Received August 13, 2017, accepted September 18, 2017, date of publication September 29, 2017,
date of current version October 25, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2757959

Multi-Objective On-Line Optimization Approach for the DC Motor Controller Tuning Using Differential Evolution

MIGUEL G. VILLARREAL-CERVANTES[1], ALEJANDRO RODRÍGUEZ-MOLINA[1], CONSUELO-VARINIA GARCÍA-MENDOZA[2], OLLIN PEÑALOZA-MEJÍA[3], AND GABRIEL SEPÚLVEDA-CERVANTES[1]

[1]Postgraduate Department, Centro de Innovación y Desarrollo Tecnológico en Cómputo of the Instituto Politécnico Nacional, Mexico City 07700, Mexico
[2]Instituto Politécnico Nacional, ESCOM, Mexico City 07700, Mexico
[3]Instituto Tecnológico de Sonora, Sonora 85000, Mexico

Corresponding author: Miguel G. Villarreal-Cervantes (mvillarrealc@ipn.mx)

This work was supported in part by the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN) under Grant SIP-20170783, Grant SIP-20172317, and Grant SIP-20160819, and in part by the Consejo Nacional de Ciencia y Tecnología under Grant 281728. The work of A. Rodríguez-Molina was supported by the Mexican Consejo Nacional de Ciencia y Tecnología through a scholarship to pursue graduate studies at CIDETEC-IPN.

ABSTRACT The dc motor is one of the most fundamental electromechanical devices of mechatronic systems, which plays an important role in maintaining the accuracy in the execution of tasks. One of the main issues in the accuracy and robustness of dc motor control system is how to optimally tune its parameters. In this paper, a multi-objective online tuning optimization approach is proposed to adaptively tune up the velocity control parameters of the permanent magnet dc motor. This approach simultaneously considers the modeled error and the corresponding sensitivity to choose the best compromise solution in the Pareto dominance-based selection process of solutions to deal the changing optimum solutions in the dynamic environment of the tuning approach based on online optimization method and moreover, the modified differential evolution with induced initial population based on non-dominated solution through a memory is proposed to guide the search into the feasible region, and to promote the exploitation of solutions found in the previous time interval. Simulation results verify that proposed modifications provide higher robustness and better quality in the velocity regulation control of the dc motor under parametric uncertainties, and also under discontinuous dynamic load, than multi-objective differential evolution, particle swarm optimization, and non-dominated sorting genetic algorithm-II.

INDEX TERMS Controller tuning, on-line tuning optimization method, intelligent control, multi-objective evolutionary optimization, DC motor.

I. INTRODUCTION

The interest of methodologies from computational intelligence and soft computing to rule-based and knowledge-based systems in the control engineering field has increased in the last years. Those methodologies are grouped in the definition of intelligent control [1], [2]. One of the basic tasks in intelligent control is the control tuning problem. Its study could improve the closed-loop performance of a process or system.

One of the main issues in the tuning of the control strategy for an electro-mechanical system is the uncertainty in the system parameters, signal noise as well as dynamical changes in the load. Such uncertainties perturb the stabilization of the system. Generally, the fulfilment of a set of specifications in control engineering problems is a challenge in such situations.

Tuning methods have been classified according to their nature and use in [3]: *i*) Analytical methods where control gains are obtained by analyzing the stability of the closed-loop system; *ii*) Heuristic methods where the experience in the manual tuning of the controller design is considered to set the control parameters; *iii*) Optimization methods where a mathematical programming problem is stated and optimization techniques are used to obtain the fixed control gain parameters; and *iv*) Adaptive tuning methods where an identification process and a combination of the three previous methods are used to on-line tune the control gains.

Due to the increment of precision machines with a required trade-off, the problem of control tuning has being tackled with the use of optimization methods. Over the past decades meta-heuristic algorithms and in particular evolutionary algorithms (EAs) [4], whose inspiration is taken from the natural evolution theory and the survival of the fittest have been used as a successful alternative for the controller tuning based on optimization methods [5]–[7], because they can efficiently handle the highly non-linear trade-offs among multiple closed-loop performance indicators, and incorporate mechanisms (flexibility) to improve their convergence and diversity.

In spite of emerging new EAs, the differential evolution (DE) algorithm is one of the most robust and precise methods when compared with particle swarm optimization (PSO), artificial bee colony (ABC), and cuckoo search (CS) for over fifty different benchmark functions [8]. Moreover, it has been successfully applied for several engineering problems.

To establish a clear background with a specific research direction, we review here only those studies which deal directly with tuning based on optimization methods where the solution is given by meta-heuristic algorithms. In this research direction, these studies can be classified according to the way in which the optimization is carried out: *i*) Off-line optimization and *ii*) On-line optimization (based on the adaptive tuning method).

Tuning approaches with off-line optimization methods require a detailed dynamic description of the system to simulate its behavior in order to search the optimal control parameters with evolutionary techniques. Once the control parameters are optimized, they can be used to the specific application and remain fixed. Several works have tuned linear controllers (PID, PI or PD control) by using off-line optimization methods [2], [9]–[11]. In [2], control engineering preference handling techniques are incorporated in the optimization process of multi-variable PI controller tuning for the benchmark multi-variable control problem known as ''the distillation column''. The results indicate that the approach is useful when a list of performance requirements in the individual loops and the overall system must be fulfilled and presents competitive results compared with other multi-variable tuning techniques. With a similar approach as in the previous research, in [9] the boiler control problem is solved. In that research the controllers are tuned using plant linear models and the obtained control gain is implemented to the corresponding non-linear explicit plant model for which the control gain is more sensitive to noise because the optimization problem finds the corresponding fixed gain in a linear dynamic environment being that the ''real'' environment is non-linear. In order to deal the uncertain process parameters, in [10], the robustness of the solution is considered for the control tuning of two benchmark control problems. The PSO is used to find such gains by using the linearized system. The results show that without considering the system perturbations, the optimal controller is only adequate for the linearized model, but the stability of the closed-loop system in the whole operating range (i.e., by using the non-linear model which includes the uncertainties) cannot be guaranteed. Another approach tackles the way to maintain the diversity of solutions and the exploration capability in evolutionary algorithms for the PI and PID control of a multi-input multi-output (MIMO) system [11]. The proposal 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. The simulation results show an improvement in the solution quality of the proposal in a distillation column model.

Several applications can be benefited by the good features of meta-heuristic algorithms in the off-line optimization methods as in the estimation of the operator functional state based on electro-encephalography measure with the use of incremental-PID-controlled particle swarm optimization [12]; in the control tuning for the azimuth and tilt angles of a solar tracking system by using swarm intelligence algorithms [13]; in the optimum control gain tuning for the desired drug dose in the cancer chemotherapy treatment by implementing a multi-objective genetic algorithm [14]; in the PID control tuning to the structure-control design framework by using the differential evolution algorithm [15], [16]; in the passive optical networks [17], and in the optimum linear quadratic regulator for the tracking control of a laboratory helicopter by linearizing the non-linear model and using an adaptive particle swarm algorithm [18].

On the other hand, tuning approaches with on-line optimization methods often require an identification process to estimate the plant dynamics and the optimal control parameters are usually obtained at a predefined sampling time such that, the control parameters change during time in order to improve the closed-loop system performance under parametric uncertainties. Tuning approaches with on-line optimization methods by using meta-heuritic algorithms have been recently explored. Some approaches use the data of a real prototype in the optimization problems instead of requiring a simplified dynamic system model [19], [20]. Then, fixed control gains are obtained which produce a better performance due to the consideration of the data of the real system. Nevertheless, according to the previous commented classification of the tuning methods, such approaches can be considered as those with off-line optimization methods requiring real data.

As it was commented in [9] and [10], one of the main issues in the control tuning approach based on off-line optimization methods is that the obtained optimum gains are not suitable with environments where time-varying parameters are presented and not included in the optimization problem. In real applications, the experimental system often includes parametric uncertainties such as, sensor noise, changes in the load and variations in the system parameters due to the mechanical wear, tear, etc. In the control tuning approach based on on-line optimization methods, parametric uncertainties may be adequately compensated whether the optimization problem and the optimization technique are well established. To the best author's knowledge, on-line optimization methods with uncertainties in the plant have not been formally addressed and analyzed in the control tuning based on evolutionary algorithms. Hence, in order to deal the changing optima in the dynamic environment of the on-line optimization methods (robustness), the sensitivity of the error velocity with respect to the design variables are included as one of the performance functions in the multi-objective dynamic optimization problem. In addition, the differential evolution algorithm is modified to enhance the performance of the control objective such changes include the constraint handling technique based on the set of feasibility rules and Pareto dominance; the use of an external memory of non-dominated solutions through generations and the preference handling mechanism based on L_p-metrics. Through the statistical analysis, it was observed that the proposal outperformed the velocity control accuracy of the DC motor with parametric uncertainties in the dynamic environment with respect to three different meta-heuristic algorithms.

The rest of the paper is organized as follows: In Section II, the adaptive control for the DC motor is formally stated as an on-line optimization method. The Modified Differential Evolution with Induced Initial Population based on Non-Dominated Solution through a Memory (MDE-IIP-NDSM) is explained in Section III. The comparative analysis among other DE variants and meta-heuristic algorithms are presented and discussed in Section IV. Finally, in Section V, the conclusions are given and the future work is stated.

II. ADAPTIVE CONTROLLER FOR DC MOTOR BASED ON ON-LINE OPTIMIZATION METHOD

A. DC MOTOR DYNAMICS AND VELOCITY CONTROL SYSTEM

Let the state variable vector be $x = [q_m, \dot{q}_m, i_a]^T$, the input signal $u = V_{in}$, and the parameter vector of the permanent magnet DC motor $p = [p_1 = \frac{b_0}{J_0}, p_2 = \frac{k_m}{J_0}, p_3 = \frac{k_e}{L_a}, p_4 = \frac{R_a}{L_a}, p_5 = \frac{1}{L_a}, p_6 = \frac{\tau_L}{J_0}]^T$, where V_{in} is the armature voltage, R_a is the armature resistance, L_a is the armature inductance, k_e is the back electro-motive force constant (back emf), i_a is the armature current, b_0 is the viscous friction coefficient of the motor shaft bearing, J_0 is the inertia of the rotor, k_m is the torque constant, τ_L is the load torque and q_m, $\dot{q}_m$, $\ddot{q}_m$ are the position, velocity and acceleration of the rotor, respectively, then, the dynamic model of the DC motor in the state variable vector x can be expressed in state-variable form $\dot{x} = f(x(t), u(t), p)$ as in (1).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -p_1 & p_2 \\ 0 & -p_3 & -p_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} 0 \\ p_6 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ p_5 \end{bmatrix} u \tag{1}$$

A velocity control is included to make the closed-loop system. Assuming that the system state is available so the inverse dynamic control $u(t) = \tilde{f}(x(t), \bar{p})$ can be used. The terms in (2) are: $e = w_r - x_2(t)$ is the error between the desired angular velocity w_r and the current angular velocity $x_2(t)$, $\dot{e} = \dot{w}_r - \dot{x}_2(t)$ is the error between the desired angular acceleration $\dot{w}_r$ and the current angular acceleration $\dot{x}_2(t)$, $\ddot{w}_r$ is the rate of change of the desired angular acceleration, $\bar{p}(t) \in R^6$ is the control parameter vector, and $k_p = 34524$, $k_d = 368$ are the selected proportional and derivative gains. It is important to mention that the regulation control problem is regarded in this paper, where the velocity reference is set as a constant reference $w_r = 52.35\ rad/s$ and $\dot{w}_r = 0\ rad/s^2$.

$$u = \frac{\ddot{w}_r + k_p e + k_d \dot{e} + \bar{p}_1 \bar{p}_2 x_3 - \bar{p}_1^2 x_2 + \bar{p}_1 \bar{p}_6}{\bar{p}_2 \bar{p}_5} + \frac{\bar{p}_3 x_2}{\bar{p}_5} + \frac{\bar{p}_4 x_3}{\bar{p}_5} \tag{2}$$

It is assumed that the current parameter vector $p(t)$ in the dynamic model of the DC motor (1) dynamically changes its value. Hence, the parameters $\bar{p}$ of the control system (2) must be estimated at each time interval Δt in order to compensate the nonlinearity effects of the DC motor parameter variations.

Uncertainties assumed in the plant are given by changing up to 10% of the nominal DC motor parameters in a sinusoidal way and by giving a discontinuous dynamic load in a specific time interval. Hence, time-varied parameters (TVP) with the discontinuous dynamic load can be written in a mathematical form as shown in Table 1.

TABLE 1. Uncertainties into DC motor nominal parameters. The continuous uncertainty is added to each nominal parameter in a sinusoidal way. The discontinuous uncertainty is included into the load torque in a specific time interval.

TVP	Value	Unit
J_0	$3.45E-4(1+0.1sin(2/3\pi t))$	Nms^2
k_m	$0.3946(1+0.1sin(2\pi t))$	Nm/Amp
b_0	$5.85E-4(1+0.1sin(\pi t))$	Nms
R_a	$9.665(1+0.1sin(2/3\pi t))$	Ω
k_e	$0.4133(1+0.1sin(2\pi t))$	Vs/rad
L_a	$0.10244(1+0.1sin(\pi t))$	H
τ_L	0.05 if $t \in [1,2]$ else 0	Nm

B. CONTROL DESIGN PARAMETER VECTOR

Definition 1: Let the current time be $t = 0, \Delta t, 2\Delta t, \ldots, n\Delta t = t_0, t_1, t_2, \ldots, t_f$ where the Δt is the sampling time. The time space Ω is defined as $\Omega = \{\lambda \in R \mid \lambda \in [t_m, t_n] \subseteq t, t_m = t_n - \Delta w, \ \Delta w > \Delta t, \ t_n \geq \Delta w\}$.

In this paper the vector $\bar{p}(t) = [\bar{p}_1, \bar{p}_2, \bar{p}_3, \bar{p}_4, \bar{p}_5, \bar{p}_6]^T$ in the control system (2) is chosen as the design parameter vector to be found at each sampling time Δt as long as $t \in \Omega$ by solving the on-line dynamic optimization method in the back time interval Δw.

C. PERFORMANCE FUNCTION

Definition 2: If the dynamic model of the DC motor (1) considers the control design variable vector $\bar{p}$ and the control input u, then the resulting dynamics $\dot{\bar{x}} = \bar{f}(\bar{x}(t), u(t), \bar{p})$ is called estimated dynamics of the DC motor (estimated DC motor model), where $\bar{x}$ is the estimated state vector considering the control design parameter vector $\bar{p}$.

Otherwise, if the dynamic model of the DC motor includes the parameter vector p and the control input u, then the resulting dynamics $\dot{x} = f(x(t), u(t), p)$ is called real dynamics of the DC motor (DC motor model).

In order to compensate the non-linear effects of the current parameter vector $p(t)$ in the dynamic model of the DC motor, the parameter vector $\bar{p}$ is estimated according to the minimization of J_1 which is the modelled error $\bar{e} = x - \bar{x} \in R^3$ between the states of the DC motor x and the estimated DC motor $\bar{x}$ considering the time interval Δw. The performance function J_1 provides a measure to know how close the behavior of the estimated DC motor resembles the behavior of the DC motor. If the performance function J_1 tends to zero, then the design variable vector $\bar{p}$ in the estimated DC motor approximates its dynamic behavior to the behavior of the DC motor.

On the other hand, another performance function J_2 is considered in order to decrease the sensitivity of the velocity closed-loop system performance with respect to changes of the design variable vector $\bar{p}$. Hence, the sensitivity of the modelled error with respect to the design variable vector $\frac{\partial J_1}{\partial \bar{p}} \in R^6$ (3) is given as the second performance function to be minimized. Those performance functions are included in $J \in R^2$ as shown in (3).

$$\begin{aligned} J &= [J_1, J_2]^T \\ J_1 &= \int_{t\in\Omega} \bar{e}^T \bar{e} dt \\ J_2 &= \int_{t\in\Omega} \frac{\partial J_1}{\partial \bar{p}}^T \frac{\partial J_1}{\partial \bar{p}} dt. \end{aligned} \tag{3}$$

The sensitivity of the performance function J_1 with respect to the $i-th$ control design parameter vector $\bar{p}_i$ is given by

$$\frac{\partial J_1}{\partial \bar{p}_i} = -2\left(\bar{e}_1 \frac{\partial \bar{x}_1}{\partial \bar{p}_i} + \bar{e}_2 \frac{\partial \bar{x}_2}{\partial \bar{p}_i} + \bar{e}_3 \frac{\partial \bar{x}_3}{\partial \bar{p}_i}\right) \quad \forall i = 1, ..6 \tag{4}$$

where the estimated state sensitivity vector with respect to the design parameter $\bar{p}_i$ (renamed as $\tilde{s}^i_j = \frac{\partial \bar{x}_j}{\partial \bar{p}_i}$ $\forall\, i = 1, \ldots, 6$, $j = 1, 2, 3$) is obtained by deriving the estimated dynamics of the DC motor $\dot{\bar{x}} = f(\bar{x}, u, \bar{p})$ with respect to the design parameter $\bar{p}_i$, as follows:

$$\frac{\partial \dot{\bar{x}}}{\partial \bar{p}_i} = \frac{\partial \bar{f}(\bar{x}, u, \bar{p})}{\partial \bar{p}_i} \tag{5}$$

$$\frac{d}{dt}\frac{\partial \bar{x}}{\partial \bar{p}_i} = \frac{\partial \bar{f}(\bar{x}, u, \bar{p})}{\partial \bar{p}_i} + \frac{\partial \bar{f}(\bar{x}, u, \bar{p})}{\partial \bar{x}}\frac{\partial \bar{x}}{\partial \bar{p}_i} + \frac{\partial \bar{f}(\bar{x}, u, \bar{p})}{\partial u}\frac{\partial u}{\partial \bar{p}_i} \tag{6}$$

Simplifying (6), the sensitivity of the estimated state vector $\tilde{s}^i_j$ is given by solving the differential equations (7)-(12) with zero initial conditions.

$$\frac{d}{dt}\begin{bmatrix}\tilde{s}^1_1\\ \tilde{s}^1_2\\ \tilde{s}^1_3\end{bmatrix} = \begin{bmatrix}\tilde{s}^1_2\\ \bar{p}_2\tilde{s}^1_3 - \bar{p}_1\tilde{s}^1_2 - \bar{x}_2\\ -\bar{p}_3\tilde{s}^1_2 - \bar{p}_4\tilde{s}^1_3\end{bmatrix} \tag{7}$$

$$\frac{d}{dt}\begin{bmatrix}\tilde{s}^2_1\\ \tilde{s}^2_2\\ \tilde{s}^2_3\end{bmatrix} = \begin{bmatrix}\tilde{s}^2_2\\ -\bar{p}_1\tilde{s}^2_2 + \bar{p}_2\tilde{s}^2_3 + \bar{x}_3\\ -\bar{p}_3\tilde{s}^2_2 - \bar{p}_4\tilde{s}^2_3\end{bmatrix} \tag{8}$$

$$\frac{d}{dt}\begin{bmatrix}\tilde{s}^3_1\\ \tilde{s}^3_2\\ \tilde{s}^3_3\end{bmatrix} = \begin{bmatrix}\tilde{s}^3_2\\ \bar{p}_2\tilde{s}^3_3 - \bar{p}_1\tilde{s}^3_2\\ -\bar{p}_3\tilde{s}^3_2 - \bar{p}_4\tilde{s}^3_3 - \bar{x}_2\end{bmatrix} \tag{9}$$

$$\frac{d}{dt}\begin{bmatrix}\tilde{s}^4_1\\ \tilde{s}^4_2\\ \tilde{s}^4_3\end{bmatrix} = \begin{bmatrix}\tilde{s}^4_2\\ \bar{p}_2\tilde{s}^4_3 - \bar{p}_1\tilde{s}^4_2\\ -\bar{p}_3\tilde{s}^4_2 - \bar{p}_4\tilde{s}^4_3 - \bar{x}_3\end{bmatrix} \tag{10}$$

$$\frac{d}{dt}\begin{bmatrix}\tilde{s}^5_1\\ \tilde{s}^5_2\\ \tilde{s}^5_3\end{bmatrix} = \begin{bmatrix}\tilde{s}^5_2\\ \bar{p}_2\tilde{s}^5_3 - \bar{p}_1\tilde{s}^5_2\\ -\bar{p}_3\tilde{s}^5_2 - \bar{p}_4\tilde{s}^5_3 + u\end{bmatrix} \tag{11}$$

$$\frac{d}{dt}\begin{bmatrix}\tilde{s}^6_1\\ \tilde{s}^6_2\\ \tilde{s}^6_3\end{bmatrix} = \begin{bmatrix}\tilde{s}^6_2\\ -\bar{p}_1\tilde{s}^6_2 + \bar{p}_2\tilde{s}^6_3 - 1\\ -\bar{p}_3\tilde{s}^6_2 - \bar{p}_4\tilde{s}^6_3\end{bmatrix} \tag{12}$$

D. CONSTRAINTS

The constraints involve the differential equations describing both the dynamic model of the DC motor and the estimated one. These constraints provide the behavior of the DC motor output states with different input values. Those dynamic constraints are given in (13) and (14).

$$\dot{x}(t) = f(x(t), u(t), p) \tag{13}$$

$$\dot{\bar{x}}(t) = f(\bar{x}(t), u(t), \bar{p})|_{t\in\Omega} \tag{14}$$

The control signal bounds are included as inequality constraints (15)-(16). Those constraints limit the applied voltage to the DC motor.

$$g_1 : u(t_n) - u_{MAX} \leq 0|_{t_n\in\Omega} \tag{15}$$

$$g_2 : u_{MIN} - u(t_n) \leq 0|_{t_n\in\Omega} \tag{16}$$

The last inequality constraints involve the design variable vector bounds given in (17)-(18), where $\bar{p}_{MIN}$ and $\bar{p}_{MAX}$ are the lower and upper limits in the design variable vector $\bar{p}$.

$$g_3 : \bar{p} - \bar{p}_{MAX} \leq 0 \tag{17}$$

$$g_4 : \bar{p}_{MIN} - \bar{p} \leq 0 \tag{18}$$

E. ON-LINE DYNAMIC OPTIMIZATION PROBLEM STATEMENT

Assuming that the parameter vector $p(t)$ in the DC motor changes with respect to the current time variable $t \in [0, \Delta t, 2\Delta t, \ldots, t_f]$, the on-line dynamic optimization problem consists in finding the control design variable vector $\bar{p}(t)$ at each time space Ω to track a desired velocity under the effect of DC motor parameter uncertainties by minimizing the modelled error and its sensitivity subject to DC motor dynamics, the estimated one, the limits in the control signal and bounds in the design variable vector $\bar{p}$. The general problem of the on-line dynamic optimization is formulated

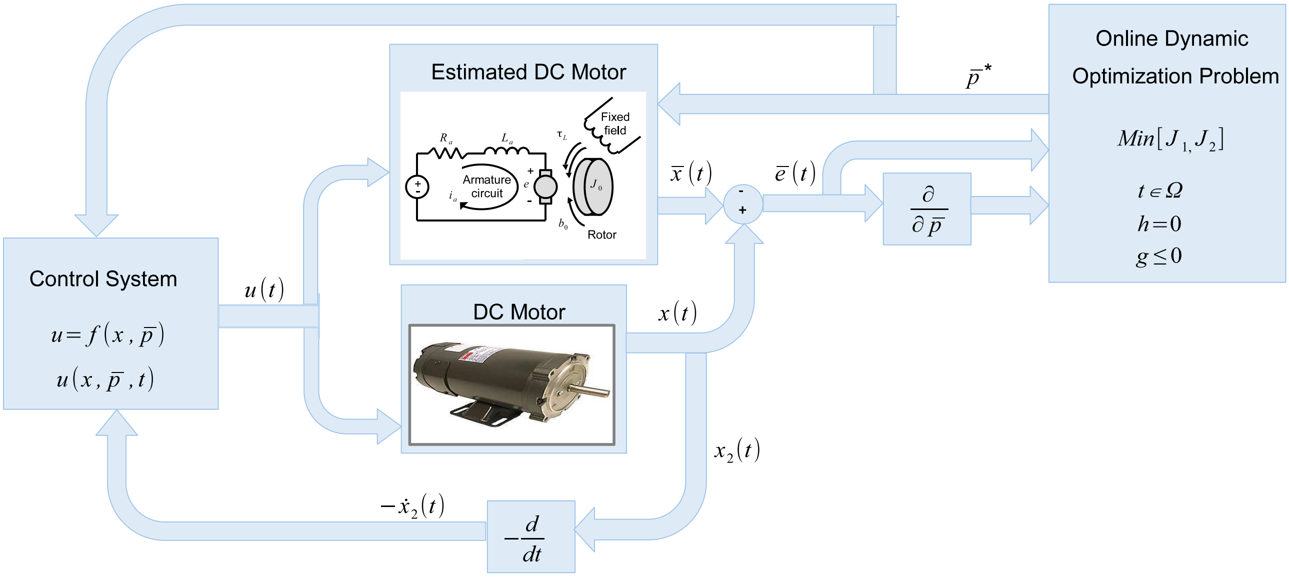

FIGURE 1. Schematic diagram of the dynamic optimization process for the on-line control parameter estimation.

as in (19)-(25).

$$\underset{\bar{p}^*}{Min} J \tag{19}$$

$$\text{Subject to: } \frac{dx}{dt} = f(x(t), u(t), p), \quad x(0) = [0, 0, 0]^T \tag{20}$$

$$\frac{d\bar{x}}{dt} = f(\bar{x}(t), u(t), \bar{p})\Big|_{t \in \Omega}, \quad \bar{x}(t_1) = x(t_1) \tag{21}$$

$$g_1(x(t_n), \bar{p}) \leq 0 \tag{22}$$

$$g_2(x(t_n), \bar{p}) \leq 0 \tag{23}$$

$$g_3(\bar{p}) \leq 0 \tag{24}$$

$$g_4(\bar{p}) \leq 0 \tag{25}$$

A schematic diagram of the on-line dynamic optimization problem of the closed-loop system for the on-line tuning of the velocity control parameter vector is shown in Fig. 1.

III. MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION ALGORITHM IN ON-LINE OPTIMIZATION METHOD

In the dynamic optimization problem (DOP), required in the on-line optimization method, the optimum solutions change continuously over time. Then, the optimization technique should be able to follow the optimum solution in the dynamic environment and in this paper a selection process based on both the Pareto dominance and the set of feasibility rules; the storage of non-dominated solutions in a memory and a preference handling mechanism are incorporated into the differential evolution algorithm.

A. FUNDAMENTALS OF MULTI-OBJECTIVE OPTIMIZATION

The fundamental concepts of multi-objective optimization are detailed below [21].

Definition 3: Let $\bar{p} \in \Omega$ be all design space solutions, the feasible region is represented as $\hat{\Omega} = \{\bar{p} \in \Omega \mid g(x, \bar{p}) < 0, h(x, \bar{p}) = 0\}$

Definition 4: Dominance: A vector $u = [u_1, \ldots, u_k]^T$ dominates a vector $v = [v_1, \ldots, v_k]^T$ (denoted as $u \preceq v$), if and only if, u is smaller than v, i. e., $\forall i \in \{1, \ldots, k\}$, $u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$.

Definition 5: Pareto optimality: A solution $\bar{p} \in \hat{\Omega}$ is a Pareto optimum in $\hat{\Omega}$, if and only if, there is not $\bar{p}' \in \hat{\Omega}$ which $v = J(\bar{p}') = [J_1(\bar{p}'), \ldots, J_i(\bar{p}')]^T$ dominates $u = J(\bar{p}) = [J_1(\bar{p}), \ldots, J_i(\bar{p})]^T$.

Definition 6: The Pareto optimum set $\mathfrak{P}^*$ is defined as:

$$\mathfrak{P}^* := \{\bar{p} \in \hat{\Omega} \mid \nexists \bar{p}' \in \hat{\Omega}, \Phi(\bar{p}') \preceq \Phi(\bar{p})\}$$

Each solution of the Pareto optimum set is called non-dominated solution.

Definition 7: The Pareto front $\mathfrak{PF}$ is defined as:

$$\mathfrak{PF} := \{u = \Phi(\bar{p}) \mid \bar{p} \in \mathfrak{P}^*\}$$

B. OVERVIEW OF DIFFERENTIAL EVOLUTION ALGORITHM

Differential evolution (DE) algorithm, first proposed by Kenneth Price and Rainer Storn [22], is one of the most used global optimization methods simple to implement, easy to use, reliable and fast.

The original DE version is not able to efficiently handle the dynamic constraint optimization problem. In this paper some modifications in the DE algorithm are proposed in order to enhance the explorative and exploitative search to locate the changing optimum in the dynamic environment of the tuning approach based on the on-line optimization method. Those modifications can be numbered as: *i*) The inclusion of a constraint handling technique based on the set of feasibility rules and the Pareto dominance to guide the search to the feasible region $\hat{\Omega}$, *ii*) The use of an external memory to keep the non-dominated solutions through generations to provide them in the initial search of the next time interval into the dynamic environment, and *iii*) The use of preference handling mechanism based on L_p-metrics.

The DE algorithm known as Modified DE with Induced Initial Population based on Non-Dominated Solution through a Memory (MDE-IIP-NDSM) requires an initial population at the beginning of the algorithm. NP individuals are generated and stored in the population matrix $\mathbf{X}_G = [\vec{x}_G^i, \ldots, \vec{x}_G^{NP}]^T \in R^{NP \times D}$ called population of parents. The initial population of individuals is randomly generated if the external memory (non-dominated solutions) is empty. If the external memory is not empty, the individuals of the external memory are incorporated into the initial population and the remaining individuals in the population are randomly generated. At each generation G, the individuals $\vec{x}_G^i \in R^{1 \times D}$ in the population $\mathbf{X}_G$, mutate and recombine to generate NP child individuals $\vec{u}_G^i \in R^{1 \times D}$. At the last stage, children compete with their parents. Based on their performances, the apt individuals survive and conform the population of parents to the next generation $G+1$. The constraint handling technique based on the set of feasibility rules [23] and the Pareto dominance is included into the selection process in order to efficiently explore the search space in the multi-objective constrained optimization problem. At each generation the non-dominated solutions are stored in an external memory. In the new generation $G+1$ the same processes

Algorithm 1 Pseudo-Code of the DE Algorithm for Multi-Objective Dynamic Optimization. *randint*$(1, D)$ Is a Uniformly Distributed Integer Random Number Generator

Initialization: time $t = 0$, sampling time Δt, back time interval Δw, initial condition $x(t = 0)$, initial control signal $u(t = 0) = 20$, control design vector $\bar{p}(t) = \mathbf{0}\ \forall t \in [0, \Delta w]$ and external memory $EM = \emptyset$.
while $t < t_f$ **do**
  Compute the behavior of the real DC motor behavior at the present time t (definition 2). Then, solve by Euler method the dynamics of the DC motor (1) at the present time t: $x(t + \Delta t) = x(t) + f(x(t), u(t), p(t))\,\Delta t$.
  if $t < \Delta w$ **then**
    Set the control signal $u(t + \Delta t) = 20$.
  else
    $G = 0$
    if $EM = \emptyset$ **then**
      $\mathbf{X}_G = \{\bar{p}(t) \bigcup \textit{Random individuals}\}$
    else
      $\mathbf{X}_G = \{EM \bigcup \textit{Random individuals}\}$
    end if
    Solve by Euler method the dynamics of the estimated DC motor (definition 2) in the time space $\Omega = \{t' \in R \mid t' \in [t - \Delta w, t]\}$ per each individual in the population: $\bar{x}(t' + \Delta t) = \bar{x}(t') + f(\bar{x}(t'), u(t), \bar{p})\,\Delta t$. Hence, considers $\bar{p} = \vec{x}_G^i \in \mathbf{X}_G$.
    Evaluate $J(\vec{x}_G^i)$, $g(\vec{x}_G^i)$, $\forall i = 1, \ldots, NP$.
    while $G < G_{Max}$ **do**
      for $i = 1$ to NP **do**
        Select three individuals $\{r_1 \neq r_2 \neq r_3 \neq i\} \in \mathbf{X}_G$.
        j_{rand} =randint$(1, D)$
        for $j = 1$ to D **do**
          Use the mutation and crossover operators according to DE variants to generate $\vec{u}_{G+1}^i$.
        end for
        Evaluate $J(\vec{u}_{G+1}^i)$, $g(\vec{u}_{G+1}^i)$.
        Select the best individual between $\vec{u}_{G+1}^i$ and $\vec{x}_G^i$.
      end for
      Keep the non-dominated solutions of $EM \bigcup \mathbf{X}_{G+1}$ and store them in EM.
      $G = G + 1$
    end while
    Use the preference handling mechanism to find the best compromise solution.
    Store the best compromise in $\bar{p}(t + \Delta t)$.
    Set the control signal by computing $u(t + \Delta t) = \tilde{f}(x(t), \bar{p}(t + \Delta t))$ in (2).
  end if
  $t = t + \Delta t$
end while

are repeated (mutation, recombination, selection and storage) until the maximum number of generations G_{Max} is reached. The final stage involves the decision maker. A posteriori preference handling technique [24] is included to provide the best compromise from the Pareto front and to give the appropriate control design variable vector to the next sample interval.

In the dynamic environment, the solution of the multi-objective dynamic optimization problem for the tuning approach based on the on-line optimization method must be given at each time interval. As was commented previously, the general schematic diagram is given in Fig. 1. A complete pseudo-code to implement the modified DE algorithm in the velocity control of the DC motor is shown in Algorithm 1.

1) MUTATION AND CROSSOVER PROCESSES

There are different variants of DE which are known with the nomenclature DE/X/Y/Z. In such nomenclature *"DE"* means differential evolution, *"X"* and *"Y"* indicates the way to choose the base vector $\vec{x}_G^{r0}$ and the number of difference vectors in the mutation process, respectively, and *"Z"* refers to the type of the crossover operator. The selection of three individuals in the mutation process depends on the DE variants and those are identified by an index r_1, r_2, and r_3. The index r_1 is called base vector index and the indexes r_2 and r_3 the difference vector indexes.

The performance of the DE variant depends on the problem at hand as is studied in [25] for benchmark problems and also for a specific real world problem in [26]. Based on their performance to find suitable solutions, several DE variants [27], [28] are selected in order to compare their performances to solve the on-line dynamic optimization problem. Those are: two variants with binomial discrete recombination operator $DE/Rand/1/Bin$ and $DE/Best/1/Bin$, two variants with exponential discrete recombination operator $DE/Rand/1/Exp$ and $DE/Best/1/Exp$, two variants with arithmetic recombination $DE/Current\ to\ Rand/1$ and $DE/Current\ to\ Best/1$, two variants with a combined discrete-arithmetic recombination $DE/Current\ to\ Rand/1/Bin$ and $DE/Current\ to\ Best/1/Bin$. Lastly, a variant which incorporates objective function information to the mutation and recombination operators $DE/Rand/2/Dir$.

The pseudo-code of the mutation and crossover processes in those variants are shown in (26), as shown at the bottom of this page, where F and K are the scale (mutation) factor, CR is the crossover rate and the term "best" refers to the best individual in the population. These processes are included in line 20 of Algorithm 1.

2) PROPOSED SELECTION PROCESS

In the selection process the child $\vec{u}_G^{i+1}$ and its corresponding father $\vec{x}_G^i$ must compete to pass to the next generation. As the problem at hand is a constrained multi-objective problem, the Pareto dominance and the set of feasibility rules [23] are merged to select the best individual between them (elitist selection). The proposed method uses the following criteria:

- Between two feasible solutions, the one which dominates the other is selected. If there is not an individual who dominates the other, each individual has a probability of 50% to be selected.
- Between two infeasible solutions, the one with smaller constraint violation is selected.
- Feasible solution is preferred versus an infeasible one.

$$
\begin{aligned}
&\mathbf{DE/Rand/1/Bin}: u_j^i = \begin{cases} v_j^i = x_j^{r_1} + F(x_j^{r_2} - x_j^{r_3}) & \text{if } \mathrm{rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \\
&\mathbf{DE/Rand/1/Exp}: u_j^i = \begin{cases} v_j^i = x_j^{r_1} + F(x_j^{r_2} - x_j^{r_3}) & \text{from } \mathrm{rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \\
&\mathbf{DE/Best/1/Bin}: u_j^i = \begin{cases} v_j^i = x_j^{best} + F(x_j^{r_2} - x_j^{r_3}) & \text{if } \mathrm{rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \\
&\mathbf{DE/Best/1/Exp}: u_j^i = \begin{cases} v_j^i = x_j^{best} + F(x_j^{r_2} - x_j^{r_3}) & \text{from } \mathrm{rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \\
&\mathbf{DE/Current-to-Rand/1}: \vec{u}^i = v_j^i = \vec{x}^i + K(\vec{x}^{r_3} - \vec{x}^i) + F(\vec{x}^{r_1} - \vec{x}^{r_2}) \\
&\mathbf{DE/Current-to-Best/1}: \vec{u}^i = v_j^i = \vec{x}^i + K(\vec{x}^{best} - \vec{x}^i) + F(\vec{x}^{r_1} - \vec{x}^{r_2}) \\
&\mathbf{DE/Current-to-Rand/1/Bin}: u_j^i = \begin{cases} v_j^i = x_j^i + K(x_j^{r_3} - x_j^i) + F(x_j^{r_1} - x_j^{r_2}) & \text{if } \mathrm{rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \\
&\mathbf{DE/Current-to-Best/1/Bin}: u_j^i = \begin{cases} v_j^i = x_j^i + K(x_j^{best} - x_j^i) + F(x_j^{r_1} - x_j^{r_2}) & \text{if } \mathrm{rand}_j(0,1) < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \\
&\mathbf{Rand/2/Dir}: \vec{u}^i = v_j^i = \vec{w}^1 + \frac{F}{2}(\vec{w}^1 - \vec{w}^2 + \vec{w}^3 - \vec{w}^4)\ where\ f(\vec{w}^1) < f(\vec{w}^2)\ and\ f(\vec{w}^3) < f(\vec{w}^4)
\end{aligned} \tag{26}
$$

Algorithm 2 Pseudo-Code of the Selection Process in the DE Algorithm. $rand(0, 1)$ Is a Uniformly Distributed Random Number Generator

if $\vec{x}_G^i$ & $\vec{u}_{G+1}^i \in \hat{\Omega}$ **then**
  if $\vec{x}_G^i \preceq \vec{u}_{G+1}^i$ **then**
    $\vec{x}_{G+1}^i = \vec{x}_{G+1}^i$
  else if $\vec{u}_{G+1}^i \preceq \vec{x}_G^i$ **then**
    $\vec{x}_{G+1}^i = \vec{u}_G^i$
  else if $rand(0, 1) \leq 0.5$ **then**
    $\vec{x}_{G+1}^i = \vec{u}_{G+1}^i$
  else
    $\vec{x}_{G+1}^i = \vec{x}_G^i$
  end if
else if $\vec{x}_G^i \in \hat{\Omega}$ **then**
  $\vec{x}_{G+1}^i = \vec{x}_G^i$
else if $\vec{u}_{G+1}^i \in \hat{\Omega}$ **then**
  $\vec{x}_{G+1}^i = \vec{u}_{G+1}^i$
else if $\sum_{k=1}^{4} max(0, sign(g_k(\vec{x}_G^i))) < \sum_{k=1}^{4} max(0, sign(g_k(\vec{u}_{G+1}^i)))$ **then**
  $\vec{x}_{G+1}^i = \vec{x}_G^i$
else
  $\vec{x}_{G+1}^i = \vec{u}_{G+1}^i$
end if

Algorithm 2 represents the selection process included in line 23 of Algorithm 1.

3) PREFERENCE HANDLING MECHANISM

One of the main issues in the multi-objective evolutionary algorithms is the multi-criteria decision making procedure required in the decision maker (DM) [24]. Recently, the need to integrate the multi-objective problem, the optimization process and the multi-criteria decision making procedure is discussed for the *off-line* control tuning problem [2], [9]. Nevertheless, the decision maker is more critical in the dynamic environment of the *on-line* control tuning problem since the closed-loop behavior for the next time interval depends on the selection of the most appropriate control design parameters, i.e., the control design parameters change over time (Pareto solutions dynamically change their values through the evolution of time) and a "bad" selection may even unstabilize the closed-loop system for the next time intervals. Hence, in this paper, the ideal control design parameter vector to the next time interval is selected from individuals in the last generation based on Compromising Programming [29]. The most common Compromising Programming model is given by the distance measure of the family of L_p−metrics. Then, the distance measure L_p (27) is used in this paper to select the best compromise among individuals in the last generation in order to include it in the closed-loop system for the next time interval. The best compromise is given by the individual that presents a smaller L_p distance.

$$L_p(\bar{p}) = \left[\sum_{i=1}^{2} w_i^{\eta} \left| \frac{J_i(\bar{p}) - J_i^*}{J_{iMax} - J_i^*} \right|^{\eta} \right]^{1/\eta} \qquad (27)$$

where $J_i(\bar{p})$ is the $i-th$ performance function evaluated with the design variable vector $\bar{p}$, $\eta = 2$ is the type of distance, $J_i^* = 0$ and J_{iMax} are the Utopian and the Nadir performance function values, respectively. The weights $w_1 = 0.8$ and $w_2 = 0.2$ are chosen for the modelled error and its sensitivity. Those weights were selected based on a series of trial an error procedures where the trade-off between performance functions is considered.

In addition, in order to deal the noise in the control design parameter vector $\bar{p}(t)$ obtained by the optimization technique at each integration time Δt, the discrete-time implementation of a simple Resistor-Capacitor (RC) low-pass filter (28) is included in $\bar{p}(t)$.

$$\bar{p}(t + \Delta t) = \alpha \bar{p}(t) + (1 - \alpha)\bar{p}(t - \Delta t) \qquad (28)$$

where the smoothing factor is set as $\alpha = 0.5$ to give the filter time constant equal to Δt.

Then, the preference handling mechanism is included in line 28 and 29 of Algorithm 1.

4) EXTERNAL MEMORY

In a dynamic environment, the Pareto solutions dynamically change their values through the evolution of time [30]. In order to produce improved solutions at each sampling time of the closed-loop system, an external memory (EM) is added into the DE algorithm. It is important to point out that the external memory stores the non-dominated solutions through the evolution of time, hence the Pareto front obtained at each generation G of the DE algorithm are filtered with the Pareto front obtained from the previous generation $G - 1$ at each time interval. The filtering process removes from these two populations those solutions that are dominated by at least

TABLE 2. Experiment 1: Performance of MDE-IIP-NDSM variants. The best variant is *Current to Rand*/1 according with $mean(|\dot{e}|) = 0.1339$.

MDE-IIP-NDSM variant	F	K	$mean(\|\dot{e}\|)$	$std(\|\dot{e}\|)$	$Best$	$Worst$	$mean(NDS)$
$Rand/1/Bin$	0.5	–	0.1913	0.2839	0.1653	0.2201	3.681
$Best/1/Bin$	0.6	–	0.1645	0.3237	0.1356	0.1997	4.307
$Rand/1/Exp$	0.4	–	0.1860	0.2990	0.1554	0.2384	3.725
$Best/1/Exp$	0.6	–	0.1806	0.3208	0.1559	0.2172	2.551
$Current\ to\ Best/1/Bin$	0.5	0.5	0.1501	0.3045	0.1263	0.2003	3.547
$Current\ to\ Rand/1/Bin$	0.5	0.4	0.1623	0.2477	0.1440	0.1957	3.575
$Current\ to\ Best/1$	0.5	0.5	0.1710	0.3435	0.1491	0.2176	11.14
Current to Rand/1	0.5	0.4	**0.1339**	**0.2237**	**0.1168**	**0.1518**	**4.740**
$Rand/2/Dir$	0.5	–	0.6682	2.7990	0.2922	3.8750	8.184

one solution member of the populations in order to form one single set of non-dominated individuals called filtered population. The filtered Pareto front is stored in an external memory together with their corresponding non-dominated solutions. The external memory must be used to obtain the best compromise from the Pareto front given through generations. Then, at the beginning of the DE algorithm in the optimization process for the $n - th$ time interval of the dynamic optimization problem, the non-dominated solutions of the external memory for the previous time interval ($n - 1$ time interval) replace at most the fifty percent of individuals in the initial population and it depends on the size of the external memory. Otherwise, if the number of individuals in the external memory is greater than the half the initial population $NP/2$, then the best compromises (best solutions) must be only selected by using the distance measure L_p in the EM. The EM promotes the explorative/exploitative search in the neighborhood of the non-dominated solutions of the external memory since some individuals in the initial population through the evolution of time are selected from the external memory of the previous time interval and the others are randomly generated. This procedure can be detailed in line $8 - 12$ and 25 of Algorithm 1.

IV. NUMERICAL RESULTS

The numerical results consisted in the performance evaluation of the MDE-IIP-NDSM variants which aim to enhance the explorative/exploitative search to locate the changing optima in the dynamic environment of the tuning approach based on the on-line optimization method. The on-line control tuning problem described in Section II is solved by the MDE-IIP-NDSM variants described in Section III.

Thirty independent simulations of the closed-loop system were made for each algorithm in experiments. All independent simulations were programmed in Matlab with a 3.5 GHz Core $i7$ processor with 32 GB of RAM. The simulation results of the closed-loop system used a simulation time of $t_f = 3\ s$ with an integration time of $\Delta t = 5\ ms$. The same parameter values were used in the simulation of the closed-loop system. For MDE-IIP-NDSM variants on the on-line dynamic optimization problem, the population of $NP = 25$ individuals is selected with a maximum generation of $G_{Max} = 50$, a crossover rate of $CR = 0.8$, a time interval of $\Delta w = 50\ ms$ and the bounds of the design variable vector as $\bar{p}_{MIN} = [0, 0, 0, 0, 0, -150]^T$, $\bar{p}_{MAX} = [2, 1200, 5, 100, 10, 150]^T$. The F and K parameters are displayed in the second and third column of Table 2. Those commented algorithm parameters are used for all DE variant comparison and were obtained by testing with different combinations of values such that the values with the best performance were considered in this paper. In the case of the maximum bound vector $\bar{p}_{MAX}$ of design variables, these limits are selected according to specific porcentaje of variation from the nominal parameters given in Table 1, assuming that the uncertainties are not known. The minimum bound vector $\bar{p}_{MIN}$ clearly indicates that the parameters $\bar{p}$ can not be negative. Only $\bar{p}_6$ can have either positive or negative values due to the load direction added to the shaft of the motor.

Four different experiments were carried out. Experiment 1 includes the performance of the MDE-IIP-NDSM variants to solve the on-line control tuning problem. The next experiments were given in order to compare the performance of the MDE-IIP-NDSM variants. In Experiment 2 the external memory in MDE-IIP-NDSM variants is removed. In Experiment 3, the external memory is removed and instead of using the distance measure L_p in the preference handling of MDE-IIP-NDSM variants (line 28 and 29 in Algorithm 1), the individual in the last generation with the lowest modelled error J_1 is selected (i.e., the individual in one extreme of the Pareto front for the last generation). The fourth experiment consists in evaluating the performance of the multi-objective meta-heuristic algorithms: Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Particle Swarm Optimization (PSO). The population size and the maximum generation in NSGAII and PSO are chosen as previously commented, that is, $G_{Max} = 50$, $NP = 25$. The NSGAII and PSO parameters are obtained from [31] and [32], respectively. In all experiments continuous dynamic changes in the nominal DC motor parameters and a discontinuous dynamic load as shown in Table 1, are set in order to provide uncertainties in the dynamic environment.

Tables 2-5 present the numerical results of the experiments. The first column represents the DE variant involved in the proposed tuning approach based on the on-line optimization method. The next two columns are the selected DE parameters for F and K. The performance of the algorithm is

TABLE 3. Experiment 2: Performance of MDE-IIP-NDSM variants without the use of the external memory and with the use of the preference handling mechanism. The best variant is *DE/Current to Best/1/Bin* according with *mean*($|\dot{e}|$) = 0.2630.

DE variant	$mean(\|\dot{e}\|)$	$std(\|\dot{e}\|)$	$Best$	$Worst$
$Rand/1/Bin$	0.6815	0.9745	0.5602	0.8034
$Best/1/Bin$	0.2923	0.6928	0.2231	0.4129
$Rand/1/Exp$	0.7963	1.1730	0.6397	0.9213
$Best/1/Exp$	0.5535	0.9235	0.4461	0.7194
Current to Best/1/Bin	**0.2630**	**0.5375**	**0.2198**	**0.3343**
$Current\ to\ Rand/1/Bin$	0.5098	0.6967	0.4163	0.6169
$Current\ to\ Best/1$	0.3057	0.5920	0.2368	0.3829
$Current\ to\ Rand/1$	0.5079	0.7893	0.3868	0.6283
$Rand/2/Dir$	2.4640	3.4690	2.1800	3.0710

TABLE 4. Experiment 3: Performance of MDE-IIP-NDSM variants without the use of both the external memory and the preference handling mechanism. The selection of the individual that pass to the next generation is according to the best performance in the function J_1. The best variant is *DE/Current to Best/1/Bin* with *mean*($|\dot{e}|$) = 0.2544.

DE variant	$mean(\|\dot{e}\|)$	$std(\|\dot{e}\|)$	$Best$	$Worst$
$Rand/1/Bin$	0.6587	0.8867	0.5665	0.8390
$Best/1/Bin$	0.3120	0.7622	0.2325	0.4846
$Rand/1/Exp$	0.7879	1.1620	0.5969	0.9510
$Best/1/Exp$	0.5509	0.9632	0.4119	0.7148
Current to Best/1/Bin	**0.2544**	**0.4575**	**0.1994**	**0.3477**
$Current\ to\ Rand/1/Bin$	0.2667	0.5173	0.7699	0.4100
$Current\ to\ Best/1$	0.3032	0.5704	0.2468	0.4212
$Current\ to\ Rand/1$	0.5091	0.7779	0.4257	0.6263
$Rand/2/Dir$	3.7050	12.0500	2.0020	40.960

TABLE 5. Experiment 4: Performance of NSGAII and PSO. The experiment 4 does not reduce the velocity error than other experiments.

	$mean(\|\dot{e}\|)$	$std(\|\dot{e}\|)$	$Best$	$Worst$
$NSGAII$	0.7096	1.122	0.5661	0.966
PSO	0.2138	0.4569	0.1172	0.3315

evaluated according to the velocity error generated by the DC motor in the time interval $t \in [0.1, 3]$, where the terms $mean(|\dot{e}|)$ and $std(|\dot{e}|)$ are the mean and the standard deviation of the velocity error in thirty independent runs. In the same manner, the terms *Best* and *Worst* are referred as the best and worst velocity error presented through runs. The mean number of non-dominated solutions in the external memory for all runs are calculated in *mean*(*NDS*).

Furthermore, thirty fronts obtained by runs per each variant are filtered into one front, and filtered fronts for each variant are shown in Fig. 2 in order to visualize the Pareto front obtained in each MDE-IIP-NDSM variant.

From the summary of results presented in Table 2 for the MDE-IIP-NDSM variants (Experiment 1) and their filtered Pareto fronts in Fig. 2, different findings were observed:

1) The results indicate that the arithmetic recombination with a random individual given in *Current to Rand/1*

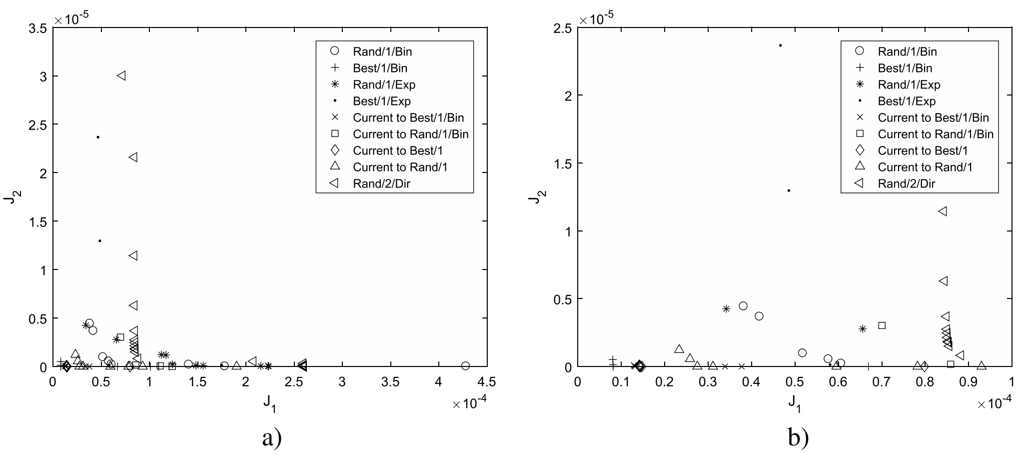

FIGURE 2. Filtered Pareto front for each MDE-IIP-NDSM variants. a) Resulting Pareto front. b) Zoom.

is the most convenient in the on-line control tuning problem because it explores and exploits the individuals such that the smallest velocity error is given among the MDE-IIP-NDSM variants. Moreover, this variant provides the best control gains because the motor velocity error will be in the smallest error interval of $[0.1168, 0.1518] rad/s$ among all experiments (see *Best* and *Worst* columns of Tables 2-5 and also the standard deviation is reduced). It is important to note that the inclusion of the best individual into the arithmetic recombination (see *Current to Best*/1) promote a premature convergence of the control tuning such that the velocity regulation is compromised.

2) Variants with a combined discrete-arithmetic recombination are the other two promising MDE-IIP-NDSM variants. Such variants are *Current to Best*/1/*Bin* and *Current to Rand*/1/*Bin*.
3) The worst MDE-IIP-NDSM variants are attributed to the variant which incorporates objective function information in the mutation and recombination operators, i.e., the MDE-IIP-NDSM variant *Rand*/2/*Dir*.
4) The two variants which find more non-dominated solutions are the *Current to Best*/1 and *Rand*/2/*Dir*. It is important to comment that the non-dominated solutions in the external memory change through the simulation time of the closed-loop system. These changes adapt the search in the dynamic environment and the obtained solution may be benefited according to the MDE-IIP-NDSM variant. In the case of *Rand*/2/*Dir* the unfit solutions are extremely exploited such that several

TABLE 6. Wilcoxon signed ranks test results. All MDE-IIP-NDSM variants of experiment 1 show an improvement over Experiment 2.

Experiment 1 versus Experiment 2	R_+	R_-	$p-value$
MDE-IIP-NDSM/Rand/1/Bin vs $DE/Rand/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Best/1/Bin vs $DE/Best/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Rand/1/Exp vs $DE/Rand/1/Exp$	465	0	1.8E-9
MDE-IIP-NDSM/Best/1/Exp vs $DE/Best/1/Exp$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Rand/1/Bin vs $DE/Current\ to\ Rand/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Best/1/Bin vs $DE/Current\ to\ Best/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Rand/1 vs $DE/Current\ to\ Rand/1$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Best/1 vs $DE/Current\ to\ Best/1$	465	0	1.8E-9
MDE-IIP-NDSM/Rand/2/Dir vs $DE/Rand/2/Dir$	463	2	5.5E-9

TABLE 7. Wilcoxon signed ranks test results. All MDE-IIP-NDSM variants of experiment 1 show an improvement over Experiment 3.

Experiment 1 versus Experiment 3	R_+	R_-	$p-value$
MDE-IIP-NDSM/Rand/1/Bin vs $DE/Rand/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Best/1/Bin vs $DE/Best/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Rand/1/Exp vs $DE/Rand/1/Exp$	465	0	1.8E-9
MDE-IIP-NDSM/Best/1/Exp vs $DE/Best/1/Exp$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Rand/1/Bin vs $DE/Current\ to\ Rand/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Best/1/Bin vs $DE/Current\ to\ Best/1/Bin$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Rand/1 vs $DE/Current\ to\ Rand/1$	465	0	1.8E-9
MDE-IIP-NDSM/Current to Best/1 vs $DE/Current\ to\ Best/1$	465	0	1.8E-9
MDE-IIP-NDSM/Rand/2/Dir vs $DE/Rand/2/Dir$	458	2	3.5E-8

TABLE 8. Wilcoxon signed ranks test results. All MDE-IIP-NDSM variants in Experiment 1 versus the NSGAII presents a significant improvement. The 77.77% of the comparisons show improvement over PSO. *MDE − IIP − NDSM*/*Rand*/1/*Bin* does not present an improvement over PSO. PSO is better than *MDE − IIP − NDSM*/*Rand*/2/*Dir*.

Experiment 1 vs Experiment 4	R_+	R_-	$p-value$
MDE-IIP-NDSM/Rand/1/Bin vs PSO	304	161	0.1459
MDE-IIP-NDSM/Rand/1/Bin vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Best/1/Bin vs PSO	387	78	0.0009
MDE-IIP-NDSM/Best/1/Bin vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Rand/1/Exp vs PSO	318	147	0.0803
MDE-IIP-NDSM/Rand/1/Exp vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Best/1/Exp vs PSO	338	127	0.0293
MDE-IIP-NDSM/Best/1/Exp vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Current to Rand/1/Bin vs PSO	401	64	0.0002
MDE-IIP-NDSM/Current to Rand/1/Bin vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Current to Best/1/Bin vs PSO	440	25	1.69E-6
MDE-IIP-NDSM/Current to Best/1/Bin vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Current to Rand/1 vs PSO	455	10	8E-8
MDE-IIP-NDSM/Current to Rand/1 vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Current to Best/1 vs PSO	368	97	0.0043
MDE-IIP-NDSM/Current to Best/1 vs NSGAII	465	0	1.8E-9
MDE-IIP-NDSM/Rand/2/Dir vs **PSO**	0	465	1.8E-9
MDE-IIP-NDSM/Rand/2/Dir vs NSGAII	362	103	0.0066

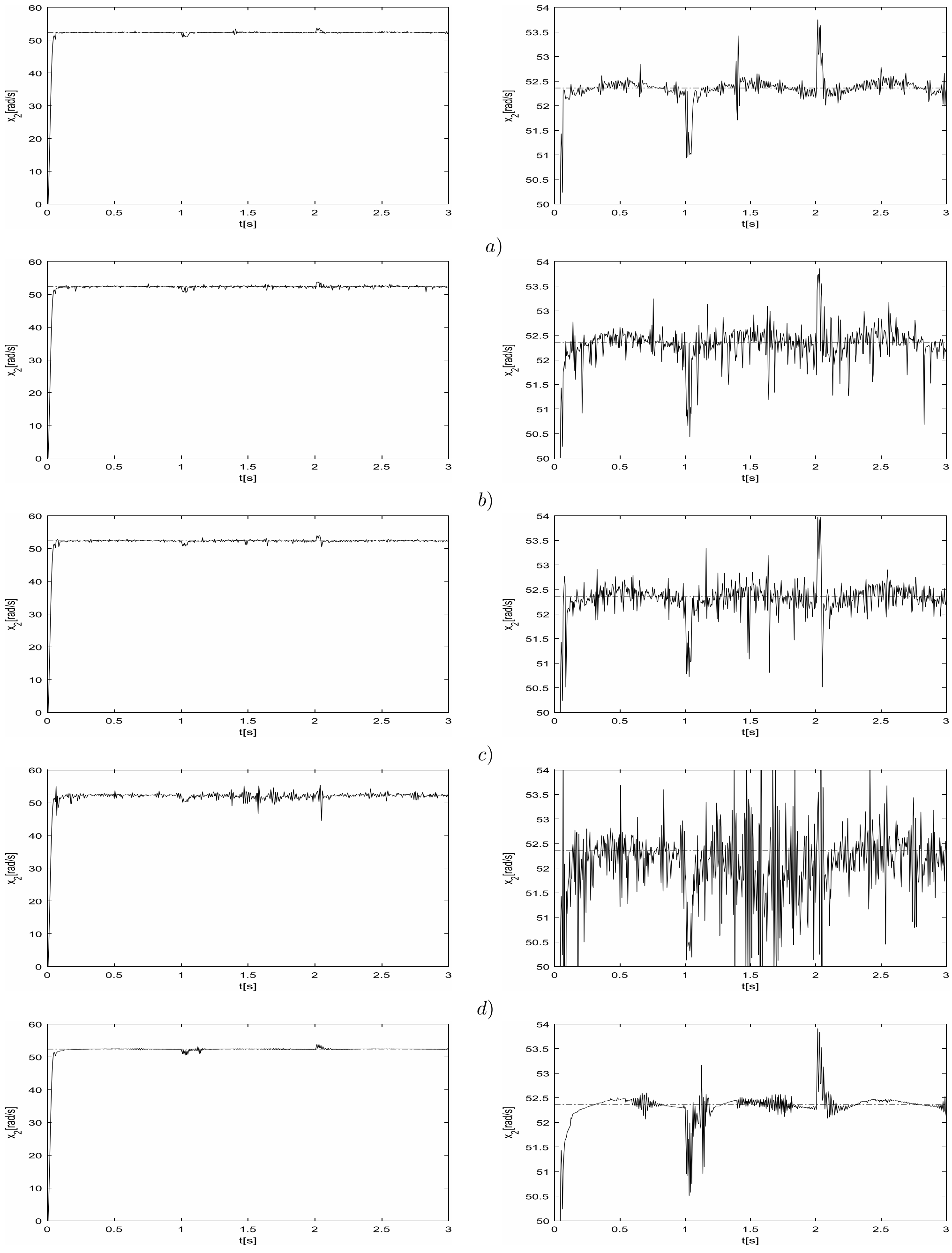

FIGURE 3. Behavior of the velocity profile of the DC motor x_2 given by the best algorithms from experiments. The dotted line denotes the reference signal and the continuous line represents the signal achieved with the on-line control tuning. *a*) Velocity profile x_2 with $MDE-IIP-NDSM/Current\ to\ Rand/1$ of Experiment 1 with a zoom (right figure). *b*) Velocity profile x_2 with $DE/Current\ to\ Best/1/Bin$ of Experiment 2 with a zoom (right figure). c) Velocity profile x_2 with $DE/Current\ to\ Best/1/Bin$ of Experiment 3 with a zoom (right figure). *d*) Velocity profile x_2 with *NSGAII* of Experiment 4 with a zoom (right figure). *e*) Velocity profile x_2 with *PSO* of Experiment 4 with a zoom (right figure).

neighbor solutions are found with a poor performance. On the other hand with $Current\ to\ Best/1$ the solutions are well exploited such that it presents the fifth best performance.

5) From the Filtered Pareto fronts in Fig. 2, it was observed that the four best compromises in the Pareto front are obtained by using $Best/1/Bin$, $Current\ to\ Best/1/Bin$, $Current\ to\ Best/1$ and $Current\ to\ Rand/1$.
6) It is observed that the second performance function referred to the sensitivity decrease the variation in the velocity error as it is observed in the standard deviation

column of Table 2 and these values are the lowest compared with the other algorithms in Tables 3-5. Then, it is confirmed that the sensitivity is related to the robustness of the velocity regulation under parametric uncertainties.

In order to know if the MDE-IIP-NDSM variants given in Experiment 1 improves the corresponding algorithms for the last three experiments, the pairwise comparison Wilcoxon signed-rank test is included to compare the performance with the other experiments. The alternative hypothesis is the set according to a two-sided test which states that for two sets of samples of $mean(|\dot{e}|)$ produced by two different algorithms between experiments, the distribution of $mean(|\dot{e}|)$ of the algorithm in Experiment 1 is different to the one of the other algorithm for the corresponding experiment. Hence, the p-value determines the degree of rejection of the null hypothesis (which asumes that the distributions are the same). A p-value less than 10% or 5% percent provides enough evidence to reject the null hypothesis and therefore accept the alternative hypothesis. Hence, once the alternative hypothesis is accepted, the best algorithm can be obtained by analyzing the rank sums (R_+ and R_-). If R_+ has larger values than R_- then the algorithm in Experiment 1 has better performance than the other, otherwise the opposite happens. Tables 6-8 show the $p-value$ associated with the Wilcoxon signed ranks test results. An analysis of the results with the comparisons among the Experiment 1 in Table 2 with the rest of experiments are presented below:

1) The comparative analysis between the MDE-IIP-NDSM variants in Experiment 1 versus the corresponding DE variants in Experiment 2 by using the Wilcoxon signed-rank test is presented in Table 6. Based on the $p-value$ and the R_+ value, the pairwise statistical comparisons state that with a significance level $\alpha < 0.001$ all MDE-IIP-NDSM variants in Experiment 1 show a significant improvement over variants in Experiment 2. This indicates that the external memory in the on-line tuning of the velocity control promote the exploration and exploitation of the search space such that a significant improvement of the velocity error is presented in the MDE-IIP-NDSM variants of Experiment 1.
2) The comparative analysis between the MDE-IIP-NDSM variants in Experiment 1 versus the corresponding DE variants in Experiment 3 by using the Wilcoxon signed-rank test is presented in Table 7. Based on the $p - value$ and the R_+ value, the pairwise statistical comparisons state that with a significance level $\alpha = 0.001$ the MDE-IIP-NDSM variants have a significant improvement over variants in Experiment 3. Those variants are benefited from the use of the distance measure L_p in the preference handling and the external memory in the on-line tuning of the velocity control. Hence, the importance of weighting the second performance function (the sensitivity) in order to decrease both the velocity error and its variations in the DC motor control is highlighted.

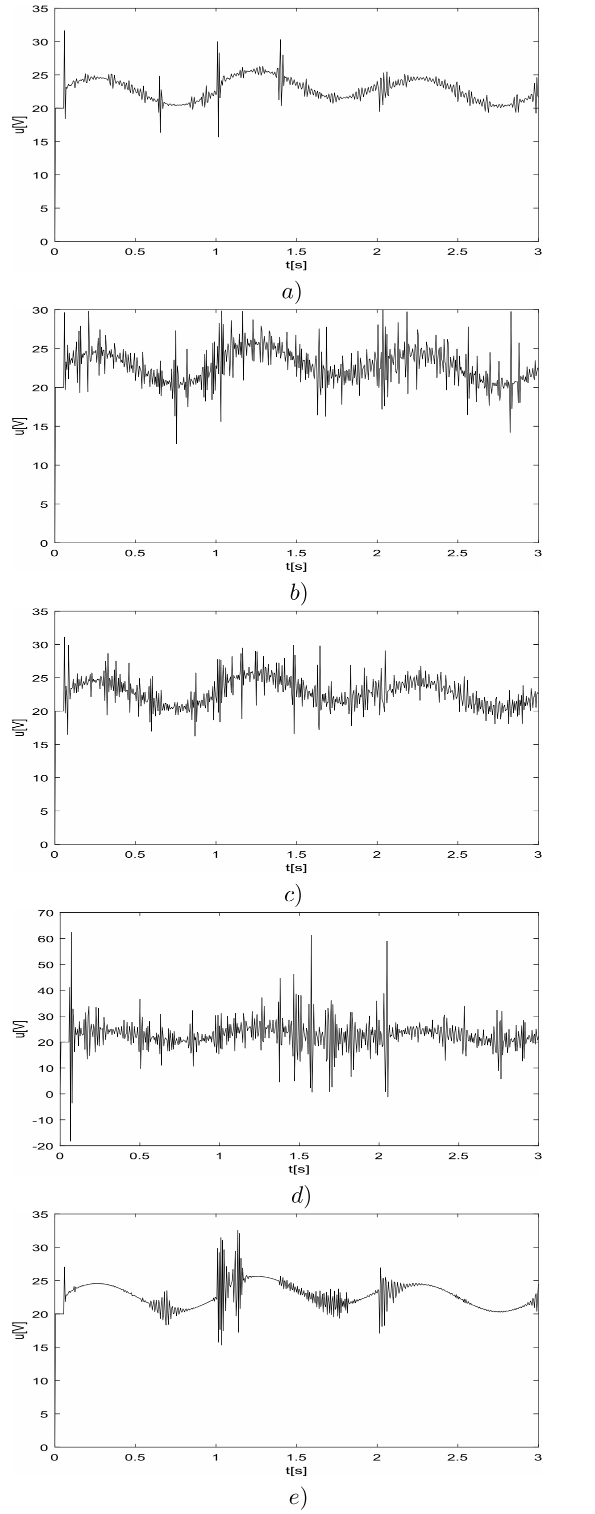

FIGURE 4. **Behavior of the control signal *u* given by the best algorithms from experiments. *a*) Control signal with *MDE − IIP − NDSM*/*Current to Rand*/1 of Experiment 1. *b*) Control signal with *DE*/*Current to Best*/1/*Bin* of Experiment 2. c) Control signal with *DE*/*Current to Best*/1/*Bin* of Experiment 3. *d*) Control signal with *NSGAII* of Experiment 4. *e*) Control signal with *PSO* of Experiment 4.**

3) From the results presented in Experiment 4 (Table 5) and confirmed from the pairwise statistical comparisons shown in Table 8, all MDE-IIP-NDSM variants

in Experiment 1 versus the NSGAII presents a significant improvement with a significance level $\alpha = 0.007$. On the other hand, the 77.77% of algorithms in Experiment 1 presents a significant improvement over PSO and only PSO is better in 11.11% i.e., in $MDE - IIP - NDSM/Rand/2/Dir$. The MDE-IIP-NDSM variant that does not present an improvement over PSO is $MDE - IIP - NDSM/Rand/1/Bin$ (the 11.11% of algorithms). The last result indicates that PSO and $MDE - IIP - NDSM/Rand/1/Bin$ have a similar behavior due to any kind of difference between the set of data is due to chance.

The velocity profile and the control signal of the best performance of algorithms from Tables 2-5 in the on-line control tuning problem of the DC motor in all experiments are shown in Fig. 3 and in Fig 4. It was confirmed that the best velocity regulation performance with the on-line control tuning is given by $MDE - IIP - NDSM/Current\ to\ Rand/1$ in Experiment 1 in spite of the discontinuos dynamic changes of the load τ_L in the time interval $[1, 2]$ and the continuous dynamic changes in the nominal DC motor parameters over all time. Moreover the control signal does not surpass its bounds. Based on Fig. 3, the second best algorithm with respect to the velocity regulation performance is given by PSO. Nevertheless, this performance varies among runs as it was observed in the standard deviation in Table 5 and verify with the above statistical analysis, and hence this algorithm is not reliable for real applications. On the other hand, it was observed that only NSGAII presents control signal above its bound in some time interval of the dynamic simulation. This indicates that NSGAII presents some issues to guide the search to feasible solutions in constrained dynamic environment.

V. CONCLUSIONS

In this paper, a multi-objective on-line tuning optimization method is proposed to adaptively tune the velocity control parameters of the DC motor under parametric uncertainties, where the modelling error and its sensitivity were the proposed conflicting performance functions. The constraint handling technique based on feasible rules and Pareto dominance, the preference handling mechanism based on L_p-metrics and the use of an external memory into the MDEIIP- NDSM deal with the changing optimum solutions in the dynamic environment of the tuning approach.

Three experiments were given in order to compare the behavior of the on-line control tuning of the DC motor with the MDEIIP- NDSM (Experiment 1): One that does not include the use of external memory (Experiment 2), other that does not use neither the distance measure L_p in the preference handling nor the external memory (Experiment 3) and the last one that uses two different multi-objective metaheuristic algorithms (Experiment 4). The general conclusions are:

- The use of the external memory in the MDE-IIP-NDSM variants decreases the velocity error.
- If the preference handling is only based on the performance of the modelling error without considering the trade-off of the sensitivity (i.e., the obtained solution given by the minimum of J_1 in the Pareto front), the MDE-IIP-NDSM variants provide solutions more sensible to uncertainties, such that the changes in the velocity error are increased. Hence, the selection of the suitable trade-offs between the modelling error and its sensitivity is an important issue in the on-line control tuning of the DC motor to increase the precision in the velocity control under parametric uncertainties.
- All MDE-IIP-NDSM variants present a superior behavior with respect to the DE variants of Experiments 2, 3 and the algorithm NSGA-II. The 77.77% MDE-IIP-NDSM variants outperform the performance in the velocity regulation with respect to PSO.

The inclusion of the external memory and the preference handling mechanism promote a better control behavior in the three most representative MDE-IIP-NDSM variants for all experiments: $Current\ to\ Ran/1$, $Current\ to\ Best/1/Bin$ and $Current\ to\ Rand/1/Bin$. Hence, a combined discrete-arithmetic recombination adapt the search space dynamically toward more promising regions (contributing the exploration/exploitation of the most promising areas of the search space), thus increasing the overall search efficacy in the dynamic environment.

REFERENCES

[1] A. E. Ruano, "Intelligent control—The road ahead," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2007, pp. 4442–4443.
[2] G. Reynoso-Meza, J. Sanchis, X. Blasco, and R. Z. Freire, "Evolutionary multi-objective optimisation with preferences for multivariable PI controller tuning," *Expert Syst. Appl.*, vol. 51, pp. 120–133, Jun. 2016.
[3] M. G. Villarreal-Cervantes and J. Alvarez-Gallegos, "Off-line PID control tuning for a planar parallel robot using DE variants," *Expert Syst. Appl.*, vol. 64, pp. 444–454, Dec. 2016.
[4] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing* (Natural Computing Series). Berlin, Germany: Springer-Verlag, 2003.
[5] P. Fleming and R. Purshouse, "Evolutionary algorithms in control systems engineering: A survey," *Control Eng. Pract.*, vol. 10, no. 11, pp. 1223–1241, 2002.
[6] G. Reynoso-Meza, X. Blasco, J. Sanchis, and M. Martínez, "Controller tuning using evolutionary multi-objective optimisation: Current trends and applications," *Control Eng. Pract.*, vol. 28, pp. 58–73, Jul. 2014.
[7] A. E. Ruano, S. S. Ge, T. M. Guerra, F. L. Lewis, J. C. Principe, and M. Colnarič, "Computational intelligence in control," *Annu. Rev. Control*, vol. 38, no. 2, pp. 233–242, 2014.
[8] P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artif. Intell. Rev.*, vol. 39, no. 4, pp. 315–346, Apr. 2013.
[9] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez, "Preference driven multi-objective optimization design procedure for industrial controller tuning," *Inf. Sci.*, vol. 339, pp. 108–131, Apr. 2016.
[10] G. Karer and I. Škrjanc, "Interval-model-based global optimization framework for robust stability and performance of PID controllers," *Appl. Soft Comput.*, vol. 40, pp. 526–543, Mar. 2016.
[11] L. dos Santos Coelho and M. W. Pessôa, "A tuning strategy for multivariable PI and PID controllers using differential evolution combined with chaotic Zaslavskii map," *Expert Syst. Appl.*, vol. 38, no. 11, pp. 13694–13701, 2011.

[12] J. Zhang and S. Yang, "An incremental-PID-controlled particle swarm optimization algorithm for EEG-data-based estimation of operator functional state," *Biomed. Signal Process.*, vol. 14, pp. 272–284, Nov. 2014.

[13] M. M. Sabir and T. Ali, "Optimal PID controller design through swarm intelligence algorithms for sun tracking system," *Appl. Math. Comput.*, vol. 274, pp. 690–699, Feb. 2016.

[14] S. Algoul, M. S. Alam, M. A. Hossain, and M. A. A. Majumder, "Multi-objective optimal chemotherapy control model for cancer treatment," *Med. Biol. Eng. Comput.*, vol. 49, pp. 51–65, Jan. 2011.

[15] M. G. Villarreal-Cervantes, C. A. Cruz-Villar, J. Alvarez-Gallegos, and E. A. Portilla-Flores, "Differential evolution techniques for the structure-control design of a five-bar parallel robot," *Eng. Optim.*, vol. 42, no. 6, pp. 535–565, 2010.

[16] M. G. Villarreal-Cervantes, C. A. Cruz-Villar, and J. Alvarez-Gallegos, "Synergetic structure–control design via a hybrid gradient-evolutionary algorithm," *Optim. Eng.*, vol. 16, no. 3, pp. 511–539, 2014.

[17] T. Jiménez *et al.*, "An auto-tuning PID control system based on genetic algorithms to provide delay guarantees in passive optical networks," *Expert Syst. Appl.*, vol. 42, no. 23, pp. 9211–9220, 2015.

[18] E. V. Kumar, G. S. Raaja, and J. Jerome, "Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter," *Appl. Soft Comput.*, vol. 41, pp. 77–90, Apr. 2016.

[19] P. Mishra, V. Kumar, and K. P. S. Rana, "An online tuned novel nonlinear PI controller for stiction compensation in pneumatic control valves," *ISA Trans.*, vol. 58, pp. 434–445, Sep. 2015.

[20] B. B. Alagoz, A. Ates, and C. Yeroglu, "Auto-tuning of PID controller according to fractional-order reference model approximation for DC rotor control," *Mechatronics*, vol. 23, no. 7, pp. 789–797, Oct. 2013.

[21] C. A. C. Coello, G. B. Lamont, and D. A. van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, NY, USA: Springer, 2007.

[22] K. Price and R. M. Storn, "Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces," Int. Comput. Sci. Inst., Berkley, CA, USA, Tech. Rep. TR-95-012, 1995.

[23] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2, pp. 311–338, 1998.

[24] C. A. C. Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," in *Proc. Congr. Evol. Comput.*, vol. 1. Jul. 2000, pp. 30–37.

[25] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello, "A comparative study of differential evolution variants for global optimization," in *Proc. Genet. Evol. Comput. Conf.*, vol. 1. 2006, pp. 485–492.

[26] M. G. Villarreal-Cervantes, C. A. Cruz-Villar, J. Álvarez-Gallegos, and E. A. Portilla-Flores, "Kinematic dexterity maximization of an omnidirectional wheeled mobile robot: A comparison of metaheuristic and SQP algorithms," *Int. J. Adv. Robot. Syst.*, vol. 9, no. 4, p. 161, 2012.

[27] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Germany: Springer-Verlag, 2005.

[28] V. Feoktistov and S. Janaqi, "Generalization of the strategies in differential evolution," in *Proc. 18th Int. Parallel Distrib. Process. Symp.*, Apr. 2004, pp. 165–171.

[29] E. Ballestero, "Compromise programming: A utility-based linear-quadratic composite metric from the trade-off between achievement and balanced (non-corner) solutions," *Eur. J. Oper. Res.*, vol. 182, no. 3, pp. 1369–1382, 2007.

[30] R. Mukherjee, S. Debchoudhury, and S. Das, "Modified differential evolution with locality induced genetic operators for dynamic optimization," *Eur. J. Oper. Res.*, vol. 253, no. 2, pp. 337–355, 2016.

[31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[32] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.

MIGUEL G. VILLARREAL-CERVANTES received the B.S. degree in electronics engineering from the Veracruz Technologic Institute in 2003, and the M.Sc. and Ph.D. degrees in electrical engineering from the Center for Research and Advanced Studies, Mexico, in 2005 and 2010, respectively. He is a Full Professor with the Postgraduate Department, Centro de Innovación y Desarrollo Tecnológico en Cómputo of the Instituto Politécnico Nacional. His research interests include mechatronic design based on optimization (mono- and multi-objective optimization), bio-inspired metaheuristics in the optimal mechatronic design, optimal tuning for the mechatronic system control based on metaheuristic algorithms (offline and online strategies), and robotics.

ALEJANDRO RODRÍGUEZ-MOLINA received the B.S. degree in computer systems engineering from the Instituto Politécnico Nacional (IPN) in 2013, and the M.Sc. degree in computer science from the Center for Research and Advanced Studies of the IPN in 2015. He is currently pursuing the Ph.D. degree in robotic and mechatronic systems engineering with the Centro de Innovación y Desarrollo Tecnológico en Cómputo of the IPN. He is interested in the design and implementation of bio-inspired metaheuristics for optimization and their application to engineering problems.

CONSUELO-VARINIA GARCÍA-MENDOZA received the degree in engineering from the Instituto Politécnico Nacional (IPN) High Computer School, and the master's and Ph.D. degrees in advanced technology from the CICATA of Legaria, IPN, in 2006 and 2009, respectively. She is a Professor with the Department of Science and Computer Engineering, IPN High Computer School. Her research interests include metaheuristics approaches to optimization.

OLLIN PEÑALOZA-MEJÍA received the B.S. degree in mechatronics engineering from the Instituto Politécnico Nacional (IPN) in 2002, and the M.Sc. and Ph.D. degrees in electrical engineering specialized in mechatronics from the Center for Research and Advanced Studies, Mexico City, in 2003 and 2009, respectively. He has made several research stays with the control groups of the CICESE in Mexico and IRCCyN in France. He is a Full Professor with the Electrical and Electronics Engineering Department, Instituto Tecnológico de Sonora. His current research interests include the analysis and control of mechanical systems with applications to robotics, and the design of mechatronics systems.

GABRIEL SEPÚLVEDA-CERVANTES received the B.S. degree in electronics and control engineering from the Instituto Politécnico Nacional (IPN) of Mexico, Mexico, in 2005, and the Ph.D. degree in electrical engineering from the Center for Research and Advanced Studies of the IPN, Mexico, in 2009. He is currently a Professor of the mechatronic section in Centro de Innovación y Desarrollo Tecnológico en Cómputo of the IPN. His current research interests include haptic interfaces, virtual reality, and mechatronics.