Two-Degrees-of-Freedom Robust PID Controllers Tuning Via a Multiobjective Genetic Algorithm

José Rubén Lagunas-Jiménez[1], Víctor Moo-Yam[1], and Benjamín Ortíz-Moctezuma[2]

[1] Universidad Autónoma de Campeche, Campeche,
Mexico

[2] Universidad Politécnica de Victoria, Tamaulipas,
Mexico

{jrlaguna, victmmoo}@uacam.mx, mortizm@upv.edul.mx

Abstract. In this paper, a design methodology for a proportional integral derivative (PID) control design is presented by means of the statement of a multiobjective optimization problem (MOP). Two-degrees-of-freedom controller (PID-ISA) is used. The objective functions are deployed considering a set point response, load disturbances and robustness to model uncertainty as its components. The time constant of measurement noise filter is a component of the vector of decision variables. The optimization problem is solved by means of a genetic algorithm.

Keywords. Multiobjective optimization, two-degrees-of-freedom PID controller, robustness, uncertainty, genetic algorithm.

Sintonización de controladores PID robustos de dos grados de libertad mediante un algoritmo genético multiobjetivo

Resumen. En este artículos e presenta una metodología de diseño de controladores PID (Proporcional, Integral y Derivativo), de dos grados de libertad mediante el planteamiento de un problema de optimización multiobjetivo. Las funciones objetivo propuestas consideran entre otros: respuesta de referencia al escalón, perturbación de carga y robustez ante incertidumbre en el modelado. También se incluye un filtro para minimizar el ruido de medición y la constante de tiempo se incluye en el vector de variables de decisión. El problema de optimización se resuelve con un algoritmo genético.

Palabras clave. Optimización multiobjetivo, controlador PID de dos grados de libertad, robustez, incertidumbre, algoritmo genético.

1 Introduction

Proportional-Integral-Derivative (PID) controllers are widely used in many control systems. In process control, more than 95% of the control loops are of PID type [2]. Since Ziegler and Nichols [28] proposed their empirical method to tune PID controllers to date, many relevant methods to improve the tuning of PID controllers have been reported in the control literature, one of them is a tutorial written by Hang *et al.* [14]. In the case of PID controller design, multiobjective optimization makes sense when designing two-degrees-of-freedom PID controllers due to a wide availability of choices concerning such controller structure, whose main advantage is the possibility of decoupling set-point and load-disturbance signals [2, 3, 13, 25, 27]. This feature can be of advantage by posing two different objective functions depending on the set of controller parameters and on the other hand, setting the sensitivity function as the objective function leads to many useful physical interpretations for feedback systems. Moreover, problem formulation by using three objective functions makes it possible to obtain distinct controllers, all of them Pareto optimal but showing different performance concerning design objectives required in terms of control specifications.

The multiobjective optimization problem makes sense when the performance indices involved are in conflict; otherwise the multiobjective optimization problem can be regarded as a monoobjective one, since one single solution is enough to simultaneously minimize or maximize

the objective functions. Very often, in real-world problems the objectives enter into conflict to varying extents. The complexity of some multiobjective optimization problems (e.g., very large search spaces, uncertainty, noise, disjoint Pareto curves, etc.), encourage to use some evolution strategy such as a genetic algorithm [8, 10].

Genetic algorithms have been successfully applied to multiobjective optimization. One of the first reported applications is due to Fonseca and Fleming (1988), who reportedly applied a multiobjective genetic algorithm (MOGA) for controlling a gas turbine [12]. In Herrero's doctoral thesis (2000), an algorithm termed "multiobjective robust control design" (MRCD) was proposed for the design of robust PID controllers, considering parametric uncertainties [15].

For the tuning of the PID-ISA controllers, the feedback control problem appears as a multiobjective optimization problem, that of a set of functions, where the controller parameters are included. The design specifications can be formulated as objective functions, subject to certain constraints, expressed in terms of different norms involving the closed-loop transfer functions of the considered control system. The NSGA-II (non-dominated sorting genetic algorithm-II), used in this work, gives a set of solutions, all good in the sense of Pareto non-dominance (Pareto optimal set), where each of the solutions contains the PID controller parameters. From this set, the person who solves an optimization problem can select some solutions according to a given suitable determined criterion. The NSGA-II has been used in many control applications as shown in [8, 18, 23] and most recently in [4, 6, 26].

The method proposed in this paper takes into account model uncertainty, for the wide spectrum of plants used for testing the design of two-degrees-of-freedom PID-ISA controllers. Due to the fact that modeled dynamics is unknown (unstructured uncertainty), the uncertainty is modeled by assuming it to be bounded by a frequency dependent weighting function, keeping in mind that uncertainty affects the system mostly at frequency higher than cutoff frequency of closed-loop system. The effect of the sensitivity function values of the closed-loop system, as a measure of robustness against possible variations

in the parameters of the plant, is also considered in the proposed PID controller design.

The rest of the paper is organized as follows. In Section 2 the basic definitions of multiobjective optimization as well as a genetic algorithm (NSGA-II) procedure are given. In Section 3, some basic principles of modeling uncertainty for control systems are presented in order to design robust PID controllers for systems subject to unstructured uncertainties. Section 3 also poses the control problem and the robust PID controller design problems. Section 4 presents the results of evaluating the performance of the designed PID controllers. For each of the proposed process models, two different solutions are shown to emphasize the concept of multiobjective optimization. Conclusions are presented in Section 5.

2 Basic Concepts

As a consequence of applying the multiobjective genetic algorithm to the optimization problem, the algorithm outputs a set of ideal solutions called Pareto optimal set; from this set, the person who solves an optimization problem (the user) can select some solutions in accordance with his/her preferences, since in the majority of practical problems it is not possible to find a unique solution that either minimizes or maximizes all objectives simultaneously.

Next, the basic definitions related to the ideal solutions or non-dominated solutions (in the Pareto sense) are presented. Solutions are non-dominated if no better solutions exist, considering all the objective functions (see [10] as an example).

Definition 1 (General MOP). Find the vector $\bar{x}^* = \left[x_1^*, x_2^*, \dots, x_n^*\right]^T$, which satisfies the *m* inequality constraints

$$g_i(\bar{x}) \geq 0 \quad i = 1,2,\dots,m ,\qquad (1)$$

the *p* equality constraints

$$h_i(\bar{x}) \geq 0 \quad i = 1,2,\dots,p ,\qquad (2)$$

and optimizes the vector function

$$\vec{f}(\vec{x}) = \left[f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x}) \right]^T, \qquad (3)$$

where $\vec{x} = [x_1, x_2, \ldots x_n]^T$ is the vector of decision variables.

Definition 2 (Pareto Dominance). A vector $\vec{u} = (u_1, \ldots u_k)$ is said to dominate $\vec{v} = (v_1, \ldots, v_k)$ (denoted by $\vec{u} \preceq \vec{v}$) iff **u** is partially less than **v**, i.e.,

$$\forall i \in \{1,2,\ldots,k\}, u_i \leq v_i \wedge \exists i \in \{1,2,\ldots,k\} : u_i < v_i.$$

Definition 3 (Pareto Optimality). A point $\vec{x}^* \in \Omega$ is Pareto optimal, if for every $\vec{x} \in \Omega$ and $I = \{1,2,\ldots k\}$ either

$$\forall_{i \in I} (f_i(\vec{x}) = f_i(\vec{x}^*)), \qquad (4)$$

or there is at least one $i \in I$ such that

$$f_i(\vec{x}) > f_i(\vec{x}^*), \qquad (5)$$

where Ω is the feasible region. In other words, this definition says that $\vec{x}^*$ is Pareto optimal if there exist no feasible vector $\vec{x}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion.

Definition 4 (Pareto Optimal Set). For a given MOP $\vec{f}(x)$, the Pareto optimal set (P^*) is defined as

$$P^* := \{x \in \Omega | \neg \exists x' \in \Omega \vec{f}(x') \preceq \vec{f}(x)\}. \qquad (6)$$

Definition 5 (Pareto Front). For a given MOP $\vec{f}(x)$ and Pareto optimal set (P^*), the Pareto front (FP^*) is defined as

$$PF^* := \{\vec{u} = \vec{f} = (f_1(x), \ldots, f_k(x)) \mid x \in P^*\}. \qquad (7)$$

In contrast to simple genetic algorithms which look for the unique solution, the multiobjective genetic algorithm tries to find as many elements of the Pareto set as possible. For the case of the NSGA-II, this one is provided with operators who allow it to know the level of not-dominance of every solution as well as the grade of closeness
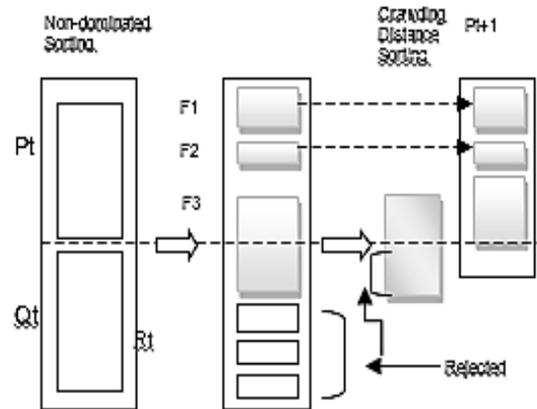

Fig. 1. NSGA-II procedure

with other solutions; which allows it to explore widely inside the feasible region.

In a brief form, the functioning of the MOGA NSGA-II can be described by the following steps.

Fast Non-dominated Sort. A very efficient procedure is used to arrange the solutions in fronts (non-dominated arranging) in accordance with their aptitude values. This is achieved by creating two entities for each of the solutions. A domination counts n_p, the number of solutions which dominates the solution p and a set (S_p) which contains the solutions that are dominated for p. The solutions of the first front have the higher status of not-dominance in the Pareto sense.

Diversity Preservation. This is achieved by means of calculation of the crowding degree or closeness for each of the solutions inside the population. This quantity is obtained by calculating the average distance of two points on either side of a particular solution along each of the objectives. This quantity serves as an estimate of the cuboid perimeter formed by using the nearest neighbors as the vertices. There is also an operator called crowded comparison ($\prec_n$), which guides the genetic algorithm towards the Pareto optimal front in accordance with the following criterion:

$$i \prec_n j \ \text{if}(i_{rank} < j_{rank})$$
$$\text{or}(i_{rank} = j_{rank})$$
$$\text{and}(i_{distance} > j_{distance})$$

Algorithm 1. NSGA-II algorithm

1: Procedure NASG-II (N, g, $fx(\boldsymbol{xk})$) where N members evolved g generations to solve $fk(\boldsymbol{x})$
2: Initialize Population $\boldsymbol{P}$
3: Generate random population – size N
4: Evaluate Objective Values
5: Assign Rank (level) based on Pareto dominance – *sort*
6: Generate Child Population
7: Binary Tournament Selection
8: Recombination and Mutation
9: **for** i=1 to g **do**
10: **for** each Parent and child in Population **do**
11: Assign Rank (level) based on Pareto – sort
12: Generate sets of non-dominated vectors Along PF$_{known}$
13: Loop (inside) by adding solutions to next generation starting from the *first* front until N individuals found determine crowding distance between points on each front
14: **end for**
15: Select points (elitist) on the lower front (with lower rank) and outside a crowding distance
16: Create next generation
17: Binary Tournament Selection
18: Recombination and Mutation
19: **end for**
20: **end Procedure.**

In accordance with the previous criterion, between two non-dominated solutions, we prefer the solution with the better rank. Otherwise, if both solutions belong to the same front, then we prefer the solution that is located in a lesser crowded region.

Initial Loop. Initially, a random parent population (P_o) of size N is created. Later this one is ordained using the procedure of non-dominated arranging. Then the usual binary tournament selection, recombination and mutation operators are used to create a new population (Q_0), of size N.

Main Loop. The NSGA-II procedure can be explained by describing the i-th generation just as it is showed in Fig. 1. The procedure begins with the combination of P_t and Q_t forming a new population called R_t, then the population R_t is sorted using the non-domination criterion. Since all previous and current population members are included in R_t, elitism is ensured [17]. The population R_t has a size of $2N$; later, different fronts of non-dominated solutions are created, being F_1 the front that contains the better rank solutions. Fig. 1 shows that during the process of forming a new population P_{t+1}, the algorithm takes all members of the fronts F_1 and F_2 and some elements of the front F_3; this is because N solutions are needed exactly for the new population P_{t+1} to find exactly N solutions. The last front is ordained, which for this description is the number 3, arranging the solutions in descending order by means of crowded comparison ($\prec_n$) and selecting the best solutions needed to fill all population slots. After having the population P_{t+1}, the genetic operators of selection, crossing and mutation are used to create a new population Q_{t+1} of size N. Finally it is mentioned that in the selection process, the crowded comparison operator is used. The NSGA-II algorithm is shown in Algorithm 1, [10].

The selected algorithm (NSGA-II) turns out to be of complexity $O(M\,N^2)$, where M is the number of objectives and N is the population size.

3 PID-ISA Controller

The Proportional-Integral-Derivative (PID) controller proposed in this work is a two-degrees-of-freedom controller. PID-ISA controller (Instrument Society of America), which contains seven parameters, is tuned by means of the multiobjective genetic algorithm NSGA-II. The control scheme proposed in this work is shown in Fig. 2 where

   r denotes the reference input signal,
   e denotes the error signal,
   $\tilde{y}$ denotes the filtered output signal,
   u denotes the control signal,
   l denotes the load disturbance signal,
   d denotes the noise signal,
   y denotes the output signal,
   G(s) denotes a Linear Time-Invariant (LTI) Single-Input Single-Output (SISO) plant,
   PID(s$)$ denotes the PID-ISA Controller.

The filter F(s) is used to reduce the noise effect of high frequency in the output signal, where T_f is the filter time constant:
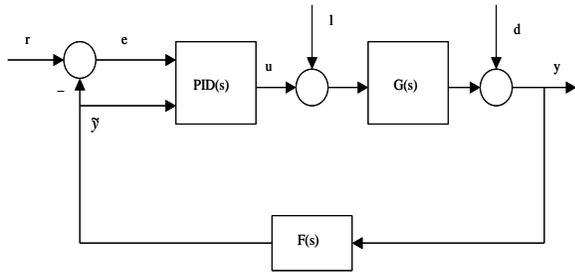
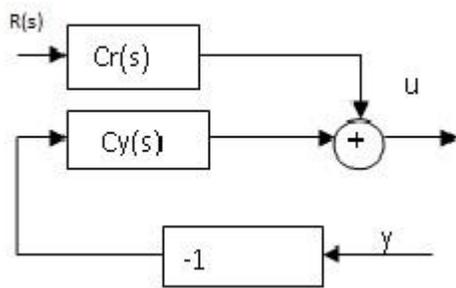**Fig. 2**. Description of PID-ISA controller

Fig. 3. Equivalent diagram for the PID-ISA controller

$$F(s) = \frac{1}{1 + sT_f}, \qquad (8)$$

and the PID-ISA model controller:

$$u(s) = k\left(e_p(s) + \frac{1}{T_i s}e(s) + \frac{T_d s}{1 + T_d s/N}e_d(s)\right)$$

$$e(s) = r(s) - \tilde{y}(s) \qquad (9)$$

$$e_p(s) = br(s) - \tilde{y}(s) = be(s) + (b-1)\tilde{y}(s)$$

$$e_d(s) = cr(s) - \tilde{y}(s) = ce(s) + (c-1)\tilde{y}(s).$$

Also, the PID-ISA controller can be represented as in Fig. 3:

$$C_r(s) = k\left(b + \frac{1}{T_i s} + \frac{cT_d s}{1 + T_d s/N}\right), \qquad (10)$$

$$C_y(s) = k\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + T_d s/N}\right). \qquad (11)$$

In (10) and (11), k, T_i and T_d correspond to the controller gain, the integral time and the derivative time, respectively. The parameters b and c are the weightings that influence the set point response without altering the response of the controller to the load disturbances and measurement noises. Also the high frequency gain of the derivative term $sT_d/(1+sT_d/N)$ is limited to avoid noise amplification [2]. The gain limitation can be parameterized in terms of the parameter N. The control scheme can be represented by means of nine transfer functions, Equation (12) (for instance, see [11]), where each close-loop transfer function ($T_{zw}(s)$) denotes the relationship between the output signal z and the input signal ω:

$$\begin{bmatrix} e(s) \\ y(s) \\ u(s) \end{bmatrix} = T(s)\begin{bmatrix} r(s) \\ l(s) \\ d(s) \end{bmatrix} = \begin{bmatrix} T_{er}(s) & T_{el}(s) & T_{ed}(s) \\ T_{yr}(s) & T_{yl}(s) & T_{yd}(s) \\ T_{ur}(s) & T_{ul}(s) & T_{ud}(s) \end{bmatrix}\begin{bmatrix} r(s) \\ l(s) \\ d(s) \end{bmatrix}. \qquad (12)$$

3.1 Modeling Uncertainty

No mathematical model can exactly describe a physical system. For this reason, it is necessary to be aware of the impact that modeling errors have on a control system. Model uncertainty is presented in two different forms: parametric uncertainty and unstructured uncertainty. In this paper an unstructured uncertainty is considered, specifically, a multiplicative perturbation model is proposed. Equation (13) describes the plant with multiplicative perturbation [1, 11, 12, 21, 22]:

$$G(s) = (1 + W_\Delta(s)\Delta(s))G_o(s), \qquad (13)$$

where $G_o(s)$ corresponds to nominal plant, $\Delta(s)$ is the multiplicative perturbation acting on the plant and it is assumed that $\|\Delta(s)\|_\infty \leq 1$. $W_\Delta(s)$ is known as a weighting function and scales $\Delta(s)$ to account for the frequency dependent magnitude of the uncertainty, where it is assumed that the model uncertainty is less than $\pm|W(j\omega)|$.

If $\|\Delta(s)\|_\infty \leq 1$, then inequality (14) holds:

$$\left\|\frac{G(s)}{G_0(s)} - 1\right\| \leq \|W_\Delta(s)\|. \qquad (14)$$

Taking into account the upper bounds just described, the perturbation may be represented as

$$\Delta(s) = \frac{G(s) - G_0(s)}{G_0(s)}.$$ (15)

Since no experimental values of the plant parameters are available, an a priori upper bound for the multiplicative uncertainty is proposed. The weighting function is chosen taking into account the uncertainty introduced at high frequencies, where the influence of neglected or unmodeled dynamics may be significant

The weighting function magnitude plot is shown in Fig. 5.

The weighting function $|W_\Delta(j\omega)|$ is typically small (i.e., $|W_\Delta(j\omega)| = l_g$) at low frequencies, where *Go(s)* accurately represents the system, and large (i.e., $|W_\Delta(j\omega)| = h_g$) at high frequencies $w \gg w_T$, where the influence of unmodeled dynamics may be significant. The W_T represents the transition frequency where the model *Go(s)* becomes unreliable. Transition frequency is considered close to the desired band-width w_B of the closed-loop system. The weighting function, see [1], is modeled by means of

$$[W_\Delta(j\omega)] = \left| h_g \frac{j\omega + \omega_T l_g}{j\omega + \omega_T h_g} \right|.$$ (16)

3.2 Robust Stability for Unstructured Uncertainty

In correspondence with [9], assuming that the nominal feedback system (i.e., with $\Delta=0$) is internally stable for controller C(s), the necessary and sufficient condition for the robust stability of the control system is given in the following **Theorem 1**.

Theorem 1. The uncertain closed loop in Fig. 4 is robustly stable for all Δ such that $\|\Delta(s)\|_\infty \le 1$, iff

$$\| W_\Delta(s)T(s) \|_\infty < 1.$$

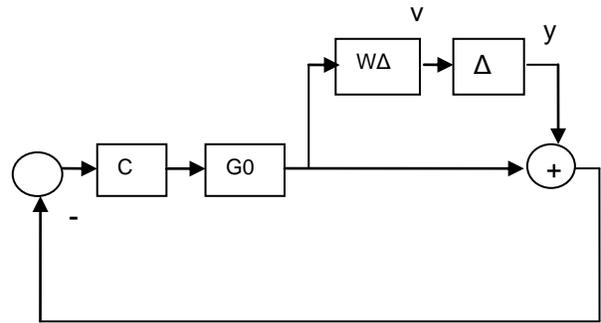*T(s)* in the previous expression is the complementary sensitivity function given by

Fig. 4. Standard feedback loop with multiplicative perturbation

$$T(s) = 1 - S = \frac{L}{1+L} = \frac{G(s)C(s)}{1+G(s)C(s)}.$$ (17)

Likewise, *S* is called the sensitivity function.

4 PID-ISA Tuning Procedure

The design of PID-ISA controllers is formulated as an optimization problem of a series of norms of certain transfer functions that evaluates the control process specifications. The wished specifications of the control system (showed in Fig. 2), can be formulated in terms of the minimization of the following H_2, H_∞ and L_1 norms.

The objective functions are proposed considering set point response, load disturbances, measurement noises and robustness to model uncertainty. Then, the objective functions are formulated.

4.1 Attenuation of Load Disturbance

This effect is measured by the integral of the output signal for an input step, applied in the signal control *u*. Using the principle of superposition, i.e., making zero the values of the reference signal and the measurement noise signal,

$$J_1 = \int_0^\infty |y(t)|^p dt, \quad p \in N,$$ (18)

and normally 1 or 2 is selected for *p*.

Since $y(t)$ in Equation (18) can be expressed by the inverse Laplace transform of $T_{yl}(s)\frac{1}{s}$ t, where $T_{yl}(s)$ is the transfer function between the perturbation signal l and the output signal y, the Equation (18) can be expressed as

$$J_1 = \left(\left\| T_{yl}(s)\frac{1}{s} \right\|_p \right)^{1/p}, \qquad (19)$$

where $\left\| \cdot \right\|_p$ denotes the *p-norm*. In this case, $p=1$(L_1 norm), which allows us to minimize the changes of the output signal y in the temporary domain and in presence of the disturbance signal. This objective is defined in [3, 10]:

$$J_1 = \left\| T_{yl}(s)\frac{1}{s} \right\|_1 \cdot \qquad (20)$$

4.2 Set Point Response

It is important to have a good response to set point changes. Most important points are considered to be the raise time, the settling time, the decay ratio, the overshoot and the steady-state error. To characterize the temporary response of the control system to a reference signal, the following index performance [11] is used:

$$J_2 = \left(\int_0^\infty \left| te(t) \right|^p dt \right)^{1/p} \qquad (21)$$

where $e(t)$ is the output signal corresponding to a step input in $r(t)$. On the other hand, if $e(t)$ is the impulse response of a system, with a transfer function $T_{er}(s)\frac{1}{s}$, then $te(t)$ is the impulse response to a control system with a transfer function $\frac{d}{ds}\left(T_{er}(s)\frac{1}{s} \right)$. Therefore the performance Index J_2 is equivalent to

$$J_2 = \left\| \frac{d}{ds}T_{er}(s)\frac{1}{s} \right\|_p \qquad (22)$$
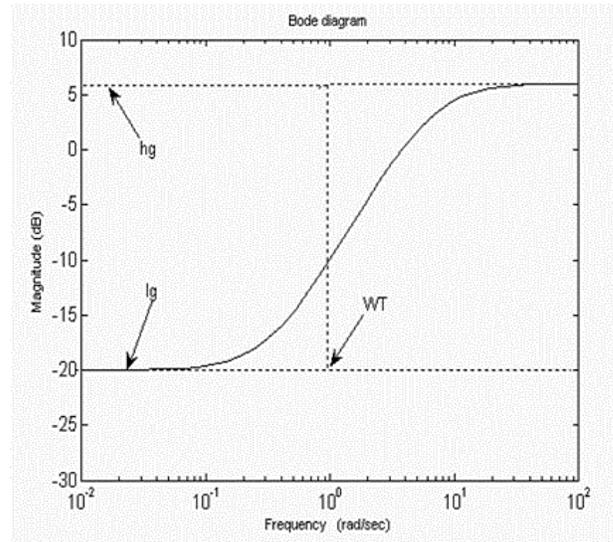
and in this case $p=2$.

Fig. 5. Weighting function magnitude plot $W_\Delta(j\omega)$

4.3 Sensitivity to Modeling Errors

Since the controller is tuned for a particular process, it is desirable that the closed loop system should be not very sensitive to variations of the process dynamics. A convenient way to express the sensitivity of the closed loop system is through the sensitivity function $S(s)$ defined as $S(s) = \frac{1}{1+L(s)}$, where $L(s)$ denotes the loop transfer function. $L(s)$ is given by

$$L(s) = F(s)C(s)G(s)$$
$$= F(s)G(s)k\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1+T_d s/N} \right) \cdot$$

The sensitivity function $S(s)$ is given by the closed-loop transfer function $T_{yd}(s) = \frac{1}{1+L(s)}$, included in Equation (12). The maximum sensitivity (frequency response) is then given by $M_s = \max_\omega |S(i\omega)| \cdot$ Therefore M_s is given by $M_s = \left\| T_{yd}(s) \right\|_\infty$. On the other hand, it is known that the quantity M_s is the inverse of the shortest distance from the Nyquist curve of loop transfer function to the critical point s=-1 [2]. Typical values of M_s are in the range from 1.2 to 2.0. This way, the third objective is given by

$$J_3 = M_s = \|T_{yd}(s)\|_\infty . \tag{23}$$

Objective Function 1 in (20) measures the effect of the load disturbance (*l*); Objective Function 2 in (21) assures a good response to the set point changes, and finally, Objective Function 3 in (23) measures the effect of robustness from the model uncertainties.

The constraints given by (24), (25) y (26), define the feasible region (Ω).The first constraint in (24) guides the algorithm towards the solutions that make the control system stable. The second constraint limits the magnitude of the signal control *u*(*s*). This is very important in practice, since the control designs often produce very large values of *u*(*s*) which would cause actuator saturation. Constraint 2 is expressed by Equation (25).

$$poles(real(T_{yd})) < 0 , \tag{24}$$

$$r_1 = \|T_{ur}(s)\|_\infty < \gamma . \tag{25}$$

Constraint 3 is aimed at ensuring robust stability of the control system subject to unstructured uncertainty [11, 16, 20-22, 24].

$$\|W_\Delta(s)T(s)\|_\infty < 1. \tag{26}$$

In order to build the weight function needed in equality (16), the proposed plants are assumed to have 10% uncertainty at lower frequencies and 200% at high frequencies (with respect to the cutoff frequency).

4.4 Test Plants and Tuning Procedure

The proposed plants in this article cover a wide range**: stable and integrating, with short and long dead times, whit real and complex poles, and with positive and negative zeroes**, which are representative of the automatic control literature [2, 14]. The test plants are shown in Table 1.

5 Results

In this research, the objective functions and constraints of the optimization problem were made as a MATLAB® function to be called from the multiobjective genetic algorithm NSGA-II by using the MATLAB® engine library.

In case of multiobjective optimization, from the Pareto front, different values of PID controllers based on different objectives can be obtained for a particular process. As it was expected, the different Pareto front solutions present different overshoot values such as settling time and rise time ones. Thus, one can select solutions based on system conditions and requirements.

The obtained values of the objective functions and constraints of the optimization problem are consistent with the controller performance according to the requirements from the point of view of the requirements in the time domain and frequency. The proposed objective functions capture the essence of the process control as it can be seen in Table 3. It shows that for larger values of the objective function, M_s, the system response, is increased, however, the overshoot also increases. The response to load disturbance is in accordance with the values of the objective

Table1. Proposed processes

Process	Process Transfer Function
1	$\dfrac{1}{(s+1)^3}$
2	$\dfrac{1}{(s+1)(0.2s+1)(0.04s+1)(0.008s+1)}$
3	$\dfrac{e^{-15s}}{(s+1)^3}$
4	$\dfrac{1}{s(s+1)^4}$
5	$\dfrac{1-2s}{(s+1)^3}$
6	$\dfrac{9}{(s+1)(s^2+2s+9)}$
7	e^{-s}
8	$\dfrac{e^{-s}}{s}$

function *J1*, where the settling time and amplitude are lower when the value of the function *J1* is smaller. Set point following responses are as fast as the values of the objective function *J2* decrease.

In order to assess the performance of the controllers obtained via the procedure described in this paper, several simulations were performed using MATLAB/SIMULINK®. Two distinct PID controllers, namely PID-A$_i$ and PID-B$_i$, were selected from the Pareto set of the *i*-th process example (*i = 1, 2,3,4,5,6,7,8*).

The operation of NSGA-II was configured with the following parameter values:

– Population size: 100.
– Number of generation: 100
– Crossover probability: 0.80
– Mutation Probability: 0.09

Table 2 presents the parameters of robust PID controllers tuned by multiobjective optimization and Table 3 shows the values of the performance indices and constraints.

Table 2. PID parameters for *i*

PID-A i PID-B	T_i	T_F	T_D	N	k	C	b
PID-A	2.21	0.0028	0.645	35.48	4.16	0.618	0.0002
1							
PID-B	2.3	0.005	0.808	35.59	3.84	0.616	0.0141
PID-A	0.70	0.0018	0.100	16.76	7.588	0.691	0.0168
2							
PID-B	0.77	0.0018	0.136	16.29	7.632	0.687	0.0169
PID-A	9.56	2.8828	3.679	51.93	0.335	0.897	0.0967
3							
PID-B	7.45	1.4897	3.895	52.0	0.315	0.929	0.3218
PID-A	13.6	0.0632	1.431	57.24	0.38	0.646	0.1899
4							
PID-B	13.9	0.0052	1.618	66.92	0.525	0.674	0.1100
PID-A	2.6	0.1789	0.797	16.5	0.469	0.937	0.0271
5							
PID-B	2.43	0.0192	1.007	61.20	0.508	0.937	0.0883
PID-A	1.32	0.8977	0.311	48.08	0.586	0.963	0.0011
6							
PID-B	2.53	2.0279	0.383	51.01	2.308	0.116	0.0143
PID-A	0.36	0.0015	0.0015	0.0082	0.258	0.570	0.0016
7							
PID-B	0.34	0.0012	0.0774	0.1245	0.152	0.617	0.0168
PID-A	5.62	0.001	0.467	84.89	0.542	0.57	0.579
8							
PID-B	6.68	0.0009	0.426	84.53	0.693	0.713	0.551

Table 3. Objective functions and constraints

PID	J1	J2	M_s	r_1	$\|TW\|_\infty$
PID-A	0.532	0.931	1.6	4.38	0.806
Process 1 PID-B	0.615	1.083	1.489	4.3	0.724
PID-A	0.092	0.086	1.333	7.38	0.6916
Process 2 PID-B	0.100	0.096	1.283	7.35	0.6364
PID-A	33.56	70.99	1.392	1.98	0.7231
Process 3 PID-B	30.23	57.65	1.516	5.56	0.9919
PID-A	35.69	5.181	1.212	4.37	0.6364
Process 4 PID-B	26.56	4.416	1.237	4.22	0.6364
PID-A	6.766	6.069	1.746	1	0.6488
Process 5 PID-B	6.268	5.326	1.868	3.22	0.6364
PID-A	2.289	1.701	1.176	8.98	0.6364
Process 6 PID-B	1.200	2.741	1.489	8.05	0.6701
PID-A	1.764	0.833	1.449	9.00	0.7022
Process 7 PID-B	2.586	1.619	1.236	9.00	0.6603
PID-A	10.75	2.189	1.347	4.0	0.6880
Process 8 PID-B	9.797	1.232	1.436	4.0	0.8446

The ranges of parameter values of the controllers used in the optimization problem are presented in Table 4.

5.1 Results of Process 1

Fig. 6 shows the Pareto front in three dimensions and their corresponding projections of the objective vectors associated with the PID controllers of Process 1. The dots in Fig. 6 indicate the solutions obtained from the final population. The model Process 1 is a third order model. Fig. 7 shows the simulation result which illustrates the resulting closed-loop system responses to unit-step followed by a load disturbance corresponding to Process 1.

Fig. 8 illustrates the Nyquist curve of a nominal loop transfer function, and the circles show the uncertainty regions. This means that the perturbed Nyquist curve will not reach the critical

Table 4. Lower and upper bounds of controller parameters

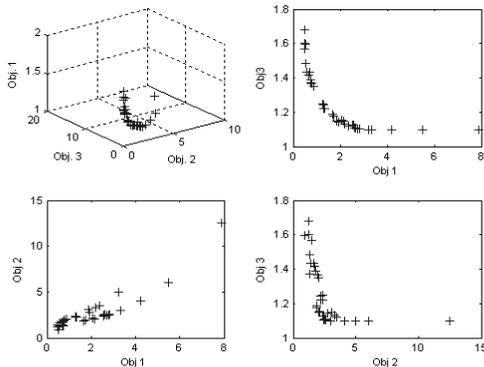| Ti	Tf	Td	N	K	b	c
0-20	0 -20	0-20	0-100	0-10	0-1	0-1

Fig. 6. Pareto front (Process 1, PID-B)
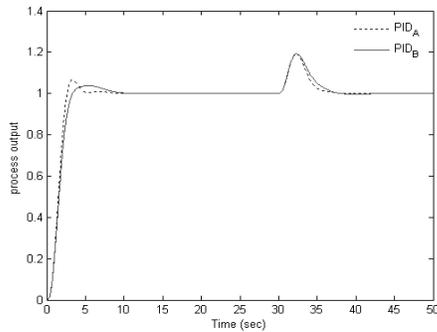
Fig. 7. Set point and load disturbance response process with transfer function $1/(s+1)^3$
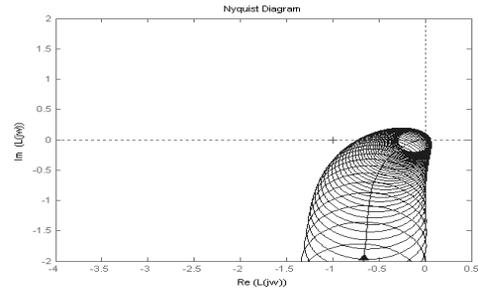

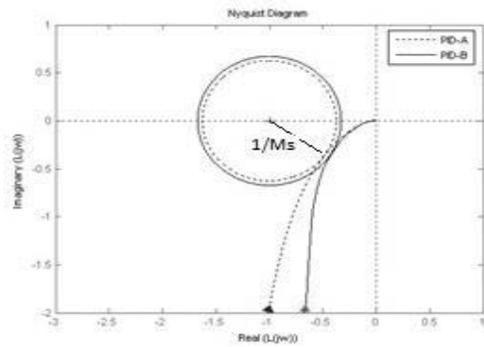**Fig. 8.** Robust stability in Nyquist diagram, Process1 (PID-B)

Fig. 9. Sensitivity M_s in Nyquist diagram, Process 1
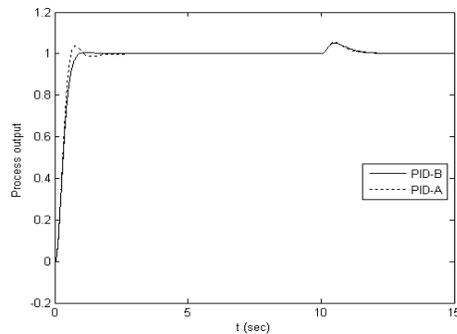
Fig. 10. Set point and load disturbance responses of the Process 2

point −1. Also, the plot in Fig. 8 shows an interpretation of the robust stability condition $\left\| W_\Delta(s)T(s) \right\|_\infty < 1$ in the Nyquist diagram.

Fig. 9 shows the distance from Nyquist curve and the critical point s=-1 for Process 1. A sensitivity M_s guarantees that the distance from the critical point to the Nyquist curve is always greater than $1/M_s$.

5.2 Results of Process 2

The responses to changes in the set point and load disturbance of Process 2 are shown in Fig. 10. The parameter values of the selected controllers (PID-A and PID-B) are found in Table 2. In Fig. 10 the set point response corresponding to PID-B controller can be seen to be very soft (almost no overshoot) as compared to PID-A. The PID-B presents a minor sensitivity value (M_s),

compared with PID-A as it is seen in Table 3, that implies a minor overshoot in relation to PID-A controller. The disturbance load response is very similar in both controllers.

5.3 Results of Process 3

Plant 3 is a model with long dead time. In this work, the dead time was replaced by their first-order Pade approximation. This approximation

added a zero on the right half complex plane (systems of non-minimum phase). The systems of non-minimum phase present slow temporal responses.

The transport delay is a common example of the non-minimum phase systems and occurs mainly in thermal, hydraulic and pneumatic systems. The output signal of Process 3 is depicted in Fig. 11, where one can observe the inverse response at the beginning of the step responses and the load disturbance caused by the delay in the model of the proposed plant.

5.4 Results of Process 4

The model of Process 4 is an integral one, and the results found by the optimization method shows that integrating processes can be treated in the same way as a stable process. The temporal response is depicted in Fig. 12.

5.5 Results of Process 5

The model of Process 5 has a zero on the right half complex plane. This kind of system is said to be a non-minimum phase one; the systems which do not have the minimum phase are more difficult to control.

The output response of the closed loop system is illustrated in Fig. 13 which shows the step
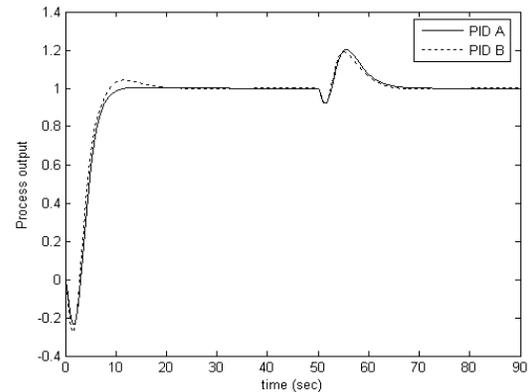

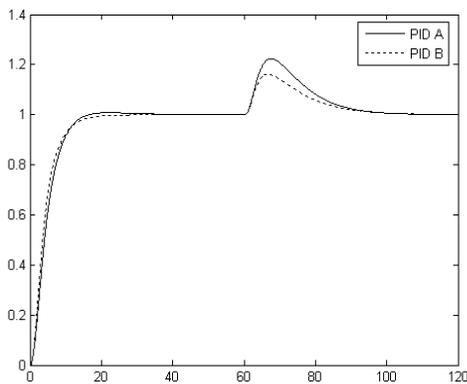**Fig. 11.** Set point and load disturbance response of the Process 3

Fig 13. Set point and load disturbance response of the Process 5

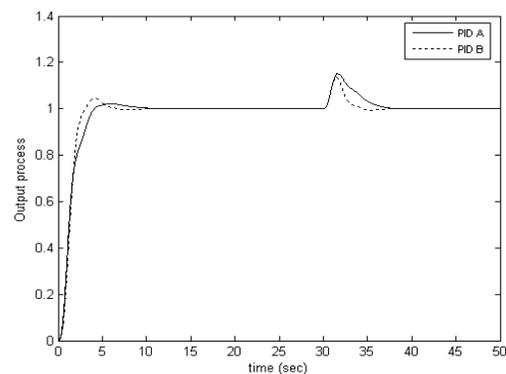**Fig. 12.** Set point and load disturbance response of the Process 4

Fig. 14. Set point and load disturbance response of the Process 6

response followed by a load disturbance. It is observed that the output goes in the wrong direction initially. This is sometimes referred to as inverse response.

5.6 Results of Process 6

Process 6 corresponds to an oscillatory dynamics model with relative damping ζ=0.033, which means the control is difficult [3, 14]. The designed PID controllers behave very well in spite of the poorly damped poles, as it can be seen in Fig. 14. Also, it shows the step and the load responses where the time responses for the close loop system have no oscillations, as there can be in other tuning methods of PID controllers.

5.7 Results of Process 7

Plant 7 is a pure time delay model. In this kind of systems, the phase lag increases linearly with frequency and is difficult to control. This system has $w_{90}= π/2$ and $w_{180}=π.$ The obtained results show that the design procedure produces suitable controller parameters in this example too, as it can be seen in Fig. 15 which shows the closed loop response of Process 7.

5.8 Results of the Process 8

The final example corresponds to a pure integrator with time delay model, this system has w90= π/2 and w180=π. In this case the obtained controllers have good performance too, as it is shown in Fig. 16, where the temporal responses are depicted.

6 Conclusions

The NSGA-II was selected since it is a tool in the public domain and has been quoted in multiple reports concerning evolutionary multiobjective optimization. It is a robust algorithm of general application that can be executed in different platforms. The components of the objective function, as well as the restrictions defining the admissible domain, were formulated in terms of control performance specifications for a PID controller. Results also highlight the merits of
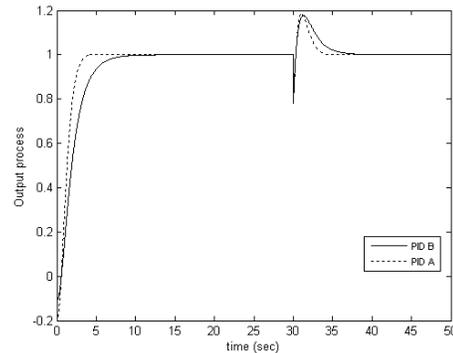

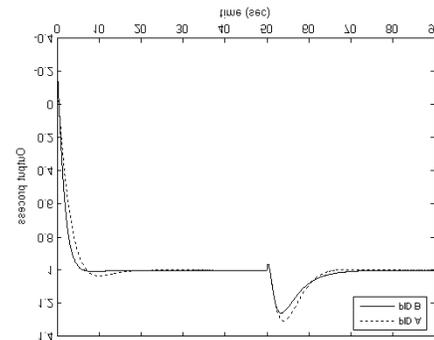**Fig. 15**. Set point and load disturbance response of the Process 7

Fig. 16. Set point and load disturbance response of the Process 8

multiobjective genetic algorithm (NSGA-II) in the application of automatic control.

It was observed that in all presented cases, the two-degrees-of-freedom PID controllers have good performance in terms of set point and load disturbance response process.

It is also important to mention that M_s value was always a referent in relation to a good performance of the designed PIDs, especially at the relative stability; on the other hand, when the M_s value is within the proposed range, this ensures that the controlled systems are insensitive to possible changes in plant models [3]. As shown in Table 2, the fact that constraint 3 is satisfied ensures robust stability to control system subject to unstructured uncertainty. Concerning the problem of high

frequency noise measurement, this is attenuated by the addition of low-pass filter in the derivative term. This means an advantage over the PID controller with pure derivative action.

On the other hand, with regard to the convergence of the genetic evolutionary algorithm NSGA-II, it is known that in practice there is no way to know whether it has reached or not the real Pareto Front (that applies any MOP). A possible stopping criterion is the consecutive lack of new solutions that dominate the ones which are better up to the moment. If there is no progress after a certain number of iterations, it is reasonable to assume that the algorithm converged already, but obviously there is no guarantee of that. This is a handicap of heuristic strategies, so when they stop, there is no guarantee that they have reached the ideal solution.

References

1. **Ackermann, J., Blue, P., Bünte, T., Güvenc, L., Kaesbauer, D., Kordt, M., Muhler, M., & Odenthal, D. (2002).** *Robust Control: The Parameter Space Approach.* London: Springer.

2. **Åström, K.J., Panagopoulos, H., & Hägglund, T. (1998).** Design of PI Controllers based on Non-Convex Optimization. *Automatica*, 34(5), 585–601.

3. **Åström, K.J. & Hägglund, T. (1995).** *PID Controllers: Theory, Design, and Tuning (2nd Ed.).* Research Triangle Park, N.C.: International Society for Measurement and Control.

4. **Bechikh, S., Belgasmi, N., Said, L.B., & Ghedira, K. (2008).** PHC-NSGA-II: A Novel Multi-objective Memetic Algorithm for Continuous Optimization. *20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '08),* Dayton, OH, 1, 180–189.

5. **Campos-Delgado, D.U. & Zhou, K. (2003).** Mixed $\pounds_1/H_2/H_\infty$ control design: numerical optimization approaches. *International Journal of Control*, 76(7), 687–697.

6. **Chakraborty, J., Konar, A., Nagar, A., Das, S. (2009).** Rotation and translation selective Pareto optimal solution to the box-pushing problem by mobile robots using NSGA-II. *IEEE Congress on Evolutionary Computation (CEC '09),* Trondheim, Norway, 2120–2126.

7. **Coello, C.A., Veldhuizen, D.A., & Lamont, G.B. (2002).** *Evolutionary Algorithms for Solving Multi-Objective Problems.* New York: Kluwer Academic Publishers.

8. **Coello, C.A. (1999).** An Update Survey of Evolutionary-Based Multiobjective Optimization Techniques: State of the Art and Future Trends. *1999 Congress on Evolutionary Computation*. Washington, D.C., 3–13.

9. **Coello, C.A. & Becerra, R.L. (2003).** Evolutionary Multiobjective optimization using a Cultural Algorithm. *2003 IEEE Swarm Intelligence Symposium*, Indianapolis, In., USA, 6–13.

10. **Coello, C.A., Lamont, G.B., & Veldhuizen, D.A. (2007).** *Evolutionary Algorithms for Solving Multi-Objective Problems (2nd ed.).* New York: Springer.

11. **Doyle, J.C., Francis, B.A., & Tannenbaum, A.R. (1992).** *Feedback Control Theory.* New York: Macmillan Pub. Co.

12. **Fonseca, C.M. & Fleming, P.J. (1998).** Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part I: A unified Formulation. *IEEE Transactions on Systems, Man, Cybern*etics, 28(1), 26–37.

13. **Gambier, A. (2011).** Control of a Reverse Osmosis plant by using a robust PID design based on multi-objective optimization. *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Orlando, FL, 7045–7050.

14. **Hang, C.C., Åström, K.J., & Wang, Q.G. (2002).** Relay feedback auto-tuning of process controllers - a tutorial review. *Journal of Process Control*, 12(1), 143–162.

15. **Herreros, A. (2000).** *Diseño de controladores Robustos Multiobjetivo por medio de Algoritmos Genéticos*. Tesis Doctoral, Universidad de Valladolid, España.

16. **Jamshidi, M., Krohling, R.A., Coelho, L.D.S., & Fleming, P.J. (2003).** *Robust Control Systems with Genetic Algorithms*. Boca Raton, Fla.: CRC Press.

17. **Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002).** A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.

18. **Kasat, R.B., Kunzru, D., Saraf, D.N., & Gupta, S.K. (2002).** Multiobjective Optimization of Industrial FCC Units using Elitist Nondominated Sorting Genetic Algorithm. *Industrial & Engineering Chemistry Research*, 41(19), 4765–4776.

19. **Kristiansson, B. & Lennartson, B. (2006).** Robust tuning of PI and PID Controllers: using derivative action despite sensor noise. *IEEE Control Systems,* 26(1), 55–69.

20. **Lennartson, B. & Kristiansson, B. (1997).** Pass Band and High frequency Robustness for PID Control. *36th Conference on Decision and Control*, San Diego California, USA, 3, 2666–2671.

21. **Ming-Tzo, H. & Chia-Yi, L. (2002).** PID Controller Design for Robust Performance. *41st IEEE Conference on Decision and Control*, Las Vegas, Nevada USA, 1, 1062–1067.

22. **Panagopoulos, H., Åström, K.J., & Hägglund, T. (1999).** Design of PID Controllers based on Constrained Optimization. *1999 American Control Conference*, San Diego California, USA, 6, 3858–3862.

23. **Sánchez, G., Villasana, M., & Strefezza, M. (2008).** Solving Multi-Objective Linear Control Design Problems Using Genetic Algorithms. *17th World Congress, the International Federation of Automatic Control*, Seoul, Korea, 12324–12329.

24. **Zhou, K. & Doyle, J.C. (1998).** Essentials of Robust Control. Upper Saddle River, N.J.: Prentice Hall.

25. **Wuhua, H., Gaoxi, X., & Wen-Jian, C. (2011).** PID controller design based on two-degrees-of-freedom direct synthesis. *2011 Chinese Control and Decision Conference* (CCDC), Mianyang, China, 629–634.

26. **Xiang-Zhong, G. (2009).** Multi-objective PID Controller Based on NSGA-II Algorithm with Application to Main Steam Temperature Control. 2009 *International Conference on Artificial Intelligence and Computational Intelligence* (AICI '09). Shanghai, China 21–25.

27. **Yamamoto, S., Hirahara, H., Tanaka, A., & Ara, T. (2012).** Simple robust design of Two-Degree-of-Freedom PID controller for tracking drives of Linear Servo Motors. *15th International Conference on Electrical Machines and Systems (ICEMS),* Sapporo, Japan, 1–6.

28. **Ziegler, J.G. & Nichols, N.B. (1942).** Optimum Settings for Automatic Controllers. *Transactions ASME*, 64(11), 759–768.

José Rubén Lagunas Jiménez received a degree in Communications and Electronics Engineering from ESIME-IPN and his Ph.D. in Automatic Control from CINVESTAV-IPN, México, D.F. He is a Researcher and a Professor at the University of Campeche, Mexico. His research includes multiobjective optimization using genetic algorithms and fuzzy control.

Victor Moo Yam received a degree in Electronics Engineering from the University of Campeche and his Ph.D. in Bioelectronics from CINVESTAV-IPN, Mexico, D.F. He is a Researcher and a Professor at the University of Campeche, Mexico. His research includes instrumentation and chemical sensors.

Benjamín Ortíz Moctezuma. received a degree in Mechanical Engineering from ESIME-IPN and his Ph.D. in Automatic Control from CINVESTAV-IPN, Mexico, D.F. He is a Researcher and a Professor at the Polytechnic University of Victoria, Tams., Mexico. His research includes delay systems in automatic control.