Off-line PID control tuning for a planar parallel robot using DE variants

Miguel G. Villarreal-Cervantes[a,*], Jaime Alvarez-Gallegos[b]

[a] *Mechatronic section, Posgraduate Department, Instituto Politécnico Nacional, CIDETEC, Mexico city, 07700, Mexico e-mail: mvillarrealc@ipn.mx*
[b] *Mechatronic section, Cinvestav, Mexico city, 07360 , Mexico e-mail: jalvarez@cinvestav.mx*

Abstract

Optimization methods have shown to be a very important approach for control engineers. They emulate the decision-making ability of a human expert to tune the control gains for a process or system with the formulation and solution of a mathematical optimization problem. In such formulation, evolutionary algorithms (EAs) have been widely used to obtain the control gains. Nevertheless a bad selection of the control gains through the optimization process can result in instability of the closed-loop control system such that the convergence and diversity in the EAs can be compromised. In this paper the PID control tuning for a planar parallel robot with a five-bar mechanism to follow a highly nonlinear trajectory is stated as an off-line nonlinear dynamic optimization problem (OLNLDOP). In order to promote individuals with a stable behavior in the closed-loop control system, a dynamic constraint and a method to handle such constraint is proposed into the OLNLDOP and into eight different variants of the differential evolu-

*Corresponding author

tion algorithm, respectively. Comparative analysis shows that the proposal finds suitable solutions for the OLNLDOP with a better convergence time. Laboratory testing with the optimum PID control gains on a real prototype validates the tuning optimization method.

Keywords:
Controller tuning, optimal tuning, differential evolution, parallel robot.

1. Introduction

Nowadays, intelligent control has increased the interest among researchers (Ruano et al., 2014). One definition for intelligent control is the collection of methodologies, techniques and paradigms from the computational intelligence and soft computing, namely neural networks, fuzzy logic systems or evolutionary algorithms, to rule-based and knowledge-based systems for control engineering field (Ruano, 2007; Reynoso-Meza, Sanchis, Blasco & Freire, 2016a).

In industrial demand the widely used controller is the proportional-integral-derivative (PID), due to the structural simplicity, performance characteristics and the application to a broad class of dynamic systems. Since the emergence of PID control by Elmer Sperry in 1910, there have been more concerns about tuning procedures that guarantee the best performance of the PID controller. The first tuning rules were introduced by Ziegler and Nichols in 1942 (Ziegler & Nichols, 1942) and thereafter, several tuning rules have been proposed. Nevertheless, some of them have been focused on stabilizing the linear system expressed in the Laplace domain (Åström & Hagglund, 2006) and are not suitable for nonlinear dynamic systems such as robot manipulators. The control tuning problem is one of the basic tasks in intelligent control and hence its study could improve the closed-loop performance of a process or system.

Tuning methods can be classified in four categories (a similar classification is found in (Kiam, Chong & Yun, 2005)):

i) *Analytical methods*, where the stability of the closed loop system is analyzed (Lyapunov approaches for nonlinear systems or frequency response, root locus method, etc. for linear systems). In (Kelly, Santibáñez & Loria, 2006) an analytical method to tune the PID controller of robot manipulators is proposed based on knowledge of the inertia matrix and the gravity vector which result from the stability analysis.

ii) *Heuristic methods*, where the experience in the manual tuning of the controller designer is considered, for example in the Ziegler-Nichols rule (Ziegler & Nichols, 1993). These methods have been patented most of the times (Kiam, Chong & Yun, 2005).

iii) *Optimization methods* where a mathematical programming problem (MPP) is stated and an optimization techniques are used to solve it. The main characteristic of these methods is that the control parameters are fixed and obtained through the solution of an off-line numerical optimization method. The off-line numerical optimization method will require the state vector of the system obtained either by the simulated system or by the real one. Once the fixed control parameters are obtained, they can be included in the closed-loop system. A comparative study of three different auto-tuning methods of the PID control gains (heuristic optimization, simplified optimization and direct rules) is discussed in (Romero et al., 2012) for the disturbance rejection problem in linear systems. The analyzed results show a suitable performance in all cases. In (Calva, Niño, Villarreal-Cervantes, Sepúlveda & Portilla, 2013) the optimal gains of the PI controller for a four-bar mechanism with spring and damping forces are obtained by proposing and solving a nonlinear dynamic optimization problem (NLDOP). In order to handle the nonlinearities in the dynamic system, the differential evolution algorithm with a constraint handling mechanism is used to solve the NLDOP. In (Reynoso, Sanchis, Blasco & Herrero, 2012) a multivariable PI controller is tuned by using a multi-objective-evolutionary algorithm. The approach is applied to a linear multivariable process and three different evolutionary algorithms are used. The empirical simulation results show that the performance of the controllers presents different trade-offs between objectives but those controllers can be useful in different situations. In (Wei & Shun, 2010), the quality of the solution in the optimal PID controller gains is improved by modifying the velocity updating equation in the particle swarm optimization (PSO). The simulation results with a pendulum system verify the stable convergence of the proposed modified PSO. In (Helon & Leandro, 2012), the PID control of two-degree-of-freedom robotic system is tuned by proposing a multi-objective optimization problem. NSGA-II algorithm is implemented to solve the problem. Based on simulation results, they state that their approach is a viable alternative. Nevertheless, the experimental results require another analysis and an iterative process is required to link the simulation results to the experimental ones.

iv) *Adaptive methods* where the gains of the controller parameters are on-line tuned based on both the previous methods and a real-time identification technique (Tae & Woong, 2000; Zhiyong, Mingyi & Zhongcai, 2010; Segovia, Sbarbaro & Ceballos, 2004; Kao, Chuang & Chuang, 2006). The main characteristic of these methods is that the control parameters change over a predefined time interval in the closed-loop system.

Regarding tuning approaches based on optimization methods, gradient and meta-heuristic algorithms have been used to solve the optimization problem. Gradient based algorithms have been mainly applied to linear systems (Aström, Panagopoulos & Hagglund, 1998; Bevrani, Hiyama & Bevrani, 2011). In nonlinear systems, the gradient based algorithm presents the drawbacks of the premature convergence to local minimum near the initial condition and the case of failure if discontinuities are presented.

On the other hand, meta-heuristic based algorithms (MHBA) have been widely used to solve real world optimization problems, (Villarreal-Cervantes, Cruz & Alvarez, 2015; Villarreal-Cervantes, Cruz, Alvarez & Portilla, 2010; Cabrera, Nadal, Munoz & Simon, 2007, Portilla et al., 2011). Differential evolution (DE) (Price, Storn & Lampinen, 2005) is a MHBA which can be used to solve continuous, discontinuous, linear, nonlinear, dynamic and static optimization problems. DE performs mutation based on the distribution of the solutions in the current population such that, the search direction depends on the location and selection of individuals. The simplicity, easy implementation, reliability and high performance makes the DE algorithm the most used algorithm in a real world optimization problem (Das & Suganthan, 2011; Ferrante & Ville, 2010) and one of the algorithms used to compare the performance of new algorithms. Nevertheless, some modifications to the original DE schemes have been made to enhance the explorative and exploitative performance (Ferrante & Ville, 2010). The algorithm used to solve the optimization problem depends on the problem at hand (Mezura, Velázquez & Coello, 2006) and hence, a study of the algorithm performance is required. One way to maintain the diversity and the exploration capabilities in evolutionary algorithms for the PI and PID control of a multi-input multi-output (MIMO) is to use chaos theory concepts (chaotic Zaslavskii map) in the mutation process of the differential evolution algorithm instead of the pseudo-random number generator (Dos Santos & Wicthoff, 2011). The simulation results show an improvement in the solution quality of the proposal in a distillation column model. Other researchers have focused on incorporating mechanisms to improve convergence, diversity and constraint handling in multi-objective differential evolution (Reynoso-Meza et al., 2016a, Reynoso-Meza, Sanchis, Blasco & Martínez, 2016b). In (Reynoso-Meza et al., 2016a), control engineer preference handling techniques are incorporated in the optimization process of multi-variable PI controller tuning for a distillation column represented by a linear system with delay. It incorporates a pruning mechanism to promote diversity in the Pareto front and competitive results compared with other multi-variable tuning techniques are shown. With a similar approach to the previous research, in (Reynoso-Meza et al., 2016b) the boiler control problem is solved. In that research the controllers are tuned using linear models of the plant and the obtained control gain is implemented to the corresponding nonlinear explicit model of the plant. The obtained control gains are more sensitive to noise because the optimization problem finds the corresponding fixed gains in a linear dynamic environment being that the real environment is nonlinear.

Similarities in previous works about PID control tuning based on optimization methods are found: 1) In the majority of the cases, they are applied to linear systems, such that the fixed control gain could not efficiently develop the desired task in the real nonlinear plant. 2) Few works are related with nonlinear systems and in those works, simulation results are only taken into account because the real implementation (laboratory testing) of the optimum PID control gains for nonlinear systems requires an appropriate identification of the system and additional considerations related to the real implementation, such that, the control system with optimum control gains obtained by simulation results may even unstabilize the closed-loop system in the real implementation. 3) Different mechanisms and MHBA have been proposed recently in order to improve the performance of the algorithm and then the obtained PID control tuning. To the best of the authors' knowledge, the unstable dynamics of the closed-loop system has not been considered neither the optimization problem nor the solver algorithm, being that it is an important factor in the convergence and diversity of the algorithm to improve the obtained solution.

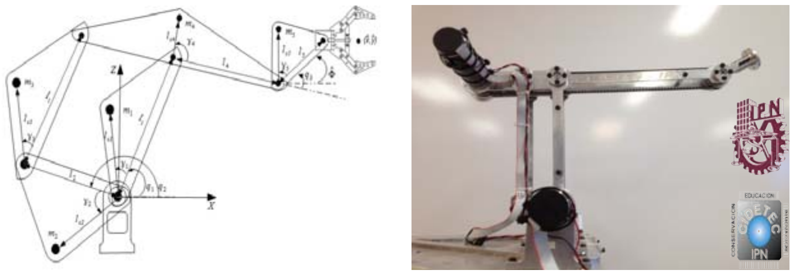
Hence, the main contribution of this paper is *i*) The PID control tuning method for a planar parallel robot with five-bar mechanism (nonlinear system) based on a *dynamic* optimization method with a highly nonlinear trajectory, and its applications to a real prototype, i.e, this work is related to a real-world *nonlinear* optimization problem with laboratory testing. *ii*) The inclusion of a dynamic constraint into the OLNLDOP as well as a method to efficiently handle unstable dynamics into DE variants that promote the generation of better individuals in a dynamic environment and the reduction of the convergence time.

The rest of the paper is organized as follows: The planar parallel robot with five-bar mechanism and its control system are described in Section 2. In Section 3, the off-line nonlinear dynamic optimization problem for the PID control tuning is formulated. The DE variants and the proposed method to handle dynamic constraints are explained in Section 4. The results and discussion are given in Section 5 and finally, in section 6 the conclusions are presented.

2. The robotic system and its PID control

The robot dynamics that is studied in the off-line PID control tuning based on optimization approach, is the planar parallel robot with five-bar mechanism (ppr5bm). The ppr5bm has three degrees of freedom ($d.o.f$) in the joint space which provides the ability to move the tip of the end-effector represented by ($\hat{x}$, $\hat{z}$), in the plane $X-Z$ with an orientation $\hat{\phi}$ with respect to the axis X of the inertial coordinate system $X-Z$. The parallel mechanism into the robot achieves higher precision and higher stiffness than a serial robot (Lung-Wen, 1999), nevertheless it is more difficult to control because it presents more singularity configurations. The ppr5bm is shown in Fig. 1, where $q = [q_1, q_2, q_3]^T$, $\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$, are the joint position and velocity vector, $\tau = [\tau_1, \tau_2, \tau_3]^T$ is the input torque vector. The kinematic and dynamic parameter of the links for the experimental prototype are described in Table 1.

The dynamic model of the ppr5bm in the state space $x = [x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6,\, x_7,\, x_8,\, x_9]^T = [q,\, \dot{q},\, \int_0^\tau e_1(\tau)d\tau,\, \int_0^\tau e_2(\tau)d\tau,\, \int_0^\tau e_3(\tau)d\tau]^T \in R^9$ with



a) *b*)

Figure 1: *a*) Schematic diagram of the $3R$ manipulator with a parallelogram five-bar mechanism. *b*) Real photo of the robot.

the input vector $u = [u_1, u_2, u_3]^T = \tau \in R^3$ is given in (1), where $M \in R^{3\times 3}$ is the inertia matrix, $C[x_3,\, x_4,\, x_5]^T \in R^3$ is the centrifugal and Coriolis force vector, $G \in R^3$ is the gravity vector with the elements of the matrices and vector given by M_{ij}, C_{ij}, G_i $\forall$ $i, j \in \{1, 2, 3\}$, respectively, $e_i = \bar{x}_i - x_i$, $\dot{e}_i = \bar{x}_{i+3} - x_{i+3}$ $\forall$ $i = 1, 2, 3$ is the $i - th$ joint position and velocity error, respectively and the desired state vector is represented by $\bar{x}_i$ $\forall$ $i = 1$, 2, ..., 6.

$$\dot{x} = \begin{bmatrix} \begin{bmatrix} x_4 & x_5 & x_6 \end{bmatrix}^T \\ M^{-1}\left(u - C\begin{bmatrix} x_3 & x_4 & x_5 \end{bmatrix}^T - G\right) \\ \begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}^T \end{bmatrix} \tag{1}$$

where:

$M_{11} = I_1 + I_3 + l_1^2 m_5 + l_{c3}^2 m_3$

$M_{22} = m_3 l_2^2 + m_5 l_4^2 + 2m_5 cos(q_3) l_4 l_{c5} + m_2 l_{c2}^2 + m_4 l_{c4}^2 + m_5 l_{c5}^2 + I_2 + I_4 + I_5$

$M_{33} = m_5 l_{c5}^2 + I_5$

$M_{12} = M_{21} = l_2 l_{c3} m_3 cos(q_1 - q_2) - l_1 l_{c4} m_4 cos(-q_1 + q_2) - l_1 l_{c5} m_5 cos(-q_1 +$

$q_2 + q_5) - l_1 l_4 m_5 cos(q_1 - q_2)$

$M_{32} = M_{23} = m_5 l_{c5}^2 + l_4 m_5 cos(q_5) l_{c5} + I_5$

$M_{31} = M_{13} = -l_1 l_{c5} m_5 cos(-q_1 + q_2 + q_5)$

$C_{11} = 0$

$C_{12} = (l_2 l_{c3} m_3 sin(q_1 - q_2) + l_1 l_{c4} m_4 sin(-q_1 + q_2) + l_1 l_{c5} m_5 sin(-q_1 + q_2 + q_3) - l_1 l_4 m_5 sin(q_1 - q_2))\dot{q_2} + l_1 l_{c5} m_5 sin(-q_1 + q_2 + q_3)\dot{q_3}$

$C_{13} = l_1 l_{c5} m_5 sin(-q_1 + q_2 + q_3)(\dot{q_2} + \dot{q_3})$

$C_{21} = l_1 l_4 m_5 sin(q_1 - q_2)\dot{q_1} - l_1 l_{c5} m_5 sin(-q_1 + q_2 + q_3)\dot{q_1} - l_2 l_{c3} m_3 sin(q_1 - q_2)\dot{q_1} - l_1 l_{c4} m_4 sin(-q_1 + q_2)\dot{q_1}$

$C_{22} = -l_4 l_{c5} m_5 sin(q_3)\dot{q_3}$

$C_{23} = -l_4 l_{c5} m_5 sin(q_3)(\dot{q_2} + \dot{q_3})$

$C_{31} = -l_1 l_{c5} m_5 sin(-q_1 + q_2 + q_3)\dot{q_1}$

$C_{32} = l_4 l_{c5} m_5 sin(q_3)\dot{q_2}$

$C_{33} = 0$

$G_1 = g(l_1 m_4 cos(q_1) + l_1 m_5 cos(q_1) + l_{c1} m_1 cos(q_1) + l_{c3} m_3 cos(q_1))$

$G_2 = g l_{c2} m_2 cos(q_2) - g m_5 (l_4 cos(q_2) + l_{c5} cos(q_2 + q_3)) - g l_{c4} m_4 cos(q_2) + g l_2 m_3 cos(q_2)$

Due to the use of the PID control, the last three states (x_7, x_8, x_9) in (1) are included into the robot dynamics. The PID controller is shown in (2), where k_{p_i}, k_{i_i}, k_{d_i} are the $i - th$ proportional, integral and derivative gains, respectively.

$$u_i = k_{p_i} e_i + k_{d_i} \dot{e}_i + k_{i_i} \int_0^t e_i(\tau) d\tau \ \forall \ i = 1, 2, 3 \tag{2}$$

Then, the closed loop system can be represented by the nonlinear differential equation in (3), where $p = [k_{p_1},\ k_{i_1},\ k_{d_1},\ k_{p_2},\ k_{i_2},\ k_{d_2},\ k_{p_3},\ k_{i_3},\ k_{d_3}]^T \in$

R^9 is the PID control gain vector.

$$\frac{dx}{dt} = f(x, p) \tag{3}$$

3. Off-line PID control optimization problem statement

The PID control gain design for the planar parallel robot with five-bar mechanism (ppr5bm) is formulated as a nonlinear mono-objective dynamic optimization problem (NLMODOP). The control gains are chosen as the design variable vector $p = [k_{p1},\ k_{i_1},\ k_{d_1},\ k_{p2},\ k_{i_2},\ k_{d_2},\ k_{p3},\ k_{i_3},\ k_{d3}]^T \in R^9$.

The control design objective is to provide the gains of the PID control where the end-effector of the ppr5bm follows a desired trajectory and orientation in the Cartesian Space. As the map from Cartesian space to the joint space does not depend on the design variable vector (PID gains), then the Integral Square Error (ISE) in the joint space is chosen as the performance function. The term t_f is included in (4) to obtain the arithmetic mean of the squared error for all time intervals and then, the obtained PID control gains will provide a uniformed error behavior.

$$\bar{J} = \sum_{i=1}^{3} \int_0^{t_f} \frac{e_i^2}{t_f} dt \tag{4}$$

In order to map the desired trajectory from Cartesian space to joint space, the inverse kinematics (Spong & Vidyasagar, 2004) of the ppr5bm must be used. A hypocycloid trajectory in the Cartesian space is selected as the desired one because it presents a highly nonlinear behavior. The hypocycloid trajectory is given in (5)-(7).

$$\bar{\hat{x}} = 0.32 + 0.08181\cos(1.2566t) + 0.01818\cos(5.6548t) \tag{5}$$

$$\bar{\hat{z}} = 0.2 + 0.08181\sin(1.2566t) - 0.01818\sin(5.6548t) \tag{6}$$

$$\bar{\hat{\phi}} = 0.4363\sin(2.0943t) \tag{7}$$

Making a kinematic decoupling (Spong & Vidyasagar, 2004), the wrist desired position ($\bar{\hat{x}}_m$, $\bar{\hat{z}}_m$) is computed according to (8), (9).

$$\bar{\hat{x}}_m = \bar{\hat{x}} - l_5 cos(\bar{\hat{\phi}}) \tag{8}$$

$$\bar{\hat{z}}_m = \bar{\hat{z}} - l_5 sin(\bar{\hat{\phi}}) \tag{9}$$

By using trigonometry in the arm of the ppr5bm, the inverse kinematics is obtained taking into account the wrist desired position. The inverse kinematics is shown in (10)-(12) and they are included as dynamic equality constraints into the NLMODOP.

$$h_1(t) : A\tan 2\left(\frac{\bar{\hat{z}}_m}{\bar{\hat{x}}_m}\right) - A\tan 2\left(\frac{l_4\sin(q_2')}{l_1 + l_4\cos(q_2')}\right) - \bar{x}_1 = 0 \tag{10}$$

$$h_2(t) : \bar{x}_1 + q_2' + \pi - \bar{x}_2 = 0 \tag{11}$$

$$h_3(t) : \bar{\hat{\phi}} - \bar{x}_2 + \pi - \bar{x}_3 = 0 \tag{12}$$

where

$$q_2' = A\tan 2\left(\frac{\sqrt{4l_1^2l_4^2-\left(\bar{\hat{x}}_m^2+\bar{\hat{z}}_m^2-l_1^2-l_4^2\right)^2}}{\bar{\hat{x}}_m^2+\bar{\hat{z}}_m^2-l_1^2-l_4^2}\right)$$

On the other hand, the dynamic behavior of the closed loop robot control system (3) is included into the optimization problem.

3.1. Proposal of constraints to penalize unstable dynamic behavior

The closed-loop system may be unstabilized when the control gains are not properly selected. Once this happens, the robot system cannot be controlled anymore. Hence, one important issue in the dynamic behavior of the closed-loop control system is when unstable design variable vectors (called in this paper unstable individuals) unstabilize the control system and induce undesirable higher order dynamics in the ppr5bm. Then, the evolution of the dynamic behavior of the closed loop robot control system is not suitable (singularity configurations are presented) and the evolution of the dynamic behavior must be stopped. It is important to remark that a gradient based algorithm cannot evaluate the next solution when unstable design variable vector are presented because it diverges. Hence, the inequality constraints given by (13) are proposed in order to penalize unstable individuals. The maximum value of the state vector is selected as $Tol_{Max} = 1e10$. The evaluation of these constraints require the method proposed in Section 4.1.1.

$$g_i(t) : |x_i(t)| - Tol_{Max} < 0, \forall i = 1, 2, .., 9 \tag{13}$$

3.2. Mathematical programming problem

The off-line PID control optimization problem for the ppr5bm consists in finding the design parameter vector p such that the ISE performance is minimized subject to the dynamic constraint (15) given by the nonlinear differential equation that describes the dynamic behavior of the closed loop system, the dynamic equality constraint $h(t) = [h_1(t), h_2(t), h_3(t)]^T \in R^3$ (16) that describes the desired trajectory to be followed at the end-effector of the ppr5bm in the joint space, dynamic inequality constraint $g(x, p, t) = [g_1(t), ..., g_9(t)]^T \in R^9$ (17) to reject undesirable higher order dynamics in the ppr5bm and bound constraints in the design variable vector p (18).

$$\underset{p \in R^9}{Min} \ \bar{J} \tag{14}$$

Subject to:

$$\frac{dx}{dt} = f\left(x, p\right) \tag{15}$$

$$h(t) = 0 \tag{16}$$

$$g(x, p, t) < 0 \tag{17}$$

$$p_{Min} \leq p \leq p_{Max} \tag{18}$$

4. Differential evolution algorithm

The differential evolution (DE) algorithm (Price, Storn & Lampinen, 2005) is a population-based meta-heuristic search algorithm proposed by Storn and Price to solve optimization problems with floating-point param-

BEGIN
$G = 0$; $FNE = 0$; $FE = 0$
Create a random population $\vec{x}^i_G$ $\forall i = 1, ..., NP$
Evaluate $\bar{J}(\vec{x}^i_G)$, $g(\vec{x}^i_G)$, $\forall i = 1, ..., NP$
Do
For $i = 1$ to NP **Do**
Select randomly $\{r_1 \neq r_2 \neq r_3\} \in \vec{x}_G$.
j_{rand} =randint$(1, D)$
For $j = 1$ to D **Do**
Mutation and crossover
End For
Evaluate $\bar{J}(\vec{u}^i_{G+1})$, $g(\vec{u}^i_{G+1})$
If $\vec{u}_{i,G+1}$ is better than $\vec{x}^i_G$ (Based on CHM) **Then**
$\vec{x}^i_{G+1} = \vec{u}^i_{G+1}$
Else
$\vec{x}^i_{G+1} = x^i_G$
End
$G = G + 1$
While $(G \leq G_{Max})$
END

Figure 2: Pseudocode of the DE algorithm with the constraint handling mechanism. $randint(1, D)$ is a uniformly distributed integer random number generator.

eters. The DE is an evolutionary algorithm which consists on four basic stages (initialization, mutation, crossover, and selection) to generate a new individual (child vector). In Fig. 2 the DE algorithm with a constraint handling mechanism (CHM) is presented, where the "FE" and "FNE" terms refer to counter values for the times that the dynamic behavior is completely evaluated or not, respectively.

In the initialization stage, each element of the initial population design vector $x^i_{j,G=0}$ is randomly generated in the interval $[p_{jMin}, p_{jMax}]$ for all individuals of the populations, $i \in [1, .., NP]$ is a population index, $j \in [1, ..., D]$ is the design parameter index, $G \in [1, ..., G_{Max}]$ is the generation number,

NP is the population size, D is the total number of design parameters and G_{Max} is the maximum generation number. For each initial design vector, its objective function is computed.

Once the initial population vector is established, the mutation stage creates a mutant vector $\vec{v}_G^i$ by the mutation of the individuals of the population. The scale factor $F \in (0, 1)$ controls the influence of the selected individuals in order to generate the mutant vector and the mutant vector depends on the different individuals r_1, r_2 and r_3 which are randomly chosen from the range $[1, NP]$.

In the crossover stage, trial vector $\vec{u}_G^i$ is generated when the target vector $x_{j,G}^i$ is recombined with the mutant vector $\vec{v}_G^i$. The influence of the parent vector in the generation of the offspring is controlled by the crossover probability $CR \in [0, 1]$ (higher values mean less influence of the parent vector, hence higher influence of the mutant vector).

The mutation and the crossover stages depend on the DE variant. Some of the most representative variants (Price et al., 2005, Feoktistov & Janaqi, 2004) in the DE algorithm are DE/rand/1/bin, DE/rand/1/exp, DE/best/1/bin, DE/best/1/exp, DE/Current-to-Rand/1, DE/Current-to-best/1, DE/Current-to-rand/1/Bin and DE/Rand/2/Dir. In Fig. 3, the mutation and crossover of those variants are summarized. The term DE means Differential Evolution in the general classification "DE/x/y/z", 'x' represents the way of selecting the individuals that creates the mutant vector, 'y' is the number of difference vectors considered in the mutant vector and 'z' stands for the type of crossover being used (exponential or binomial).

The last stage involves a selection process between the trial vector $\vec{u}_G^i$ and the target vector $x^i_{j,G}$, based on the constraint handling mechanism (CHM) proposed by Deb (Deb, 1998). The constraint handling mechanism provides an elitism way to pass the best individual to the next generation, based on the following rules:

- Any feasible solution is preferred to any infeasible solution.
- Among two feasible solutions, the one having better objective function value is preferred.
- Among two infeasible solutions, the one having smaller constraint violation is preferred.

4.1. Handling dynamic constraints

Dynamic optimization problems (Biegler, 2007) include dynamic constraints due to the dynamic behavior of the system. One way to represent system dynamic constraints is by establishing nonlinear differential equations. An Euler method is used to solve the nonlinear differential equation because for the particular problem, the approximation to the solution with this method is suitable requiring few function evaluations. Two steps are considered (Villarreal-Cervantes et al., 2015) in the dynamic optimization problem in order to transform into a nonlinear programming problem (Betts, 2001):

- Divide the time interval $[t_0, t_f]$ into n time stages, each of length dt, i.e. $t = t_0 = 0$, $t_1 = t_0 + dt, \dots, t_f = t_n$.
- Choosing an initial state vector $\mathbf{x(t_0)}$, the nonlinear differential equation is solved from $t = 0$ to t_f to generate the state trajectory $\mathbf{x(t_k)}$

$\forall k = 1, .., n$. An Euler method is used in this paper to solve the non-linear differential equation.

4.1.1. Handling the unstable dynamic behavior

A method to handle the inequality dynamic constraint (13) is proposed into the DE variant. This method is included as an algorithm in each integration step dt with the evaluation of the inequality dynamic constraint. If the inequality dynamic constraint is not fulfilled in the $k-th$ time stage, the system dynamic behavior is not evaluated and the value $\bar{J} = 1e10-k$ is assigned to the objective function with a counter value of the constraint violation of $1e10 - k$. The time stage k is included into the objective function value and the constraint violation counter, in order to promote individuals that provoke less undesirable higher order dynamics when both individuals are unstable (third rule of the Deb's CHM). In Fig. 4, the pseudocode to handle the inequality dynamic constraint is presented. The "FE" and "FNE" terms refer to counter values for the times that the dynamic behavior is completely evaluated or not, respectively.

5. Results and discussion

The computational experiments are programmed in Matlab on Windows platform on a PC with 3.5 GHz Core i7-4770K and 32 GB of RAM. Ten independent runs are carried out per each previously commented DE variants. The parameters of the DE variants are proposed as follows: The population size $NP = 50$, the maximum number of generation $G_{Max} = 100$, the scaling factor is randomly generated in the interval $F \in [0.3, 0.9]$ at each generation, the factor introduced in the arithmetic recombination from variants Current-to-rand/1, Current-to-best/1 and Current-to-rand/1/Bin is randomly chosen in the interval $K \in [0.3, 0.9]$ at each generation and the crossover constant is given by $CR = 0.8$. The integration time of the dynamic simulation is set as $dt = 5ms$ with a final time $t_f = 20s$.

The lower bound of the design variable vector is set as $p_{Min} = \emptyset \in R^9$. The upper bound is selected via a trial and error procedure based on real experimental tests because the vibration of the robotic system is increased when high derivative gains are selected. The real experimental test consisted in the following steps:

1. Set limits in the design variable vector. For example p_{Max} = [15000, 2000, 50, 10000, 2000, 50, 100, 50, 50]^T.
2. The control gains are obtained through the solution of the optimization problem by using simulation results.
3. The obtained gains are implemented in the real robotic system.
4. In the experimental test the derivative gains increase the noise in the control signal and therefore the vibration of the robot. If the robotic system vibrates then reduce the derivative gain and go to the step 1; else go to step 5.
5. Once the real robotic system does not vibrate, the upper bound can be set.

Through the real experimental test, the upper bound of the design variable vector is selected as $p_{Max} = [300, 2000, 2, 300, 2000, 2, 150, 50, 50]^T$ for the next computational experiments.

The statistical data, i.e., the mean of the objective function $\bar{J}_{mean}$, the standard deviation $\sigma(\bar{J})_{mean}$ and the best objective function value $\bar{J}_{BEST}$ for runs per each DE variant are shown in Table 2.

In order to show whether the results per each DE variant in Table 2 does not provide similar behavior, the Kruskal-Wallis test (Derrac, García, Molina & Herrera, 2011) with a level of significance $\alpha = 0.05$ is used. According to the returned p value ($p = 2.89e - 10 < 0.05$), the null hypothesis is rejected. Thus, the alternative hypothesis is accepted, i.e., the distributions among study cases are different and then, there are significance differences among DE variants.

A graphical result (box plot) of the statistical data is displayed in Fig. 5. It is also observed both in Fig. 5 and in Table 2, there is a significant difference in the behavior among DE variants. Only best/1/bin and best/1/exp provide a convergence to a similar solution in all runs, meanwhile Rand/2/Dir provides more diversity in the solutions (individuals) as can be verified in the standard deviation.

The empirical analysis of DE variants applied to the proposed off-line PID control tuning based on an optimization approach shows that: *i*) The mutation among random individuals in the population results in an efficient explorative search. Nevertheless, it lacks an efficient exploitative search. Therefore, the worst DE variant performance to the particular problem is attributed to Rand/1/Bin, Rand/1/Exp, Current-to-Rand/1, Current-to-Rand/1/Bin, Rand/2/Dir. *ii*) The most competitive variants are Best/1/Bin, Best/1/Exp and Current-to-Best/1. Taking the best individuals in the mutation process promotes the generation of better individuals. Moreover, only the Best/1/Bin can find the best performance function; this is attributed to the binomial crossover. Hence, the best DE variant from the engineering point of view is the Best/1/Bin due to it finds the best solution.

Moreover, it is observed in the fourth column of Table 2 that the performance function of the best individuals between the variants Best/1/Bin and Best/1/Exp vary in a very narrow scale (around $4e-9$). Some insights about their difference in the performance of the trajectory tracking are given. For the Best/1/Bin the Cartesian position error between the desired trajectory and the end-effector trajectory in the $X-Z$ plane given by $\|[\hat{x}, \tilde{z}]-[\bar{\hat{x}}, \bar{\tilde{z}}]\|$, is around 1230 μm and the angular error given by $\|\hat{\phi}-\bar{\hat{\phi}}\|$, is around 1312 μrad. For the Best/1/Exp the position and angular errors are 1232 μm and 1316 μrad, respectively. A difference of 2 μm and 4 μrad is presented between the performance of both DE variants. On the other hand, in the Rand/1/Bin the Cartesian and angular position errors are around 2534 μm 7571 μrad, respectively. When the position errors of Rand/1/Bin are compared with the position errors of Best/1/Bin, the differences are around 1304 μm and 6259 μrad. If we observed the Table 2, the difference in the performance function between Rand/1/Bin and Best/1/Bin is around $5.36e-7$. Therefore, the small variation in the performance function among the DE variants produce a great impact in the trajectory tracking when the precision in the robotic system has the higher priority in the control design.

The performance of DE variants with the inclusion of the proposed method to efficiently handle unstable dynamics is firstly compared with three metaheuristic algorithms: Particle Swarm Optimization (PSO) algorithm, Firefly Algorithm (FA) and Genetic Algorithm (GA). Ten independent runs are carried out per each meta-heuristic algorithms considering the same number of objective function evaluation as in the case of DE variants ($NP \times G_{Max} = 500$). The other parameters of the algorithms are chosen accordingly to their respective work (Poli, Kennedy & Blackwell, 2007; Yang, 2008; Goldberg, 1989). The performance of the PSO algorithm, FA and GA is shown in Table 3. It is observed that the three meta-heuristic algorithms present the term "NaN" which means indeterminate number. Indeterminate number is obtained when at least one individual in the last generation is unstable. This indicates that those algorithms do not efficiently promote the search of stable individuals and hence a poor performance is obtained. The pairwise statistical comparison called Wilcoxon signed ranks test is used to compare the performance between each DE variant (with the proposal) with each of the three meta-heuristic algorithms. In order to compute the Wilcoxon signed ranks test, the term "NaN" is changed by the maximum performance function value among solutions in the last generation. The alternative hypothesis is the left-tailed test which state that the median of each DE variant with the proposal is less than the median of each meta-heuristic algorithm. In all cases (twenty four comparisons) the p-value is less than 0.0001 which indicates that DE variants with the proposal show a significant improvement over the PSO algorithm, FA and GA with a level of significance $\alpha = 0.0001$.

In order to make a fair analysis of the proposed method to efficiently handle unstable dynamics, another ten independent runs per each DE variant are carried out without considering the proposal and using the same previously described parameters of the algorithm. In Table 4, those results are shown. It is observed that Current-to-Rand/1, Current-to-Best/1 and

Current-to-Rand/1/Bin present the term "NaN". This indicates that the use of the current vector as the base vector and the use of two difference vectors in the mutation process, does not efficiently promote the search of stable individuals.

Analyzing Table 2 and Table 4, it is observed that DE variants with the inclusion of the proposed method, find better individuals than the DE variants without the inclusion of the proposed method. This is because the proposal includes a way to know the fitness and constraint violation number of unstable individuals, such that when two unstable individuals compete with each other, the Deb's rules can select the best individual among them. Meanwhile, if the inclusion of the proposed method is not considered into DE variants and in the situation when two unstable individuals compete, the Deb's rules cannot select the best unstable individuals because both present an indeterminate fitness.

Furthermore, in order to compare the performance between DE variants with and without the proposal, the Wilcoxon signed ranks test is used. The R^+, R^- and p-values are shown in Table 5. The term "NaN" in Current-to-Rand/1, Current-to-Best/1 and Current-to-Rand/1/Bin is changed by the maximum performance function value among individuals in the last generation. The alternative hypothesis is the left-tailed test which state that the median of the DE variants with the proposal is less than the median of the DE variants without the proposal. Therefore, the pairwise statistical comparisons state the following: *i*) The inclusion of the proposal in DE variants Rand/1/Bin, Rand/1/Exp, Best/1/Exp, Current-to-Rand/1, Current-to-Best/1 and Current-to-Rand/1/Bin shows a significant improvement over

DE variants without the proposal with a level of significance $\alpha = 0.0001$. *ii*) The inclusion of the proposal in Best/1/Bin shows a significant improvement over DE variants without the proposal with a level of significance $\alpha = 0.2$. This indicates that the risk of rejecting the Null hypothesis, i.e., the Best/1/Bin with and without the proposal does not present a significant improvement, is 19.78%. *iii*) The only DE variant that the proposal does not present a significant improvement is Rand/2/Dir.

In Table 6 the average of the evaluation number of the objective function (FE_{mean}) with and without the proposal for the ten runs are displayed for all DE variants. In addition, the mean convergence time ($Time_{mean}$) is shown in the fourth and sixth column for the proposal and without it, respectively. It is observed that around $11.32\% - 23.16\%$, the objective function is not completely evaluated (see FNE_{mean}) due to the proposed dynamic constraint (13) and the method to handle the inequality dynamic constraint avoids the evaluation of the next time stages when the individuals are unstable.

The behavior of the number of feasible individuals in the population and the number of times that the trial vector is unstable through generations for a specific run of the best DE variant with the proposal is shown in Fig. 6. It is important to remark that in spite of keeping stable individuals in the population from the tenth generation to the final one (see Fig. 6a), the trial vectors could give unstable behavior through the generations (see Fig. 6b). Hence, the proposed method reduces the generation of unstable individuals through the generations and so, the computation time of the DE variant is decreased, as is confirmed in Table 6.

In Fig. 7 the behavior of the performance functions for the best individual found by each DE variant with and without the proposal, is shown through the generations. In the first generations, the individuals in the population are unstable and after that, the individual becomes stable. Only the performance function of stable individuals is displayed in Fig. 7. It is observed that Best/1/Bin and Best/1/Exp find stable individuals in fewer generations than the other DE variants. Making a comparative analysis of the performance function behavior with and without the proposal, it is concluded that the proposed dynamic constraint (13) and the method to handle it, provide stable individuals in a few generations and best individuals are found.

5.1. Performance of the off-line PID control tuning based on an optimization approach in a real platform

The best design variable vector (PID control gains) obtained by the off-line PID control tuning based on an optimization approach is chosen to verify the performance of the PID control system for the trajectory tracking in the end-effector of the ppr5bm. The best design variable vector is given by $p^* = [149.30, 8.66, 1.99, 78.89, 294.56, 1.98, 148.19, 18.06, 1.88]^T$ and with this vector the simulation and experimental results are obtained as is observed in Fig. 8. The maximum position error in the simulation results is $\pm 2e-3m$ and $\pm 2e-3rad$ meanwhile in the experimental result is $\pm 3.5e-3m$ and $\pm 0.017rad$, for the linear and angular displacement (see Fig. 9). The difference between simulation and experimental results is because the identification of the real kinematic and the dynamic parameters of the ppr5bm. In order to improve experimental results, a better identification process to the ppr5bm must be made but this is not in the scope of the paper.

The experimental results show that the proposed off-line PID control tuning can be applied to real systems. The main advantages of the proposed off-line PID control tuning are: 1) It does not require the practical experience of control engineers. 2) Complicated tasks can be handled. 3) This can be applied to a large class of nonlinear systems. On the other hand, the drawbacks are: 1) This is a task based design approach and the obtained gains are only suitable for the predefined task. If other tasks is executed by the robot using the obtained gains, the performance in the trajectory tracking could be compromised. 2) The mathematical model of the system must be known and its parameters should be clearly identified. If those issues are not satisfied, the implementation in a real system of the obtained gains can not give satisfactory results (increase the error and even unstabilize the closed-loop system)

6. Conclusion

The control tuning problem is one of the basic tasks in intelligent control and hence its study could improve the closed-loop performance of a process or system. In this paper an off-line PID control tuning for a planar parallel robot with five-bar mechanism based on *dynamic* optimization method is proposed. A method to efficiently handle unstable dynamics is included into the DE variants of control optimization process. This method promotes both the generation of better individuals when they are unstable, and the reduction of the convergence time.

The experimental results show the applicability of such an approach. In spite of the PID control not ensuring the tracking in highly nonlinear trajectory, the proposed approach finds the best control gains that guarantee a suitable control performance. With the proposed tuning approach the practical experience of the control designer is not required; complicated tasks can be followed by the end-effector of the robot and the approach can be applied to a large class of nonlinear systems. Another advantage of the proposed method is the generation of better individuals in a dynamic environment and the reduction of the convergence time by the inclusion of a dynamic constraint into the OLNLDOP as well as a method to efficiently handle unstable dynamics into DE variants. The proposed method is easy to include for other controller tuning approaches based on meta-heuristic algorithms. One of the main drawbacks is the requirement of using a dynamic model which approximately represents the real nonlinear behavior. If there is no precise mathematical model, then the obtained control gains could not give satisfactory results to the real system. Moreover, the uncertainties in the environment such as handling dynamic load, noise in the control signal and sensors, among others, can affect the performance of the obtained control gains.

The empirical analysis of DE variants applied to the proposed off-line PID control tuning, based on an optimization approach, shows that the best individuals into the mutation process and the binomial crossover (Best/1/Bin) promote the exploitation of the search space and thus the best results are obtained for this particular problem. With the inclusion of the proposed method into the DE variants to efficiently handle unstable dynamics, 75% of DE variants can improve the search to find better solutions with respect to DE variants without the proposal. In addition, the DE variants presents a superior behavior than the PSO algorithm, FA and GA. In all cases the convergence time is reduced.

Future work includes: *i*) To evaluate the inclusion of the method to efficiently handle unstable dynamics into evolutionary *multi-objective* control optimization. *ii*) To compare the performance of the previous commented method with complex multi-objective control tuning strategies. *iii*) The inclusion of a strategy to make robust the off-line controller tuning for nonlinear systems under uncertainties in the real environment based on the optimization method. Experimental results should be given to strengthen the strategy and hence increase the use of intelligent control in the industry field. *iv*) It should be worthwhile to evaluate the method to efficiently handle unstable dynamics in *on-line* controller tuning based on meta-heuristic algorithms in order to face the environment uncertainties and reduce the convergence time at each sample time such that the experimental results could be performed.

Acknowledgment

The first author acknowledges the support from the COFAA of the Instituto Politécnico Nacional and from the SEP-CONACyT, via the project numbers 20160826 and 182298, respectively. The authors gratefully acknowledge the support from Igor Moret Valenzuela for the technical support on figure's edition.

References

Åström, K. J., & Hagglund, T. (2006). *Advanced PID Control*. ISA Press.

Aström, K. J., Panagopoulos, H., & Hagglund, T. (1998). Design of {PI} controllers based on non-convex optimization. *Automatica*, *34*, 585–601.

Betts, J. T. (2001). *Practical methods for optimal control using nonlinear programming*. Society for Industrial Mathematics.

Bevrani, H., Hiyama, T., & Bevrani, H. (2011). Robust {PID} based power system stabiliser: Design and real-time implementation. *International Journal of Electrical Power & Energy Systems*, *33*, 179–188.

Biegler, L. T. (2007). An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification*, *46*, 1043–1053.

Cabrera, J., Nadal, F., Munoz, J., & Simon, A. (2007). Multiobjective constrained optimal synthesis of planar mechanisms using a new evolutionary algorithm. *Mechanism and Machine Theory*, *42*.

Calva, Y. n. M. B., Niño, S. P. A., Villarreal-Cervantes, M. G., Sepúlveda, C. G., & Portilla, F. E. A. (2013). Differential evolution for the control gain's optimal tuning of a four-bar mechanism. *Polibits*, *47*.

Das, S., & Suganthan, P. N. (2011). Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, *15*, 4–31.

Deb, K. (1998). An efficient constraint handling method for genetic algorithms. In *Computer Methods in Applied Mechanics and Engineering* (pp. 311 –338).

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*, 3–18.

Dos Santos, C. L., & Wicthoff, P. M. (2011). A tuning strategy for multivariable {PI} and {PID} controllers using differential evolution combined with chaotic zaslavskii map. *Expert Systems with Applications*, *38*, 13694 – 13701.

Feoktistov, V., & Janaqi, S. (2004). Generalization of the strategies in differential evolution. In *Parallel and Distributed Processing Symposium* (pp. 165–).

Ferrante, N., & Ville, T. (2010). Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review*, *33*, 61–106.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.

Helon, V. H. A., & Leandro, d. S. C. (2012). Tuning of {PID} controller based on a multiobjective genetic algorithm applied to a robotic manipulator. *Expert Systems with Applications*, *39*, 8968 – 8974.

Kao, C. C., Chuang, C. W., & Chuang, R. F. (2006). The self-tuning {PID} control in a slidercrank mechanism system by applying particle swarm optimization approach. *Mechatronics*, *16*, 513 – 522.

Kelly, R., Santibáñez, V., & Loria, A. (2006). *Control of robot manipulators in joint space*. Springer.

Kiam, H. A., Chong, G., & Yun, L. (2005). Pid control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, *13*, 559–576.

Lung-Wen, T. (1999). *Robot analysis: The mechanics of serial and parallel manipulators*. John Wiley & Sons.

Mezura, M. E., Velázquez, R. J., & Coello, C. C. A. (2006). A comparative study of differential evolution variants for global optimization. *Genetic And Evolutionary Computation Conference*, (pp. 485–492).

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, *1*, 33–57.

Portilla, F. E. A., Mezura, M. E., Alvarez, G. J., Coello, C. C. A., Cruz, V. C. A., & Villarreal-Cervantes, M. G. (2011). Parametric reconfiguration improvement in non-iterative concurrent mechatronic design using an evolutionary-based approach. *Engineering Applications of Artificial Intelligence*, *24*, 757 – 771.

Price, K. V., Storn, R. M., & Lampinen, J. A. (2005). *Differential evolution: A practical approach to global optimization*. Springer.

Reynoso, M. G., Sanchis, J., Blasco, X., & Herrero, J. M. (2012). Multiobjective evolutionary algorithms for multivariable {PI} controller design. *Expert Systems with Applications*, *39*, 7895 – 7907.

Reynoso-Meza, G., Sanchis, J., Blasco, X., & Freire, R. Z. (2016a). Evolutionary multi-objective optimisation with preferences for multivariable {PI} controller tuning. *Expert Systems with Applications*, *51*, 120 – 133.

Reynoso-Meza, G., Sanchis, J., Blasco, X., & Martínez, M. (2016b). Preference driven multi-objective optimization design procedure for industrial controller tuning. *Information Sciences*, *339*, 108 – 131.

Romero, P. J. A., Arrieta, O., Padula, F., Reynoso, M. G., Garcia, N. S., & Balaguer, P. (2012). Estudio comparativo de algoritmos de auto-ajuste de controladores pid. resultados del benchmark 2010-2011 del grupo de ingeniería de control de cea. *Revista Iberoamericana de Automática e Informática Industrial*, *9*, 182–193.

Ruano, A. E. (2007). Intelligent control - the road ahead. In *Control Conference (ECC), 2007 European* (pp. 4442–4443).

Ruano, A. E., Ge, S. S., Guerra, T. M., Lewis, F. L., Principe, J. C., & Colnaric, M. (2014). Computational intelligence in control. *Annual Reviews in Control*, *38*, 233 – 242.

Segovia, J. P., Sbarbaro, D., & Ceballos, E. (2004). An adaptive pattern based nonlinear {PID} controller. *{ISA} Transactions*, *43*, 271 – 281.

Spong, M. W., & Vidyasagar, M. (2004). *Robot Dynamics and Control*. John Wiley & Sons.

Tae, Y. K., & Woong, G. H. (2000). An adaptive {PID} learning control of robot manipulators. *Automatica*, *36*, 717 – 725.

Villarreal-Cervantes, M. G., Cruz, V. C. A., & Alvarez, G. J. (2015). Synergetic structure-control design via a hybrid gradient-evolutionary algorithm. *Optimization and Engineering*, *16*, 511–539.

Villarreal-Cervantes, M. G., Cruz, V. C. A., Alvarez, G. J., & Portilla, F. E. A. (2010). Differential evolution techniques for the structure-control design of a five-bar parallel robot. *Engineering Optimization*, *42*, 535–565.

Wei, D. C., & Shun, P. S. (2010). {PID} controller design of nonlinear systems using an improved particle swarm optimization approach. *Communications in Nonlinear Science and Numerical Simulation*, *15*, 3632 – 3639.

Yang, X.-S. (2008). *Nature-Inspired Metaheuristic Algorithms*. Luniver Press.

Zhiyong, T., Mingyi, Y., & Zhongcai, P. (2010). Self-adaptive pid control strategy based on rbf neural network for robot manipulator. In *First International Conference on Pervasive Computing Signal Processing and Applications (PCSPA)*.

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

Ziegler, J. G., & Nichols, N. B. (1993). Optimum settings for automatic controllers. *Journal of Dynamic Systems, Measurement, and Control*, *115*, 220–222.

Table 1: Kinematic and dynamic parameters of the ppr5bm. For all links, the mass center length γ presents zero value.

Parameter	Description	Value	Unit
	Link 1		
m_1	Mass	0.3222	kg
l_{c_1}	Mass center length	0.0524	m
I_1	Inertia	$20.3E-4$	$kg\ m^2$
l_1	Length	0.2	m
	Link 2		
m_2	Mass	0.1894	kg
l_{c_2}	Mass center length	0.0157	m
I_2	Inertia	$1.1E-4$	$kg\ m^2$
l_2	Length	0.05	m
	Link 3		
m_3	Mass	0.1691	kg
l_{c_3}	Mass center length	0.1	m
I_3	Inertia	$12.1E-4$	$kg\ m^2$
l_3	Length	0.2	m
	Link 4		
m_4	Mass	0.9587	kg
l_{c_4}	Mass center length	-0.0643	m
I_4	Inertia	$102.8E-4$	$kg\ m^2$
l_4	Length	0.25	m
	Link 5		
m_5	Mass	0.1349	kg
l_{c_5}	Mass center length	0.0185	m
I_5	Inertia	$101.1E-4$	$kg\ m^2$
l_5	Length	0.072	m

Nomenclature	Variant
Rand/1/Bin	$u^i_j = \begin{cases} v^i_j = x^{r_3}_j + F(x^{r_1}_j - x^{r_2}_j) & \text{if rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x^i_j & \text{otherwise} \end{cases}$
Rand/1/Exp	$u^i_j = \begin{cases} v^i_j = x^{r_3}_j + F(x^{r_1}_j - x^{r_2}_j) & \text{from rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x^i_j & \text{otherwise} \end{cases}$
Best/1/Bin	$u^i_j = \begin{cases} v^i_j = x^{best}_j + F(x^{r_1}_j - x^{r_2}_j) & \text{if rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x^i_j & \text{otherwise} \end{cases}$
Best/1/Exp	$u^i_j = \begin{cases} v^i_j = x^{best}_j + F(x^{r_1}_j - x^{r_2}_j) & \text{from rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x^i_j & \text{otherwise} \end{cases}$
Current-to-Rand/1	$\vec{u}^i = v^i_j = \vec{x}^i + K(\vec{x}^{r_3} - \vec{x}^i) + F(\vec{x}^{r_1} - \vec{x}^{r_2})$
Current-to-Best/1	$\vec{u}^i = v^i_j = \vec{x}^i + K(\vec{x}^{best} - \vec{x}^i) + F(\vec{x}^{r_1} - \vec{x}^{r_2})$
Current-to-Rand/1/Bin	$u^i_j = \begin{cases} v^i_j = x^i_j + K(x^{r_3}_j - x^i_j) + F(x^{r_1}_j - x^{r_2}_j) & \text{if rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x^i_j & \text{otherwise} \end{cases}$
Rand/2/Dir	$\vec{u}^i = v^i_j = \vec{w}^1 + \frac{F}{2}(\vec{w}^1 - \vec{w}^2 + \vec{w}^3 - \vec{w}^4)$ where $f(\vec{w}^1) < f(\vec{w}^2)$ and $f(\vec{w}^3) < f(\vec{w}^4)$

Figure 3: DE variants (Price et al., 2005, Feoktistov & Janaqi, 2004).

```
Begin
  Choose dt, t_f and x(t_0)
  t_1 = 0; J̄ = 0; CICV = 0
  n = t_f/dt + 1
  For k = 1 to n Do
    x(t_{k+1}) = x(t_k) + f(x(t_k), t_k)
    t_{k+1} = t_k + dt
    J̄ = J̄ + Σ_{i=1}^{3} e_i^2 / n
    If g(x_G^i, t_k) > 0 then
      J̄ = 1E10 − k
      CICV = 1e10 − k
      FNE = FNE + 1
      Break the For loop
    End If
  End For
  If J̄ != 1E10 − k
    FE=FE+1
  End If
End
```

Figure 4: Pseudocode of the proposed method included into the DE variants to efficiently handle the inequality dynamic constraint. $CICV$ is a counter of inequality constraint violation.

Table 2: Performance of DE variants for all runs **with** the inclusion of the proposed method to efficiently handle unstable dynamics.

$Algorithm$	$\bar{J}_{mean}$	$\sigma(\bar{J})_{mean}$	$\bar{J}_{BEST}$
Rand/1/Bin	$5.6930e-4$	$1.2850e-6$	$5.67213e-4$
Rand/1/Exp	$5.7046e-4$	$1.4873e-6$	$5.67502e-4$
Best/1/Bin	$\mathbf{5.6668e-4}$	$\mathbf{7.4107e-9}$	$\mathbf{5.66677e-4}$
Best/1/Exp	$5.6671e-4$	$2.3580e-08$	$5.66681e-4$
Current-to-Rand/1	$5.8085e-4$	$8.1736e-6$	$5.70419e-4$
Current-to-Best/1	$5.6789e-4$	$1.9393e-6$	$5.66771e-4$
Current-to-Rand/1/Bin	$5.6943e-4$	$6.7931e-7$	$5.67932e-4$
Rand/2/Dir	$5.7579e-4$	$1.2381e-5$	$5.66796e-4$

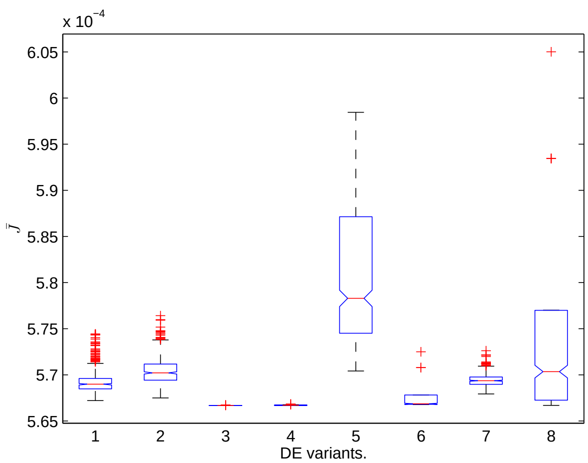


Figure 5: Box plots of the runs of each DE variant **with** the inclusion of the proposed method to efficiently handle unstable dynamics. 1) Rand/1/Bin, 2) Rand/1/Exp, 3) Best/1/Bin, 4) Best/1/Exp, 5) Current-to-Rand/1, 6) Current-to-best/1, 7) Current-to-rand/1/Bin, 8) Rand/2/Dir.

$Algorithm$	$\bar{J}_{mean}$	$\sigma(\bar{J})_{mean}$	$\bar{J}_{BEST}$	$Time$ $[h]$
PSO	NaN	NaN	$6.29414e-4$	0.64
Firefly	NaN	NaN	$5.83381e-4$	0.48
GA	NaN	NaN	$5.79563e-4$	0.53

Table 3: Performance of three different meta-heuristic algorithms for all runs **without** the inclusion of the proposed method to efficiently handle unstable dynamics.

Table 4: Performance of the DE variants for all runs **without** the inclusion of the proposed method to efficiently handle unstable dynamics.

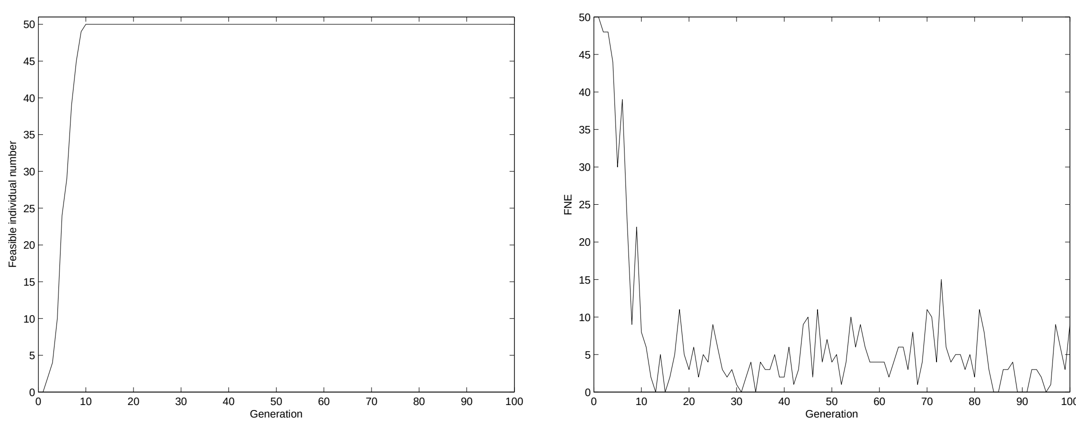
Algorithm	$\bar{J}_{mean}$	$\sigma(\bar{J})_{mean}$	$\bar{J}_{BEST}$
Rand/1/Bin	$5.7401e-4$	$2.9765e-6$	$5.68435e-4$
Rand/1/Exp	$5.8116e-4$	$1.0803e-5$	$5.68365e-4$
Best/1/Bin	$\mathbf{5.6668e-4}$	$\mathbf{1.4878e-9}$	$\mathbf{5.66677e-4}$
Best/1/Exp	$5.6679e-4$	$1.1104e-7$	$5.66685e-4$
Current-to-Rand/1	NaN	NaN	$5.78046e-4$
Current-to-Best/1	NaN	NaN	$5.66983e-4$
Current-to-Rand/1/Bin	NaN	NaN	$5.72385e-4$
Rand/2/Dir	$5.6951e-4$	$4.3610e-6$	$5.66847e-4$

Table 5: Wilcoxon signed rank test results. Comparative statistical results for the DE variants **with** and **without** the inclusion of the proposed method to efficiently handle unstable dynamics.

DE variant comparisons **with** the proposal versus **without** the proposal	R^+	R^-	$p-value$
Rand/1/Bin	55	0	< 0.0001
Rand/1/Exp	55	0	< 0.0001
Best/1/Bin	14	41	0.198
Best/1/Exp	55	0	< 0.0001
Current-to-Rand/1	55	0	< 0.0001
Current-to-Best/1	55	0	< 0.0001
Current-to-Rand/1/Bin	55	0	< 0.0001
Rand/2/Dir	5	50	1

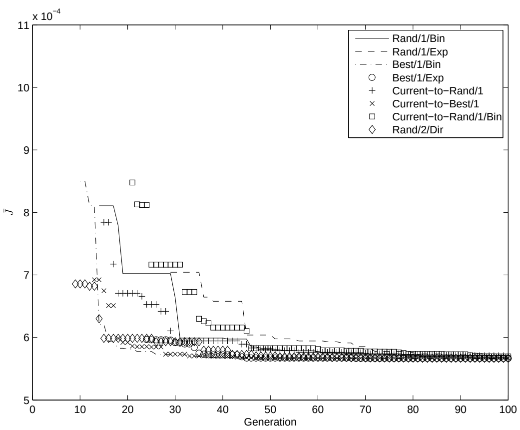
Table 6: Evaluation of the performance function and convergence time for the DE variants **with** and **without** the inclusion of the proposed method to efficiently handle unstable dynamics.

	with the proposal			**without the proposal**	
$DE\ Variant$	FE_{mean}	FNE_{mean}	$Time_{mean}\ [h]$	FE_{mean}	$Time_{mean}\ [h]$
Rand/1/Bin	4049	1001	0.37	5050	0.61
Rand/1/Exp	3880	1170	0.36	5050	0.64
Best/1/Bin	4328	722	0.41	5050	0.55
Best/1/Exp	4275	775	0.40	5050	0.56
Current-to-Rand/1	4233	817	0.41	5050	0.64
Current-to-Best/1	4478	572	0.42	5050	0.57
Current-to-Rand/1/Bin	4102	948	0.39	5050	0.62
Rand/2/Dir	4313	737	0.41	5050	0.54

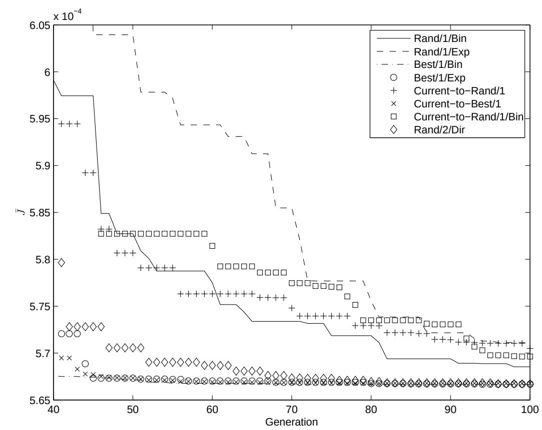


$a)$ $b)$

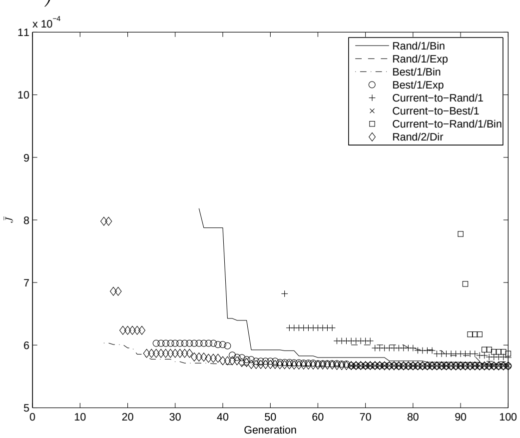
Figure 6: Run 3 of the variant Best/1/Bin with the inclusion of the proposed method to efficiently handle unstable dynamics. a) Behavior of the number of feasible individuals in the population through generations. b) Behavior of the number of times that objective functions were not completely evaluated in the population through generations (unstable vector).



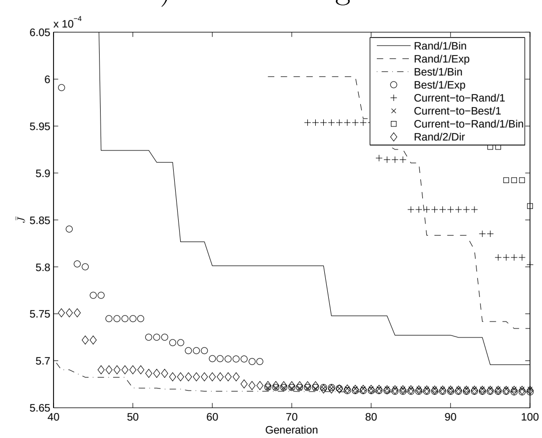
a) Performance function behavior.



b) Zoom of Fig. 7a.

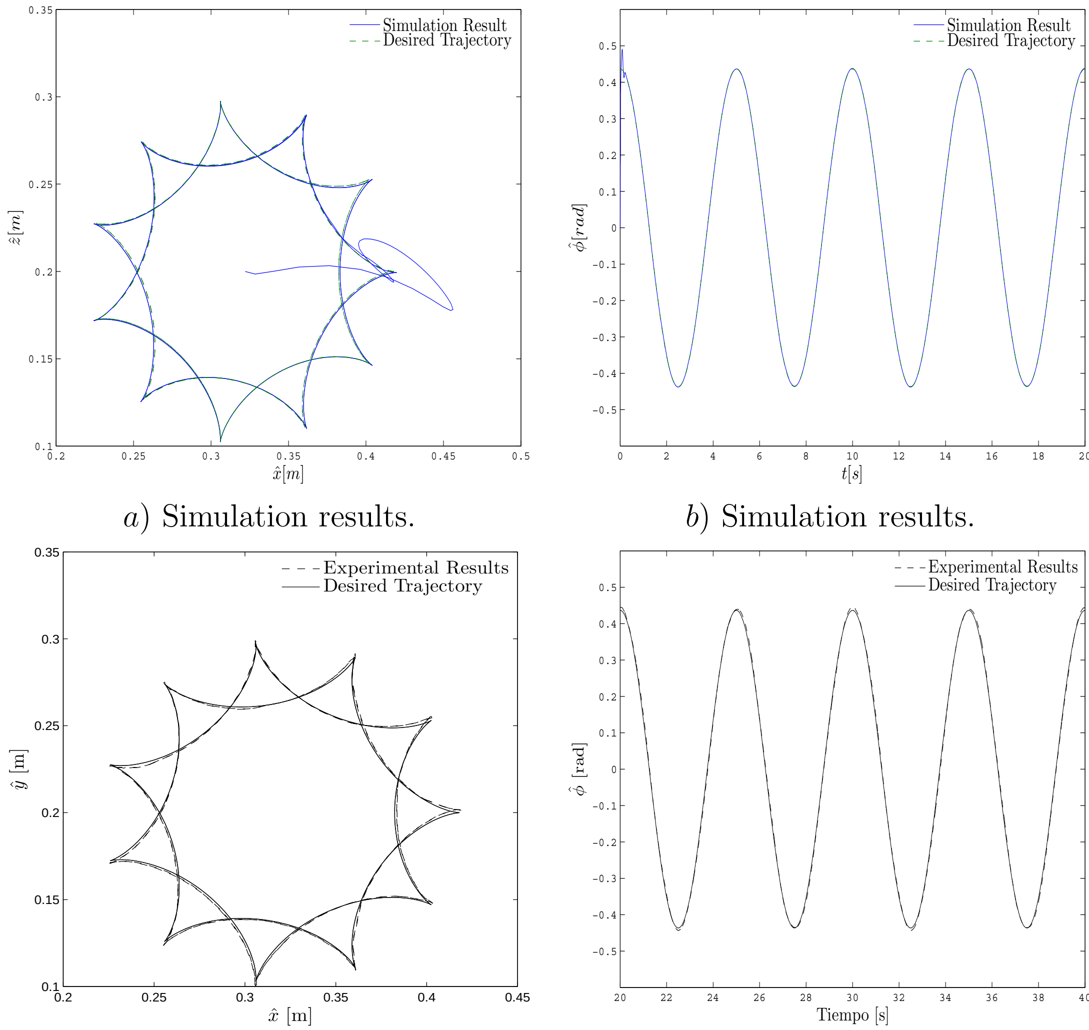


c) Performance function behavior.



d) Zoom of Fig. 7c.

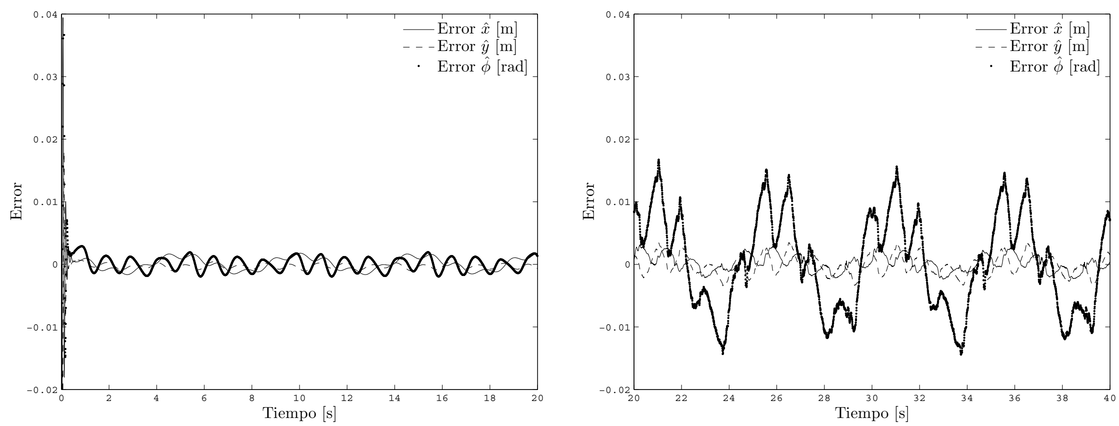
Figure 7: Behavior of the performance function in the generations. a) and b) with the proposal. c) and d) without the proposal.



$a)$ Simulation results. $b)$ Simulation results.

$c)$ Experimental results. $d)$ Experimental results.

Figure 8: Simulation and experimental results for the trajectory tracking with the best PID control gains.



a) Simulation results. *b*) Experimental results.

Figure 9: Trajectory tracking position error for the experimental results.

List of figure captions:

Figure 1: *a*) Schematic diagram of the $3R$ manipulator with a parallelogram five-bar mechanism. *b*) Real photo of the robot.

Figure 2: Pseudocode of the DE algorithm with the constraint handling mechanism. $randint(1, D)$ is a uniformly distributed integer random number generator.

Figure 3: DE variants (Price, Storn & Lampinen, 2005; Feoktistov & Janaqi, 2004).

Figure 4: Pseudocode of the proposed method included into the DE variants to efficiently handle the inequality dynamic constraint. $CICV$ is a counter of inequality constraint violation.

Figure 5: Box plots of the runs of each DE variant **with** the inclusion of the proposed method to efficiently handle unstable dynamics. 1) Rand/1/Bin, 2) Rand/1/Exp, 3) Best/1/Bin, 4) Best/1/Exp, 5) Current-to-Rand/1, 6) Current-to-best/1, 7) Current-to-rand/1/Bin, 8) Rand/2/Dir.

Figure 6: Run 3 of the variant Best/1/Bin with the inclusion of the proposed method to efficiently handle unstable dynamics. a) Behavior of the number of feasible individuals in the population through generations. b) Behavior of the number of times that objective functions were not completely evaluated in the population through generations (unstable vector).

Figure 7: Behavior of the performance function in the generations. a) and b) with the proposal. c) and d) without the proposal.

Figure 8: Simulation and experimental results for the trajectory tracking with the best PID control gains.

Figure 9: Trajectory tracking position error for the experimental results.

List of table captions:

Table 1: Kinematic and dynamic parameters of the ppr5bm. For all links, the mass center length γ presents zero value.

Table 2: Performance of DE variants for all runs **with** the inclusion of the proposed method to efficiently handle unstable dynamics.

Table 3: Performance of three different meta-heuristic algorithms for all runs **without** the inclusion of a constraint and the method to efficiently handle unstable dynamics

Table 4: Performance of the DE variants for all runs **without** the inclusion of the proposed method to efficiently handle unstable dynamics

Table 5: Wilcoxon signed rank test results. Comparative statistical results for the DE variants **with** and **without** the inclusion of the proposed method to efficiently handle unstable dynamics.

Table 6: Evaluation of the performance function and convergence time for the DE variants **with** and **without** the inclusion of the proposed method to efficiently handle unstable dynamics.