Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1999), *13*, 341–349. Printed in the USA.

Automatic tuning of PID and gain scheduling PID controllers by a derandomized evolution strategy

LEANDRO DOS SANTOS COELHO AND ANTONIO AUGUSTO RODRIGUES COELHO
Federal University of Santa Catarina, Department of Automation and Systems, P.O. Box 476, 88040.900 Florianópolis SC, Brazil
(RECEIVED November 19, 1990; ACCEPTED June 3, 1999)

Abstract

This paper evaluates an evolution strategy to tune conventional proportional plus integral plus derivative (PID) and gain scheduling PID control algorithms. The approach deals with the utilization of an evolution strategy with learning acceleration by derandomized mutative step-size control using accumulated information. This technique is useful to obtain the following characteristics: (1) freedom of choice of a performance index, (2) increase of the convergence speed of evolution strategies to get a local minimum to determine controller design parameters, and (3) flexibility and robustness in the automatic design of controllers. Performance analysis and experimental results are carried out using a laboratory scale nonlinear process fan and plate. The practical prototype contains features such as nonminimum phase, dead time, resonant, and turbulent disturbance behavior that motivate the utilization of intelligent control techniques.

Keywords: PID Control, Gain Scheduling Control, Evolutionary Computation, Evolution Strategies, Experimental Nonlinear Process

1. INTRODUCTION

Advanced techniques to design industrial control systems are, in general, dependent of mathematical models for the controlled process. In addition, the task of the controllers is to achieve optimum performance when faced with various types of disturbance that are unknown in most practical applications (Åström & Wittenmark, 1989).

Despite the huge development in control theory, the majority of industrial processes are controlled by the well-established proportional plus integral plus derivative (PID) controller. The popularity of PID control can be attributed to its simplicity (in terms of design and from the point of view of parameter tuning) and to its good performance in a wide range of operating conditions. However, PID controllers present as a disadvantage, the need of retuning whenever the processes are subjected to some kind of disturbance or when processes present complexities (nonlinearities). So, over the last few years, significant development has been established in the process control area to adjust the PID controller parameters in an automatic way, in order to ensure adequate servo and regulatory behavior for a closed-loop plant (Åström & Wittenmark, 1989; Coelho et al., 1998; Coelho & Coelho, 1998; VanDoren, 1998).

In recent decades, the theory and the practice relative to the process control area have received great attention and the importance of having well-behaved control loops have been recognized in the academic and industrial environments for a long time. PID provides low cost, implementation simplicity, and when adequately tuned, provides good dynamic behavior for the controlled process. In general, the tuning parameters K_p (proportional gain), T_i (integral time), and T_d (derivative time) can be based on different methods.

Åström and Wittenmark (1989) present a method in which a limit cycle oscillation is enforced on the process to be controlled by a relay with suitable values of amplitude and hysteresis, so that angular frequency and critical gain values can be found from the amplitude and frequency of the resulting process output with controlled oscillation. Hang et al. (1991) present the a procedure to refine the Ziegler and Nichols (1942) tuning formula in the context of PID and *PI* autotuning. Yamamoto and Shah (1998) treat a new multivariable self-tuning PID controller design scheme and the proposed scheme is experimentally evaluated on a 2 × 2 level plus temperature control system. Yusof and Omatu (1993) present

Reprint requests to: Antonio Augusto Rodrigues Coelho, Federal University of Santa Catarina, Department of Automation and Systems, P.O. Box 476, 88040. 900 Florianópolis SC, Brazil. Phone: 55 48 3319202; Fax: 55 48 3319770; E-mail: aarc@lcmi.ufsc.br

https://doi.org/10.1017/S0890060499135042 Published online by Cambridge University Press

a multivariable self-tuning controller with a PID structure with a combination of the self-tuning property, in which the controller parameters are tuned automatically on-line. Willis and Montague (1993) treat the benefits to be gained utilizing nonlinear neural networks process models for PID control system design. Malki et al. (1997) present the design and experiment of a fuzzy PID controller for a flexible robot arm driven by a dc motor in a laboratory with uncertainties from time-varying loads.

In the literature, several authors have proposed the tuning of PID controllers by genetic algorithms (GAs). Wang and Kwok (1993) presented a comparative study of PID design based on GAs, Ziegler–Nichols rules and pattern recognition method of Hooke–Jeeves and evaluated in a simulation study of a pH process. Hwang and Thompson (1993) dealt with PID control tuning in a digital form by GAs, applied to an unstable simulated process of third order. Lo Bianco and Piazzi (1996) developed the optimum PID design for H_∞ control via GAs with uncertainties through simulation. Huang and Chen (1997) shown the PID design by GAs in controlling a precision positioning table. Chen et al. (1995) developed a PID control combining H_2/H_∞ norms applied to a process with uncertainties. Takahashi et al. (1997) presented the multiobjective H_2/H_∞ type PID based on gradient and on GAs, applied to dc motor model.

In this paper, the parameter tuning task of conventional PID and gain scheduling PID (GS-PID) controllers with application in a practical nonlinear process is assessed. The tuning procedure of controller gains utilizes an evolution strategy (ES) algorithm with derandomized mutative step-size control using accumulated information. The fan and plate prototype contains features of nonminimum phase, dead-time, resonant and turbulent disturbance behavior and therefore motivates the utilization of intelligent control techniques.

The paper is organized as follows: In Section 2, notions of evolutionary computation and description of evolution strategies are presented. In Section 3, procedure and control tuning descriptions of a PID and a GS-PID are shown. The description of the fan and plate nonlinear process is presented in Section 4. Experimental results and performance analysis of optimized controllers are described in Section 5. Finally, conclusions and directions for future work are shown in Section 6.

2. EVOLUTIONARY COMPUTATION

Evolution can be regarded as a sequence of self-organization steps, that is, as the underlying universal principle of any kind of self-organization. Evolutionary computation (*EC*) is a field of research that use computational models of evolutionary processes as key elements in the design and implementation of computer-based problem-solving systems. It is important to note that the field of *EC* is not more than a small part of a greater, more complex scientific universe that, incorporating fuzzy systems and neural networks, is referred to by some authors as computational intelligence and by others as soft computing (Zadeh, 1994; Miranda et al., 1998). The main *EC* algorithms investigated by researchers in the field of *EC* include genetic algorithms, evolutionary programming, evolution strategies, and genetic programming (Bäck et al., 1997).

Evolutionary algorithms (EA) mimic the process of natural evolution, the driving process for the emergence of complex and well-adapted organic structures. To put it succinctly and with strong simplifications, evolution is the result of the interplay between the creation of new genetic information and its evaluation and selection. The effectiveness and simplicity of *EAs* have led many researchers to argue they are methods of choice for hard real-life problems, because they are especially capable of handling optimization problems in which the objective function are nonconvex, discontinuous or nondifferentiable, noisy, multiobjective or multimodal (Goldberg, 1989).

Classical methods to look for a "best" solution to such problems often rely on the use of a gradient-based search. Unfortunately, the error response surface to be examined is often a general nonlinear function possessing multiple local optima. Gradient methods are guaranteed to converge to locally optimal solutions if the step size tends to zero. But such solutions may be far from the global optima and may not provide adequate system performance (Fogel, 1995; Goldberg, 1989).

EAs have been successfully applied to solve hard problems in many fields of study, such as search, optimization, scheduling, pattern recognition, image classification, process identification, and control (Chipperfield & Fleming, 1996; Man et al., 1996). The next section describes the evolution strategies utilized in tuning of controller parameters.

2.1. Evolution strategies

Similar to GAs, ESs are algorithms that imitate the principles of natural evolution as a method to solve parameter optimization problems. ESs are an abstraction of evolution at individual behavior level, stressing the behavioral link between an individual and its offspring, while GAs maintain the genetic link. ES uses the mutation operator as main operator, it works directly with floating point vectors, and it allows self-adaptation of strategy parameters through standard deviation and covariances (Bäck et al., 1997; Miranda et al., 1998).

ESs were developed by Rechenberg and Schwefel in the 1960s, at the Technical University of Berlin, Institute of Fluid Mechanics, during their studies of aerotechnology and space technology (Rechenberg, 1965; Bäck et al., 1997). Rechenberg (1965) developed a theory of convergence velocity for the so-called $(1+1)$-ES, a simple mutation-selection mechanism working on one individual that creates one offspring per generation by means of Gaussian mutation. He also proposed a first multimember ES, a $(\mu + 1)$-ES, where $\mu \geq 1$

https://doi.org/10.1017/S0890060499135042 Published online by Cambridge University Press

individuals recombine to form one offspring, which after mutation eventually replaces the worst parent individual.

Schwefel (1981) introduced recombination and populations with more than one individual, and provided a comparative study of ESs with more traditional optimization techniques. The motivation to extend the (1 + 1)-ES and $(\mu + 1)$-ES to a $(\mu + \lambda)$-ES and a (μ, λ)-ES has been two aspects of essential importance: the use parallel computers, and to enable self-adaptation of strategic parameters like the standard deviations of the mutations. The nomenclature $(\mu + \lambda)$-ES suggests, μ parents produce λ offspring and the whole population is reduced again to the μ parents of the next generation; in other words, the selection operates on the joined set of parents and offspring. Thus, parents survive until they are superseded by better offspring. The (μ, λ)-ES suggests that only the offspring undergo selection, whereas the ancestors are forgotten. If $\mu > 1$, then the principle of recombination can be introduced (Davidor & Schwefel, 1992; Bäck & Schwefel, 1993).

Comparable with other optimization techniques, the performance of ESs depends on a suitable choice of internal strategy control parameters. Apart from a fixed setting, an ES facilitates an adjustment of such parameters within a self-adaptation process, while in conventional GAs the control parameters are adjusted by trial-and-error methods. The self-adaptation of strategy parameters provides one of the main features of the success of ESs because ESs use evolutionary principles to search in the space of object variables and strategy parameters simultaneously.

2.1.1. ES with learning acceleration by derandomized mutative step-size control

The principle of self-adaptation, as mainly utilized in ESs, facilitates the implicit control of strategy parameters by incorporating them into the representation of individuals and by evolving the strategy parameters themselves in analogy with the usual evolution of object variables. The term strategy parameters refers to parameters that control the evolutionary search process, such as mutation rates, mutation variances, and recombination probabilities, and the idea of self-adaptation consists in evolving these parameters in analogy to the object variables themselves (Bäck et al., 1997).

Most of the research of self-adaptation principles in *EAs* deal with parameters related to the mutation operator. The technique of self-adaptation is most widely utilized for the variances and covariances of a generalized *n*-dimensional normal distribution (Schwefel, 1981; Fogel, 1995).

The power of an ES is mainly based upon its ability to perform a second-level optimization of strategy parameters. This process adapts the mutation strength, in such a way that the whole algorithm presents near-optimal performance. There are different possibilities of constructing such strategies. The simplest is the *1/5 success probability rule* of Rechenberg (1973), where the ratio of successful mutations to all mutations should be 1/5. If it is greater than 1/5, increase the mutation variance; if it is less, decrease the variance, σ^2 (Beyer, 1996). This algorithm works satisfactorily in most problems optimization, but it depends on the applicability of an external model of parameter space topology and it only able one general step-size, but no an individual step-size (Ostermeier et al., 1995*a*).

The mutation operator in an ES realizes a kind of hill-climbing search procedure, when it is considered in combination with selection operator. With dedicated standard deviation, σ, for each object variable preferred directions of search may be established only along the axes of the coordinate system. In general, the best search direction (the gradient) is not aligned on those axes. Thus, an optimal rate of progress is achieved only by chance when suitable mutations coincide (when they are correlated). Otherwise, the trajectory of the population through the search space is zig-zagging along the gradient (Bäck et al., 1991; Davidor & Schwefel, 1992).

To avoid this reduction of the rate of progress, the mutation operator can be extended again to handle correlated mutations. Schwefel (1981) proposes the extension the mutation operator to handle correlated mutations that require an additional strategy vector with the idea to approximate the inverse of the Hessian probabilistically simultaneous to the optimization process. Rudolph (1992) shows that the probabilistic approximation procedure of Schwefel (1981) can be used to construct any valid correlated multinormal random vector. However, numerical results indicate that the convergence of the approximation is not yet satisfactory. The main reason might be found in the fact that the step-size adaptation process affects the angle adaptation process in a disruptive way.

The methodology of derandomized ES (Ostermeier et al., 1994, 1995*a*, 1995*b*) is utilized in this paper to tune conventional PID and gain scheduling PID control algorithms. This algorithm deals an ES based on the derandomized scheme of mutative step-size control. The adaptation concept uses information accumulated from the preceding generations with an exponential fading of old information instead of using information from the current generation only.

The implemented $(1 + \lambda)$-ES tries to derandomize the adaptation process by exploiting information gathered in preceding iterations. Derandomized adaptation usually takes place without mutation of the strategy parameters themselves. It uses selected points (more specifically: selected mutation steps) in the object parameter space for strategy parameter adjustment. Derandomized $(1 + \lambda)$-ES has a start point chosen randomly with uniform probability distribution, and it uses two operators: selection and mutation. The recombination operator is not utilized because $\mu = 1$ is adopted. The selection (adaptation) operator is completely deterministic; it just chooses the most fit μ individuals $(1 \le \mu < \lambda)$ out of the set of λ offspring individuals. The individual $(\mu = 1)$ with the highest fitness advances to the next generation, that is,

$$\vec{K}_{\mu}(t+1) = \vec{K}_{\lambda sel}(t) \tag{1}$$

https://doi.org/10.1017/S0890060499135042 Published online by Cambridge University Press

where $\vec{K}_{\mu}(t)$ is the parameter vector of generation t and *sel* marks the quantities of the *sel*ected offspring of generation t. Mutation is the main operator and introduces random modification into the population. The implemented algorithm realizes mutation ellipsoids in the direction of the selected offspring. The creation of λ offspring is given by (Ostermeier et al., 1994, 1995*a*):

$$\vec{K}_{\lambda i}(t) = \vec{K}_{\mu}(t) + \delta(t)\cdot\vec{\delta}_{scal}(t)\cdot\vec{Z}_i, \tag{2}$$

where $\delta(t)$ is the general step-size of generation t, $\vec{\delta}_{scal}(t)$ is the individual step-sizes of generation t ($\vec{\delta}_{scal}(0) = (1,\ldots,1)$), $\vec{Z} = (z_1,\ldots,z_n)$, with $z_i(0,1)$ normally distributed, n is the number of parameters to be optimized. The equation of accumulation of selected mutations is given by:

$$\vec{\vec{Z}}(t) = (1-c)\vec{\vec{Z}}(t-1) + c\vec{Z}_{sel}, \quad \vec{\vec{Z}}(0) = \vec{0}, \tag{3}$$

where $c = \sqrt{1/n}$ is the constant factor that determine how fast the contribution of former generations declines, that is, the weighting of the fast generation and the lifespan of the information of preceding generations, respectively. The adaptation scheme of the general step-size uses the convergence of the χ-distribution, $|\vec{Z}| = \sqrt{\sum(z_i)^2} \xrightarrow[N\to\infty]{} N(\sqrt{n},0.5)$, and it is given by:

$$\delta(t+1) = \delta(t)\cdot\left(\exp\left(\frac{|\vec{\vec{Z}}(t)|}{\sqrt{n}\cdot\sqrt{c/(2-c)}} - 1 + \frac{1}{5n}\right)\right)^{\beta} \tag{4}$$

$$\vec{\delta}_{scal}(t+1) = \vec{\delta}_{scal}(t)\cdot\left(\exp\left(\frac{|\vec{\vec{Z}}(t)|}{\sqrt{c/(2-c)}} + 0.35\right)\right)^{\beta scal}, \tag{5}$$

where $\beta = \sqrt{1/n}$ and $\beta_{scal} = 1/n$. The exponents β and β_{scal} are relevant factors for adaptation speed and precision. Sensible values are in the range (0,1). Small values facilitate a precise but time-consuming adaptation and vice versa. The given values yield an adequate compromise. The factor $\sqrt{c/(2-c)}$ normalizes the mean variances of the resulting distributions to one (when no selection takes place). It results from the geometric series of the mean variations of the added mutations:

$$\lim_{m\to\infty}\sqrt{c^2 + [c\cdot(1-c)]^2 + [c\cdot(1-c)^2]^2 + [c\cdot(1-c)^3]^2 + \ldots + [c\cdot(1-c)^m]^2} = \sqrt{c/(2-c)}. \tag{6}$$

The derandomized mutative step-size control in ESs enables a reliable step-size adaptation and the utilization of accumulated information decreases the locality of the adaptation process. The improvement achieved arises from the implicit use of correlations of the selected mutations in the generation sequence. Consequently, the step-sizes are adapted to such values that successive selected mutations tends to be orthogonal on average. This seems to be characteristic for optimal step-sizes in general (Ostermeier, 1994, 1995*a*).

3. EVOLUTIONARY OPTIMIZATION BY DERANDOMIZED (1 + λ)-ES

The optimization objective by derandomized $(1+\lambda)$-ES is the adjustment of three parameters (K_p, T_i, and T_d). The standard form of classic digital PID control is given by:

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2), \tag{7}$$

where the parameters q_0, q_1, and q_2 are calculated by

$$q_0 = K_p\left(1 + \frac{T_s}{2T_i} + \frac{T_d}{T_s}\right) \tag{8}$$

$$q_1 = -K_p\left(1 + \frac{2T_d}{T_s} - \frac{T_s}{2T_i}\right) \tag{9}$$

$$q_2 = K_p\frac{T_d}{T_s}, \tag{10}$$

and where $u(k) \in [0.0; 5.0]$ volts is the control action, T_s is the sampling time, and $e(k)$ is the error given by the difference between the output and desired setpoint, $y(k) \in [0.0; 5.0]$ volts is the process output.

GS-PID control design is different from conventional PID because the former is adequate for treatment of/with different dynamics for each operation range, that is, the processes can be governed by multiple models. The adopted design of GS-PID control consists in the tuning of three parameters of a PID controller, including one for each setpoint change, y_r, with total number of nine parameters. The optimization procedure is tested for three setpoint changes, as shown in Section 5.

The performance index to be minimized, $J(u,e)$, adopted for PID and GS-PID controller tuning, denotes dynamic aspects of error terms using steady-state, rise time, overshoot and relative stability, and is given by

https://doi.org/10.1017/S0890060499135042 Published online by Cambridge University Press

$$J(u,e) = \frac{\sum_{k=1}^{N} \{|e(k)| + w|\Delta e(k)|\} i}{N}, \tag{11}$$

where $\Delta e(k) = e(k) - e(k-1)$, N is the number of process samples, w is set to 1, i is equal to k, except when any change of setpoint occurs, in which case, $i = 1$ and subsequently should be increased in each iteration as k increases. The utilized functional consists of the L_1 norm of error weighted in time (Li et al., 1996). Consequently, the fitness, $F[J(u,e)]$ to be maximized by derandomized $(1 + \lambda)$-ES is given by

$$F[J(u,e)] = \frac{1}{1 + J(e,u)} \tag{12}$$

Figure 1 presents the configuration for optimization of PID and GS-PID controllers by a derandomized $(1+\lambda)$-ES.

The tuning procedure of PID and GS-PID controllers when applied to the fan and plate process follows the steps:

1. determination of the controller configuration (range for parameter search), process (number of samples, sampling time, setpoint) and derandomized $(1+\lambda)$-ES to be utilized in the practical experiment. Design parameters utilized in this work are presented in Table 1;
2. initialization with uniform distribution of parameters of the population of the PID (or GS-PID) controller, that is, a set of controllers that conform the initial population is generated randomly;
3. application of control law with the parameters of each population individual and sampling time;
4. storage of input, output, and error data of the fan and plate process for the individual under analysis;
5. evaluation of fitness of population's individuals;
6. application of deterministic selection operator for choice of μ parent for new generation;
7. application of mutation operator with self-adaptation mechanisms and generation of new population of λ individuals;
8. repeating of steps (3) to (7) until a stopping condition is satisfied—in this case, the number of generations should be equal to 10.

Table 1. *Design parameters of experiment*

Parameters	Values
T_s	200 ms
μ	1
λ	5
Number of generations	10
Experiments realized for design of each controller	5
Range of parameter search	$K_p = [0.04; 0.2]$, $T_i = [0.1; 1.0]$, $T_d = [0.0; 1.0]$
Samples	500 samples (that demand 1 min and 40 s for each fitness evaluation)

4. FAN AND PLATE NONLINEAR PROCESS

The fan and plate experiment is presented in several universities for teaching and research activities. The fan-and-plate process has complex features and motivates the design of intelligent techniques. The fan and plate control system is composed of a fan driven by a dc motor, a 50-cm-long air duct with a funneling characteristic and having on its left extremity a small rectangular plate (Fig. 2).

The 24-V dc motor is driven by an actuator circuit whose input is compatible with the D/A converter output. The angular deflection of the plate is measured by a photoconductive cell (light from LED that passes through a disk painted with varying shades, from white to black, whose incidence

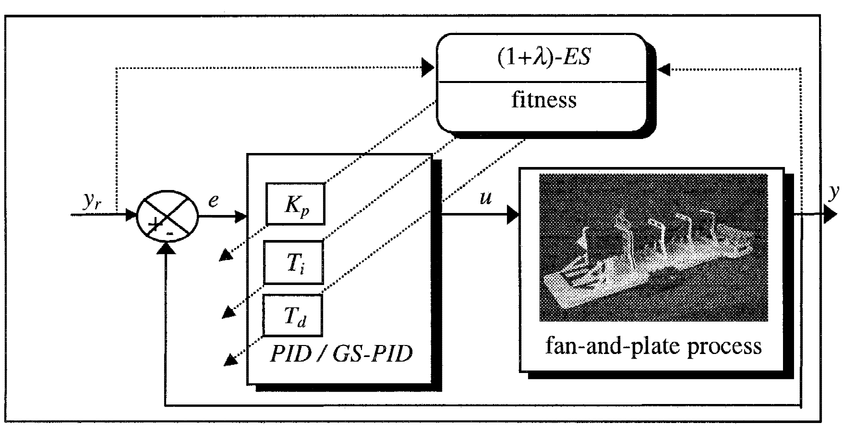
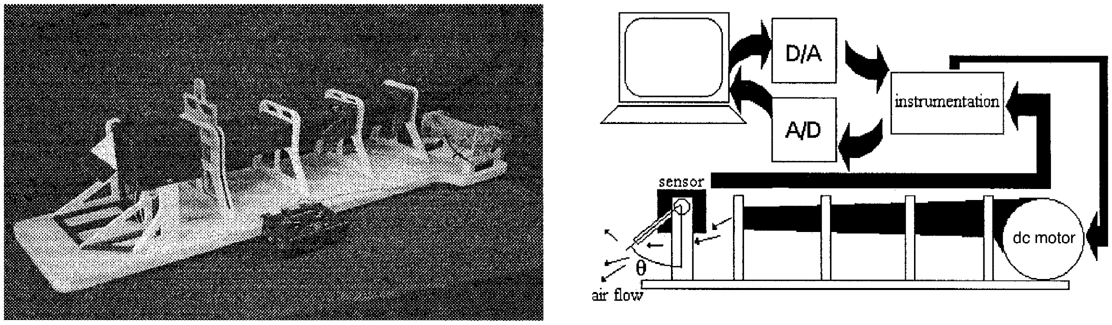


Fig. 1. PID and GS-PID controllers with tuning by a derandomized $(1 + \lambda)$-ES.

https://doi.org/10.1017/S0890060499135042 Published online by Cambridge University Press



(a) photograph (b) physical setup

Fig. 2. Physical setup of the fan and plate.

on a photo element will cause it to change its conductive properties) and connected to the measurement circuit.

The control problem is to regulate the angular deflection of the plate (controlled variable) actuating on the input voltage of the dc motor (manipulated variable). The distance between fan and plate can be changed and defines an important parameter of the system. The prototype, containing nonminimum phase, dead-time, resonant and turbulent disturbance behavior, can serve as tangible evidence of the usefulness of intelligent control techniques in difficult situations.

5. EXPERIMENTAL RESULTS

The $(1 + \lambda)$-ES is implemented in the C language and utilizes the 80486 Intel processor with clock of 33 MHz. The total time for each complete cycle of controller optimization is 1 hour and 50 min, plus the time of processing of ES and the time required for the process to start in the same position for all fitness evaluations.

Next, the experimental results of servo behavior—tuning phase—with three changes of setpoint to $y_{r1} = 2.0$ (samples 0 to 150), $y_{r2} = 3.5$ (samples 151 to 300), and $y_{r3} = 2.5$ (samples 301 to 500) are presented. In the servo/regulatory behavior analysis of the controller tuning—test phase—the parameters obtained in the tuning phase by the $(1 + \lambda)$-ES are maintained constant and a bias of + 0.5 V is added at samples 100 and 400, and removed at samples 150 and 450, respectively.

Table 2 presents the results of tuning the PID and GS-PID controllers with a derandomized $(1 + \lambda)$-ES. Results are obtained after 10 generations of evolution in 5 experiments. These data are associated with the experiments for three setpoint changes. Figures 3–6 present the best results in tuning and test phases of *PID* and GS-PID controllers.

The most adequate gains for controller configurations are: $K_p = 0.093$; $T_i = 0.167$, and $T_d = 0.143$—for the PID controller—and K_{p1}=0.168; $T_{i1} = 0.269$; $T_{d1} = 0.0$ ($y_{r1} = 2.0$); $K_{p2} = 0.040$; $T_{i2} = 0.128$, $T_{d2} = 0.002$ ($y_{r2} = 3.5$), and $K_{p3} = 0.073$; $T_{i3} = 0.714$; and $T_{d3} = 0.038$ ($y_{r3} = 2.5$)—for the GS-PID controller.

Gains for the conventional tuning by Ziegler–Nichols method (PID-ZN), based on critical gain and oscillation period (Åström & Hägglund, 1988) are $K_p = 0.153$; $T_i = 0.148$; and $T_d = 0.108$. The servo behavior is shown in Figure 7. It

Table 2. *Comparative study by derandomized $(1 + \lambda)$-ES in the tuning phase of PID and GS-PID controllers*

Experimental Data Experiment/Functional	PID $J(u,e)$	GS-PID $J(u,e)$
Number 1	9.659	10.176
Number 2	9.064	8.417
Number 3	8.529	9.674
Number 4	9.136	9.389
Number 5	8.683	8.283
Average	9.014	9.188
Standard deviation	0.441	0.816
Best value	8.529	8.283
Worst value	9.659	10.176

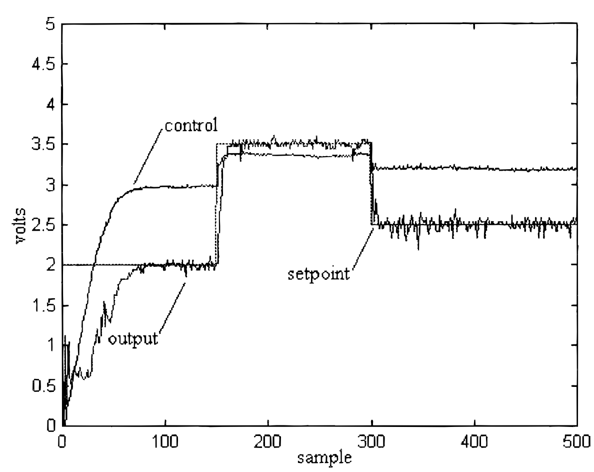


Fig. 3. PID control of fan and plate process—tuning phase—with a derandomized $(1 + \lambda)$-ES.

https://doi.org/10.1017/S0890060499135042 Published online by Cambridge University Press

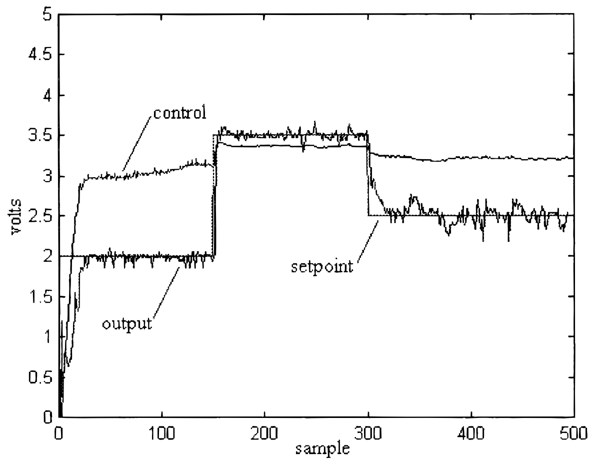


Fig. 4. GS-PID control of fan and plate process—tuning phase—with a derandomized $(1 + \lambda)$-ES.

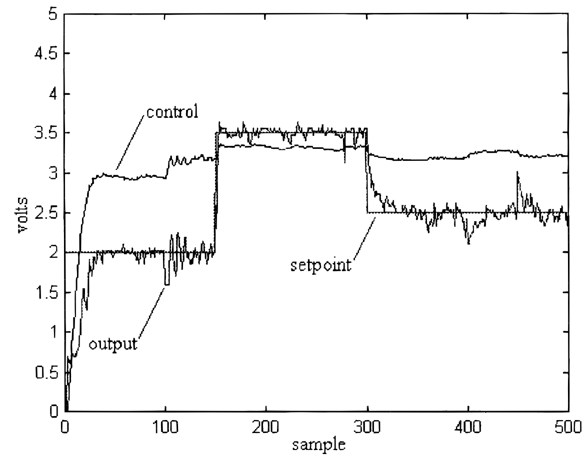


Fig. 6. GS-PID control of fan and plate process—test phase—with a derandomized $(1 + \lambda)$-ES.

can be observed that the adjusted gains do not provide an adequate tuning for PID if compared with evolutionary tuning for PID and GS-PID controllers, as presented in Figures 3–6. The obtained functional by PID-ZN is $J(e,u) = 61.310$. The complexities of the fan-and-plate process are affecting the performance of PID-ZN technique in the PID tuning phase.

Figure 8 shows the gain evolution by derandomized $(1 + \lambda)$-ES is the controller turning PID(3) and GS-PID(5), according to Table 2.

According to Table 2 and Figures 3–6, it is possible to observe that the derandomized $(1 + \lambda)$-ES is adequate to tune PID and GS-PID controllers, as evidenced by the fast convergence confirmed by average and standard deviation values in relation to the adopted configuration of number of tuning parameters, population size and small number of generations for evolutionary optimization cycle.

6. CONCLUSION

The derandomized $(1 + \lambda)$-ES procedure was successful when applied to tune PID and GS-PID controllers without the necessity of *a priori* knowledge of the fan and plate process model. In this practical application, the derandomized $(1 + \lambda)$-ES was able to converge toward adequate parameters, although there was no knowledge of fan and plate process parameters, such as process order, nonlinearities, noise properties, and other factors.

Among the relevant aspects considered in this work are: (1) application of the evolutionary methodology in control-

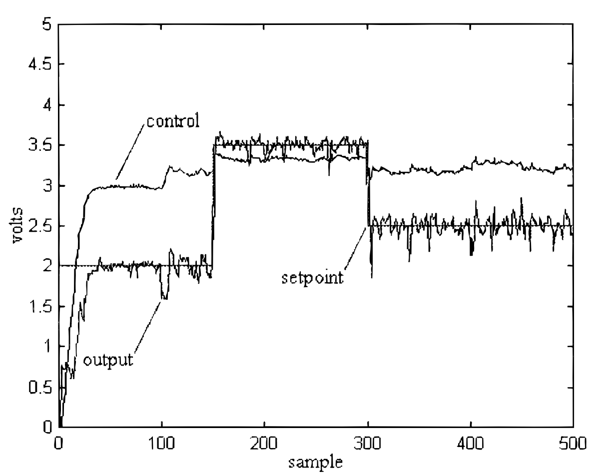


Fig. 5. PID control of fan and plate process—test phase—with a derandomized $(1 + \lambda)$-ES.

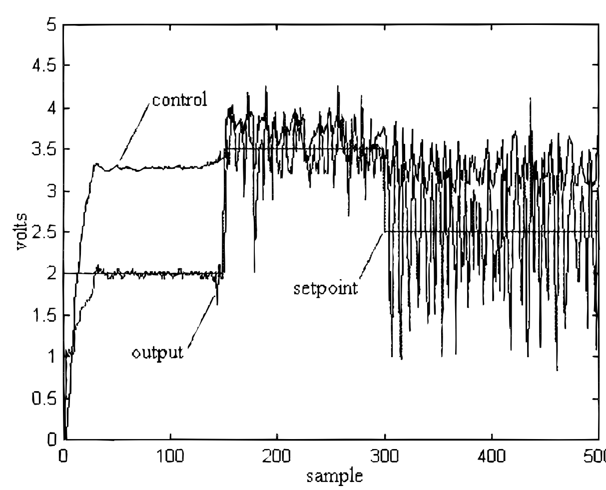


Fig. 7. PID-ZN tuning for setup changes (tuning phase).

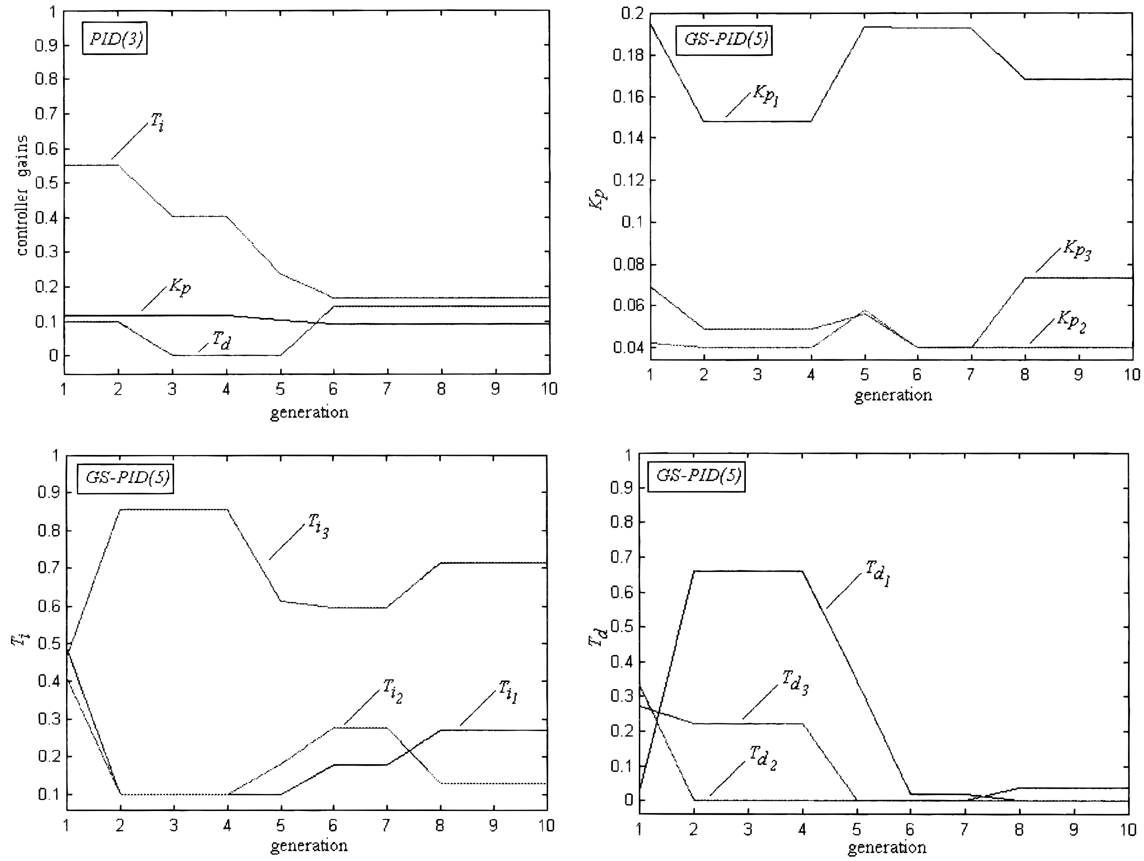


Fig. 8. Evolution of PID and GS-PID controllers gains.

ler tuning for a practical process; (2) possibility of utilization of these powerful tools of *EC* in an industrial process, and (3) features of robustness and fast convergence of the derandomized $(1 + \lambda)$-ES.

Servo and regulatory results are motivating in considering the proposed procedure as a suitable tool for controller design, and also stimulates investigating the possibility of further application to design and development in the case of practical multivariable systems.

As future works, have been evaluated to deal with the comparative study of ES and other intelligent paradigms with adaptive control methodologies. Therefore, a robust and efficient design structure for application in system identification, modeling, and process control in laboratory and industrial environments may be achievable.

ACKNOWLEDGMENTS

The authors would like to acknowledge the anonymous reviewers for many insightful remarks that contributed to the clarity of this paper.

REFERENCES

Åström, K.J., & Hägglund, T. (1988). *Automatic tuning of PID controllers*. Instrument Society of America, New York.

Åström, K.J., & Wittenmark, B. (1989). *Adaptive control*. Addison-Wesley Publishing Company, Massachusetts.

Bäck, T., Fogel, D.B., & Michalewicz, Z. (eds.) (1997). *Handbook of evolutionary computation*. Oxford University Press, New York.

Bäck, T., Hoffmeister, F., &. Schwefel, H.-P. (1991). A survey of evolution strategies. *Proc. 4th Int. Conf. Genet. Algorithms and their Applications*, pp. 2–9. Morgan Kaufmann, San Francisco, California.

Bäck, T., & Schwefel, H.P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Comput., 1(1)*, 1–23.

Beyer, H.-G. (1996). Toward a theory of evolution strategies: Self-adaptation. *Evolutionary Comput., 3(3)*, 311–347.

Chen, B.-S., Cheng, Y.-M., & Lee, C.-H. (1995). A genetic approach to mixed H_2/H_∞ optimal PID control. *IEEE Control Sys., 15(5)*, 51–60.

Chipperfield, A., & Fleming, P. (1996). Evolutionary algorithms for control engineering. *13th World Congress of IFAC*, 181–186.

Coelho, L.S., Almeida, O.M., Simas, H., & Coelho, A.A.R. (1998). Intelligent and self-tuning PID controllers: Methods and application. *XII Brazilian Automatic Control Conference*, 375–380.

Coelho, L.S., & Coelho, A.A.R. (1998). Genetic algorithms and evolution strategies applied in identification and control: Case study. In *Soft Computing in Engineering Design and Manufacturing*, (Chawdhry, P.K., et al., Eds.), pp. 430–438. Springer-Verlag, London.

Davidor, Y., & Schwefel, H.-P. (1992). An introduction to adaptive optimization algorithms based on principles of natural evolution. In *Dynamic, Genetic, and Chaotic Programming: The Sixth Generation*, (Soucek B. and the IRIS Group), pp. 183–202. John Wiley & Sons, Inc., New York.

Fogel, D.B. (1995). *Evolutionary computation: Toward a new philosophy of machine intelligence*. IEEE Press, Piscataway, New Jersey.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, Massachusetts.

Hang, C.C., Åström, K.J., & Ho, W.K. (1991). Refinements on the Ziegler–Nichols tuning formula. *IEE Proc.-D, 138(2)*, 111–118.

Hoffmeister, F., & Bäck, T. (1992). Genetic self-learning. *1st European Conference on Artificial Life*, 227–235.
Huang P.-Y., & Chen, Y.-Y. (1997). Design of PID controller for precision positioning table using genetic algorithms. *Proc. 36th IEEE Conf. Decision and Control*, 2513–2514.
Hwang, W.R., & Thompson, W.E. (1993). An intelligent controller design based on genetic algorithms. *Proc. 32nd Conf. IEEE Decision and Control*, 1266–1267.
Li, Y., Ng, K.C., Smith, D.J.M., Gray, G.J., & Sharman, K.C. (1996). Genetic algorithm automated approach to the design of sliding mode control systems. *Int. J. Control*, *63(4)*, 721–739.
Lo Bianco, C.G., & Piazzi, A. (1996). A hybrid genetic/interval algorithm for semi-infinite optimization. *Proc. 35th IEEE Conf. Decision and Control*, 2136–2138.
Malki, H.A., Misir, D., Feigenspan, D., & Chen, D. (1997). Fuzzy PID control of a flexible-joint robot arm with uncertainties form time-varying loads. *IEEE Trans. Control Sys. Technol. 5(3)*, 371–378.
Man, K.F, Tang, K.S., & Kwong, S. (1996). Genetic algorithms: Concepts and applications. *IEEE Transact. Ind. Electronics*, *43(5)*, 519–534.
Miranda, V., Srinivasan, D., & Proença, L.M. (1998). Evolutionary computation in power systems. *Electrical Power & Energy Sys.*, *20(2)*, 89–98.
Ostermeier, A., Gawelczyk, A., & Hansen, N. (1994). Step-size adaptation on non-local use of selection information. *3rd Int. Conf. Parallel Problem Solving from Nature*, 189–198.
Ostermeier, A., Gawelczyk, A., & Hansen, N. (1995*a)*. A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation 2(4)*, 369–380.
Ostermeier, A., Gawelczyk, A., & Hansen, N. (1995*b)*. A derandomized approach to self-adaptation of evolution strategies. *Evolutionary Computation 2(4)*, 369–380.
Rechenberg, I. (1965). *Cybernetic solution path of an experimental problem* (Library Translation 1122), Royal Aircraft Establishment, Farnborough, U.K.
Rechenberg, I. (1973). Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution. Frommann-Holzboog, Stuttgart.
Rudolph, G. (1992). On correlated mutations in evolution strategies. *Proc. 2nd Int. Conf. Parallel Solving from Nature*, 105–114.
Schwefel, H.-P. (1981). *Numerical optimization of computer models*. John Wiley & Sons, New York.
Takahashi, R.H.C., Peres, P.L.D., & Ferreira. P.A.V. (1997). Multiobjective H_2/H_∞ guaranteed cost PID design. *IEEE Control Sys.*, *17(5)*, 37–47.
VanDoren, V.J. (1998). Advanced control software goes beyond PID. *Control Eng. Int.*, *January*, 57–60.
Wang, P., & Kwok, D.P. (1993). Optimal design of PID process controllers based on genetic algorithms. *12th World Congress of IFAC*, 261–265.
Willis, M.J., & Montague, G.A. (1993). Auto-tuning PI(D) controllers with artificial neural networks. *12th World Congress of IFAC*, 61–64.
Yamamoto, T., & Shah, S.L. (1998). Design and experimental evaluation of a multivariable self-tuning PID controller. *Proc. IEEE International Conf. Control Applications*, 1230–1234.
Yusof, R., & Omatu, S. (1993). A multivariable self-tuning PID controller. *Int. J. Control*, *57(6)*, 1387–1403.
Zadeh, L.A. (1994). Soft computing and fuzzy logic. *IEEE Software*, *11(6)*, 48–56.
Ziegler, J.G., & Nichols, N.B. (1942). Optimum settings for automatic controllers. *Transact. ASME*, *65*, 433–444.

Antonio Augusto Rodrigues Coelho is currently an Associate Professor in the Department of Automation and Systems at Federal University of Santa Catarina, Brazil. He received the B.S. degree (in electronic engineering) from the Federal University of Pará, Brazil, in 1979, the M.S. degree (in electrical engineering) from the Federal University of Santa Catarina, in 1983, and the doctoral degree (in electrical engineering) from University of Campinas, Brazil, in 1991. His research interests include identification, predictive control, control education, and intelligent control.

Leandro dos Santos Coelho received the B.S. degree (in computer science) from the Federal University of Santa Maria, Brazil, in 1994, and the M.S. degree (in computer science) from the Federal University of Santa Catarina, in 1997. He is currently working toward the doctoral degree in electrical engineering at Federal University of Santa Catarina. His research interests include identification, predictive control, neural networks, evolutionary computation, and fuzzy systems.