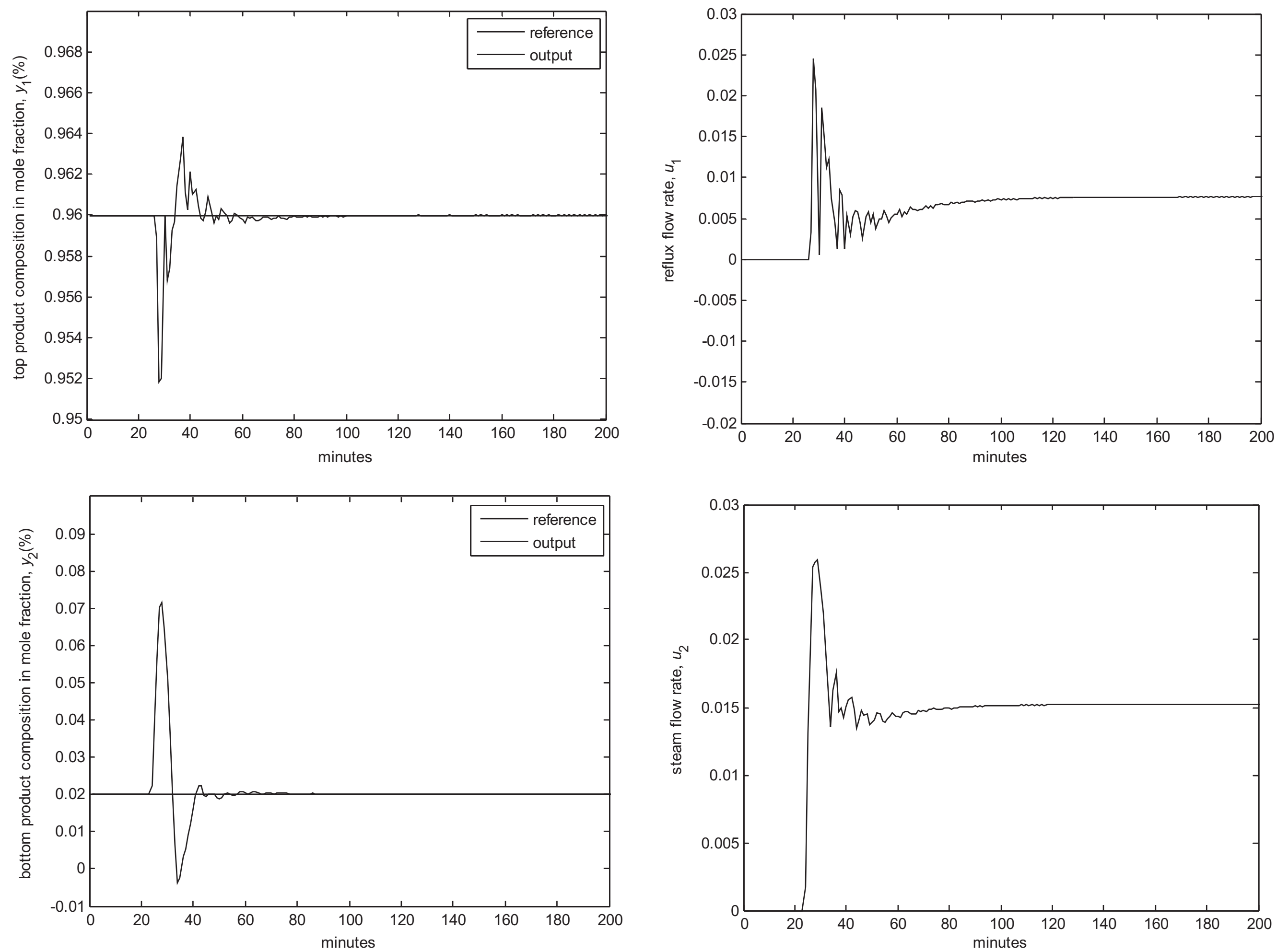

Fig. 5. Result in closed-loop of multivariable PID controller tuned by DECZ.

(Coelho, 2008). The initial values of w and ZF are equal to zero. The choice of the chaotic Zaslavskii map in DECZ design is justified theoretically by its unpredictability, i.e., by its spread-spectrum characteristic and large Lyapunov exponent (a quantitative measure of chaos) (Coelho & Bernert, 2009b; Grossberger & Procaccia, 1983; Russell, Hanson, & Ott, 1980).

5. Simulation results

This section compares the performance of the DECZ with that of the classical DE and a genetic algorithm (GA). For the classical DE, $MF = 0.5$ and $CR = 0.9$ are adopted, as suggested in Storn and Price (2005). For the DECZ, $CR = 0.9$ is adopted. For the GA algorithm with floating point representation, roulette wheel selection with elistim strategy is adopted. The crossover and mutation probabilities in GA are 0.8 and 0.05, respectively.

For all optimization algorithms is adopted population size $N = 15$ and $t_{max} = 100$ generations. The initial population was generated from a uniform distribution in the ranges specified below. The lower and upper bounds of the search space is given to both PI ($K_{p,1}$, $K_{i,1}$, $K_{p,2}$, $K_{i,2}$) and PID ($K_{p,1}$, $K_{i,1}$, $K_{d,1}$, $K_{p,2}$, $K_{i,2}$, $K_{d,2}$) in the range $[-0.5, 2]$.

Each optimization method was implemented in Matlab (MathWorks). All the programs were run on a 3.2 GHz Pentium IV processor with 2 GB of random access memory. In each case study, 30 independent runs were made for each of the optimization algorithms involving 30 different initial trial solutions for each optimization method. In this paper, the DE, DECZ and GA approaches are adopted using 1500 objective function evaluations in each run. The adopted block diagram of controller design using DECZ is presented in Fig. 3.

Table 1 summarizes the results obtained by applying the different optimization approaches to the multivariable PI and PID control. Simulation results show that the DECZ is the best performer followed by the DE and GA in terms of minimum and mean objective function in 30 runs. Best gains obtained to the multivariable PI and PID are presented in Tables 2 and 3, respectively. In these tables, the acronyms IAE(y_1) and IAE(y_2) are the sum of absolute errors of outputs y_1 and y_2, respectively.

In Figs. 4 and 5 are presented the best results of multivariable PI and PID controllers using DECZ optimization. It is observed that the PID approach tuned by DECZ is better than the PI control in closed-loop in terms of tracking the reference (setpoint) signals.

6. Conclusion and future research

Although many PI and PID controllers have been proposed for decades of years, it is still widely applied in industrial applications at present because its simple structure, easy implementation and robustness to noise. But adjusting the parameters of PI and PID controllers, mainly in multivariable case, to fit different controlled plants is complicated.

In this paper we have proposed a DECZ approach using chaos theory concepts to improve the classical DE. Our proposed DECZ is based on an increased diversity based on Zaslavskii map in mutation factor tuning.

Performance evaluation of DECZ on the design of multivariable PI and PID controller for the binary Wood–Berry distillation column was conducted in this paper. Simulation results and comparisons with DE and GA algorithms in control of the binary Wood–Berry distillation column showed that the DECZ is a promising approach in terms of solution quality.

Further analysis should be conducted in order to improve the DECZ's performance in tuning of other multivariable control systems based on adaptive, fuzzy and predictive controllers.

Acknowledgment

This work was supported by the National Council of Scientific and Technologic Development of Brazil — CNPq — under Grant No. 303963/2009-3/PQ.

References

Aceves-López, A., & Aguilar-Martin, J. (2006). A simplified version of Mamdani's fuzzy controller: the natural logic controller. *IEEE Transactions on Fuzzy Systems, 14*(1), 16–30.

Altınten, A., Ketevanlioğlu, F., Erdoğan, S., Hapoğlu, H., & Alpbaz, M. (2008). Self-tuning PID control of jacketed batch polystyrene reactor using genetic algorithm. *Chemical Engineering Journal, 138*(1–3), 490–497.

Angira, R., & Santosh, A. (2007). Optimization of dynamic systems: A trigonometric differential evolution approach. *Computers & Chemical Engineering, 31*(9), 1055–1063.

Åström, K. J., & Hägglund, T. (1995). *PID controllers: Theory, design and tuning* (2nd ed.). North Carolina, USA: Instrument Society of America, Research Triangle Park.

Åström, K. J., & Hägglund, T. (2001). The future of PID control. *Control Engineering Practice, 9*(11), 1163–1175.

Babu, B. V., & Angira, R. (2006). Modified differential evolution (MDE) for optimization of non-linear chemical processes. *Computers & Chemical Engineering, 30*(6–7), 989–1002.

Bianchi, F. D., Mantz, R. J., & Christiansen, C. F. (2008). Multivariable PID control with set-point weighting via BMI optimisation. *Automatica, 44*(2), 472–478.

Brest, J., Greiner, S., Boskovic, B., Mernik, M., & Zumer, V. (2006). Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Transactions on Evolutionary Computation, 10*(6), 646–657.

Cai, J., Ma, X., Li, Q., Li, L., & Peng, H. (2009). A multi-objective chaotic particle swarm optimization for environmental/economic dispatch. *Energy Conversion and Management, 50*(5), 1318–1325.

Chang, D.-W. (2007). A multi-crossover genetic approach to multivariable PID controllers tuning. *Expert Systems with Applications, 33*(3), 620–626.

Chang, W. D. (2009). PID control for chaotic synchronization using particle swarm optimization. *Chaos, Solitons & Fractals, 39*(2), 910–917.

Coelho, L. S. (2008). A quantum particle swarm optimizer with chaotic mutation operator. *Chaos, Solitons & Fractals, 37*(5), 1409–1418.

Coelho, L. S., & Bernert, D. L. A. (2009a). PID control design for chaotic synchronization using a tribes optimization approach. *Chaos Solitons & Fractals, 42*(1), 634–640.

Coelho, L. S., & Bernert, D. L. A. (2009b). An improved harmony search algorithm for synchronization of discrete-time chaotic systems. *Chaos, Solitons & Fractals, 41*(5), 2526–2532.

Coelho, L. S., & Coelho, A. A. R. (1999). Automatic tuning of PID and gain scheduling PID controllers by a derandomized evolution strategy. *Artificial Intelligence, for Engineering Design, and Manufacturing, 13*(5), 365–373.

Coelho, L. S., & Lee, C.-S. (2008). Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches. *International Journal of Electrical Power & Energy Systems, 30*(5), 297–307.

Coelho, L. S., & Mariani, V. C. (2007). Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints. *Energy Conversion and Management, 48*(5), 1631–1639.

Coelho, L. S., Souza, R. C. T., & Mariani, V. C. (2009). Improved differential evolution approach based on cultural algorithm and diversity measure applied to solve economic load dispatch problems. *Mathematics and Computers in Simulation, 79*(10), 3136–3147.

Cominos, P., & Munro, N. (2002). PID controllers: Recent tuning methods and design to specification. *IEE Proceedings-Control Theory and Applications, 149*(1), 46–53.

Deshpande, P. B, & Ash, R. A (1988). *Computer process control with advanced control applications* (2nd ed.). Research Triangle Park NC: ISA.

Duan, H., Wang, D., & Yu, X. (2006). Novel approach to nonlinear PID parameter optimization using ant colony optimization algorithm. *Journal of Bionic Engineering, 3*(2), 73–78.

Edgar, C. R., Postlethwaite, B. E., & Gormandy, B. A. (2000). Control of MIMO dead time processes using fuzzy relational models. In *Proceedings of European symposium on intelligent techniques, Aachen, Germany*.

Grossberger, P., & Procaccia, I. (1983). Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena, 9*(1-2), 189–208.

Herreros, A., Baeyens, E., & Perán, J. R. (2002). Design of PID-type controllers using multiobjective genetic algorithms. *ISA Transactions, 41*(4), 457–472.

Huang, P.-Y., & Chen, Y.-Y. (1997). Design of PID controller for precision positioning table using genetic algorithms. In *Proceedings of 36th conference on decision and control, San Diego, CA, USA* (pp. 2513–2514).

Hwang, W. R., & Thompson, W. E. (1993). An intelligent controller design based on genetic algorithms. In *Proceedings of 32nd conference on decision and control, San Antonio, TX, USA* (pp. 1266–1267).

Jain, M., & Lakshminarayanan, S. (2007). Estimating performance enhancement with alternate control strategies for multiloop control systems. *Chemical Engineering Science, 62*(17), 4644–4658.

Jan, R.-M., Tseng, C.-S., & Liu, R.-J. (2008). Robust PID control design for permanent magnet synchronous motor: A genetic approach. *Electric Power Systems Research, 78*(7), 1161–1168.

Kim, T.-H., Maruta, I., & Sugie, M. (2008). Robust PID controller tuning based on the constrained particle swarm optimization. *Automatica, 44*(4), 1104–1110.

Krohling, R. A., & Rey, J. P. (2001). Design of optimal disturbance rejection PID controllers using genetic algorithms. *IEEE Transactions on Evolutionary Computation, 5*(1), 78–82.

Lee, J. H., & Yu, Z. H. (1994). Tuning of model predictive controllers for robust performance. *Computers & Chemical Engineering, 18*(1), 15–37.

Li, C. -L., Jan, H. -Y., & Shieh, N. -C. (2003). GA-based multiobjective PID control for a linear brushless DC motor. *IEEE/ASME Transactions on Mechatronics, 8*(1), 56–65.

Luyben, W. L. (1986). A simple method for tuning SISO controllers in a multivariable system. *Industrial & Engineering Chemistry Product Research and Development, 25*(3), 654–660.

Mantz, R. J., & De Battista, H. (2002). Sliding mode compensation for windup and direction of control problems in two-input–two-output proportional-integral controllers. *Industrial & Engineering Chemistry Research, 41*(13), 3179–3185.

Modares, H., Alfi, A., & Fateh, M. M. (2010). Parameter identification of chaotic dynamic systems through an improved particle swarm optimization. *Expert Systems with Applications, 37*(5), 3714–3720.

Omran, M. G. H., Engelbrecht, A. P., & Salman, A. (2009). Bare bones differential evolution. *European Journal of Operational Research, 196*(1), 128–139.

Ponsich, A., & Coello Coello, C. A. (2011). Differential evolution performances for the solution of mixed-integer constrained process engineering problems. *Applied Soft Computing, 11*(1), 399–409.

Russell, D. A., Hanson, J. D., & Ott, E. (1980). Dimension of strange attractors. *Physical Review Letters, 45*(14), 1175–1180.

Shridhar, R., & Cooper, D. J. (1997). A tuning strategy for unconstrained SISO model predictive control. *Industrial & Engineering Chemistry Research, 36*(3), 729–746.

Storn, R., & Price, K. (2005). *Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces*. Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, USA.

Storn, R., & Price, K. (1997). Differential evolution — A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341–359.

Takahashi, R. H. C., Peres, P. L. D., & Ferreira, P. A. V. (1997). Multiobjective H2/H∞ guaranteed cost PID design. *IEEE Control Systems, 17*(5), 37–47.

Toscano, R. (2005). A simple robust PI/PID controller design via numerical optimization approach. *Journal of Process Control, 15*(1), 81–88.

Toscano, R., & Lyonnet, P. (2009). Robust PID controller tuning based on the heuristic Kalman algorithm. *Automatica, 45*(9), 2099–2106.

Unsihuay-Vila, C., Souza, A. C. Z., Marangon-Lima, J. W., & Balestrassi, P. P. (2010). Electricity demand and spot price forecasting using evolutionary computation combined with chaotic nonlinear dynamic model. *International Journal of Electrical Power & Energy Systems, 32*(2), 108–116.

Vlachos, C., Williams, D., & Gomm, J. B. (2002). Solution to the Shell standard control problem using genetically tuned PID controllers. *Control Engineering Practice, 10*(2), 151–163.

Wang, P., & Kwok, D. P. (1993). Optimal design of PID process controllers based on genetic algorithms. In *Proceedings of 12th World congress of IFAC, Sydney, Australia* (Vol. 5, pp. 261–265).

Wang, Q. G., Zou, B., Lee, T. H., & Qiang, B. (1997). Auto-tuning of multivariable PID controllers from decentralized relay feedback. *Automatica, 33*(3), 319–330.

Wood, R. K., & Berry, M. W. (1973). Terminal composition control of a binary distillation column. *Chemical Engineering Science, 28*(9), 1707–1717.

Wu, Q. (2010). A hybrid-forecasting model based on Gaussian support vector machine and chaotic particle swarm optimization. *Expert Systems with Applications, 37*(3), 2388–2394.

Xu, M., Li, S., Qi, C., & Cai, W. (2005). Auto-tuning of PID controller parameters with supervised receding horizon optimization. *ISA Transactions, 44*(4), 491–500.

Zaslavskii, G. M. (1978). The simplest case of a strange attractor. *Physics Letters A, 69*(3), 145–147.

Zeng, Q., Chen, Z., & Gao, Z. (2009). A practical approach to disturbance decoupling control. *Control Engineering Practice, 17*(9), 1016–1025.
Zhang, M., Luo, W., & Wang, X. (2008). Differential evolution with dynamic stochastic selection for constrained optimization. *Information Sciences, 178*(15), 3043–3074.
Zhang, J., Zhuang, J., Du, H., & Wang, S. (2009). Self-organizing genetic algorithm based tuning of PID controllers. *Information Sciences, 179*(7), 1007–1018.
Zuo, W. (2005). Multivariable adaptive control for a space station using genetic algorithms. *IEE Proceedings-Control Theory and Applications, 142*(2), 81–87.