

Offline robust tuning of the motion control for omnidirectional mobile robots

Omar Serrano-Pérez^a, Miguel G. Villarreal-Cervantes^{a,*}, Alejandro Rodríguez-Molina^b, Javier Serrano-Pérez^c

^a*Mecatronic Section, Postgraduate Department, Instituto Politécnico Nacional - CIDETEC, Mexico City 07700, México.*

^b*Tecnológico Nacional de México/IT de Tlalnepantla, Research and Postgraduate Division, Estado de México 54070, México.*

^c*Universidad Tecnológica de Tecámac, Carretera Federal México-Pachuca, km 37.5, Estado de México 55740, México.*

Abstract

In recent years, mobile robots have been helpful systems to perform a wide variety of complex tasks in daily life applications from industry, academy, and home. These robots carry out mobility on flat terrains, mainly in narrow spaces that are difficult to access or dangerous for humans. Therefore, increasing the efficiency of their movements through control technologies has become a topic of great interest for researchers. Among controllers, the linear ones are widely used to improve the efficiency of mobile robots because of their simplicity, reliability, and practicality, notwithstanding advanced control strategies. A well-tuned linear controller can show outstanding performances in controlled environments where the modeled and simulated conditions used for its adjustment are not too far from reality. However, actual operating environments are subject to uncertainties and disturbances that can hardly be accounted for during the controller tuning process. The above compromises the performance of the mobile robot in practice, and finding the appropriate controller parameters that enhance robustness becomes a crucial task. Therefore, this work presents a robust tuning approach for the controller of an omnidirectional mobile robot based on the solution of a nonlinear dynamic optimization problem through meta-heuristics.

*Corresponding author

Email address: `mvillarrealc@ipn.mx` (Miguel G. Villarreal-Cervantes)

Robustness is incorporated in the optimization problem by minimizing the sensitivity to the control performance indexes. Simultaneously, this is included through dynamic and stochastic variations in the meta-heuristic optimizer hyperparameters. A comparative statistical analysis is performed using robust and non-robust tuning approaches. Based on simulated and experimental tests, the proposed robust approach shows notable performance improvements regarding the non-robust one while minimizing operation errors in the presence of different uncertainty magnitudes.

Keywords: controller tuning; intelligent control; meta-heuristic algorithm; omnidirectional mobile robot; differential evolution.

1. Introduction

Performing automatic tasks efficiently with the least human intervention has been one of main goals of humankind since the first industrial revolution [1]. From that point, the evolution of the electrical, mechanical, control, and computational sciences allowed the development of robots [2].

Robots are programmable electromechanical systems able to perform multiple complex automated tasks in various environments [3]. In the past, robotic systems were mainly fix-based manipulators designed to operate within a predefined workspace. They were dedicated to performing intensive and repetitive tasks in assembly and manufacturing lines [4]. On the other side, modern robots are currently adopted in a wider variety of contexts, from the automation of domestic tasks in small homes [5], rescue, and surveillance activities in the wild [6].

The leap towards modern robotic systems was influenced by mobility. Nowadays, mobility is one of the most desirable features of robots [7]. This feature is related to the capacity of robots to perform extended sorts of movements, i.e., to move freely around the accessible physical space [8]. The above allows robots to operate in larger workspaces and unknown environments.

The robotic systems that incorporate the mobility feature are referred to

20 as mobile robots. This kind of system is present in almost any application that requires automated tasks in different locations, far enough apart to be unreachable by a fix-based manipulator [9].

The mobile robots can operate on land, underwater, or in the air, and develop different activities [10]. Terrestrial robots are especially useful to perform 25 complex tasks in narrow spaces that are difficult to access or dangerous to humans [11]. Some recent examples of terrestrial robots applications include object manipulation [12], lifting and transportation [13], construction [14], exploration of unknown zones [15], map building [16], and surveillance and inspection [17].

The use of wheels or legs achieves mobility in terrestrial robots. Legged 30 robots provide an enhanced movement on irregular terrains [18], while wheeled robots are simpler, faster, more energy-efficient, and very effective on flat terrains [19].

Among the wheeled robots, the omnidirectional ones are widely used in many applications because of their ability to perform the simultaneous and independent 35 translation and rotation movements, allowing them to obtain relatively high and accurate responses when executing required tasks in flat terrains and narrow scenarios [20, 21].

Omnidirectional Mobile Robots (OMRs) require suitable control systems to exploit all the above advantages. Nevertheless, due to their highly nonlinear 40 behavior and the number of actuators required to govern each wheel independently, the control of OMRs is a hard task. Moreover, the implementation costs of these systems must be affordable enough to support their use in domestic, academic, or industrial activities [22].

Today, there are many advanced control alternatives for OMRs and other 45 mobile robots. In [23], a model predictive control that incorporates the system dynamics is adopted to take an OMR over a feasible path generated by the potential field method. An effective cascaded Linear Quadratic Regulator on two loops for the speed regulation of the motors and the position/orientation control of an OMR is proposed in [24]. The work in [20] introduces an automatically 50 adjusted fuzzy controller for an OMR that can successfully handle dynamic

changes and navigation challenges. The proposals in [25, 26] utilize different sliding mode controllers to deal with uncertainties and disturbances in OMRs.

The above control proposals are effective but have very complex structures that are hard to adjust. Then, a high theoretical and technical knowledge is required to implement these controllers in practice. Also, the computational
55 cost of some alternatives makes them less affordable for a real application.

On the other hand, the Proportional Integral Derivative (PID) linear controller has been one of the most extended control choices for many decades. Also, it is known that the vast majority of industrial applications adopt a PID-like
60 controller [27]. The above is attributed to their simplicity (regarding their linear structure and low-cost implementation), universality (concerning their applicability in any context), and effectiveness in governing many complex dynamic systems such as the OMRs.

Like any other controller, the PID-like one has a set of parameters that must
65 be correctly adjusted to govern the OMR dynamics accurately. In different circumstances, a bad adjustment can cause performance losses and can even damage the controlled system. The above is one of the problems in control engineering known as controller tuning [28].

The PID-like controller tuning is a challenging task because there is an
70 immeasurable number of parameter combinations. Each choice has a different effect on the system (some choices could negatively affect the system), even more when the system is subject to many different operating conditions beyond those already imposed by its dynamic constraints [29].

The tuning of PID-like controllers and, in general, of any other control al-
75 ternative, can be addressed by four different approaches according to the taxonomy proposed in [30], which includes (a) Analytical methods, (b) Heuristic methods, (c) Optimization methods, and (d) Adaptive methods.

The approaches in (a) aim to ensure the stability of the closed-loop through the use of analysis tools from control theory [31]. On the other hand, the meth-
80 ods in (b) use the expert knowledge of the control engineer over the measurements of the controlled variables to set the appropriate controller parameters

[32]

About approaches in (a) and (b), there is no certainty that they can get the best controller performance because they are not optimal methods. Significant lower performances of alternatives (a) and (b) have been reported when compared with optimal ones [33, 34]. Moreover, the effectiveness of the controllers tuned by non-optimal methods can decrease even more when the plant is too complex to be fully modeled or when the operating conditions induce uncertainties or disturbances [35].

To handle the difficulties observed in the above non-optimal methods, the techniques in (c) state the tuning problem as a mathematical programming one, whose aim is to find the controller parameters that produce the best performance when controlling a dynamic system [36, 37, 38, 39].

Since the aforementioned complexity of the controller tuning carries over the mathematical programming problem, the effectiveness of classical optimization techniques is limited, and the use of more advanced optimizers can be required [40, 35]. In this sense, intelligent approaches such as the meta-heuristic optimization one have gained popularity in the last years [35, 41, 42]. In that approach, the controller tuning problem, as a mathematical programming one, can be successfully solved by meta-heuristic optimizers from evolutionary computation and swarm intelligence [43, 44].

Concerning the OMR, the approach (c) requires an accurate robot model to test and evaluate enough controller parameter combinations through optimization with meta-heuristics. The best parameters are implanted in the final controller and remain fixed. The research in [45] uses the Genetic Algorithm (GA) and the Differential Evolution (DE) to tune the PID controller parameters and optimize the step response of an OMR. In [46], the Particle Swarm Optimization (PSO) optimizes the PID controller parameters to accelerate the reaching of the path goals with a mobile robot. A hybrid GA-PSO algorithm is proposed in [47] to adjust the gains of the PI controller and reduce the trajectory tracking error. The work in [48] presents a comparative study of several meta-heuristics and a gradient-based optimizer for the simultaneous minimiza-

tion of the tracking error and the energy consumption through the PD controller tuning for an OMR.

115 At this point, it is worth mentioning that in real operating environments, all dynamic systems are continuously subjected to uncertainties and disturbances, which makes the models used for tuning inaccurate, and turns the tuning problem harder [49]. These two effects are typically complex, unpredictable, undesirable, and negatively affect the system operation. So, mathematical models can
120 not fully describe the physical phenomena around the robot [50, 51, 52]. Consequently, the calculated controller parameters are often unable to reach the desired performance, even when obtained through meta-heuristic optimization [53, 54, 55].

A way to deal with the problems described before is found in (d), where
125 the controller tuning is performed online based on the methods from previous classes. Particularly, adaptive methods based in (c) through the use of meta-heuristics have shown significant performance advantages concerning the rest of tuning alternatives when controlled systems are subject to uncertainties and disturbances, which happens in practice most of the time [56, 57].

130 In this way, the controller parameters are adjusted continuously during the task execution in the robot. The above effectively handle uncertainties and disturbances. Nevertheless, this approach requires a fast recalculation of the best controller parameters through successive meta-heuristic optimization processes. Hence, the computational cost of that approach is high and currently requires
135 a dedicated computer with a decent calculation capacity. Due to the previous comment, the online approach based on meta-heuristic optimization has only been applied in simpler dynamic systems than the OMR. In [57], different meta-heuristic optimizers were tested in the online controller tuning for the accurate speed regulation of the DC motor. The work in [58] proposed an online tuning
140 method to reduce the response error and enhance the smoothness of the controller output for the four-bar mechanism.

Since the on-board batteries of OMRs have energetic limitations, which caps the processing capacity, the tuning approach (d) can be unprofitable for a real

application. Alternatively, the controller performance is compromised in (c)
145 methods since they require an accurately simulated environment and also a de-
tailed OMR model. Such a model can be difficult to find in practice due to
the presence of uncertainties and disturbances. In this way, the controller pa-
rameters optimized from dynamic simulations may not achieve the best control
performance when implanted in a real prototype.

150 Despite the above difficulties, incorporating robustness features can deal
with the model limitations of the (c) tuning methods. The robust tuning aims
to identify an optimal set of control parameters or controller gains that makes
the system or process insensitive to changes in the operating conditions such as
uncertainties and disturbances.

155 Although there are researches that address the robust tuning of the con-
troller parameters, their proposals have been applied to high-order linear sys-
tems [59, 60]. Nevertheless, those proposals may not be effective enough when
controlling highly-nonlinear systems such as the OMRs. Moreover, incorporat-
ing the uncertainties and disturbances effects for robust tuning of the control
160 system has not been explored under the offline meta-heuristic optimization ap-
proach.

For the above reasons, this research aims to incorporate robustness features
in both the optimization problem for tuning and the meta-heuristic optimizer
to enhance the ability of the controller to handle unmodeled uncertainties and
165 disturbances with a real OMR. Robustness is incorporated in the optimization
problem by minimizing the sensitivity to the control performance indexes. Si-
multaneously, this is included through dynamic and stochastic variations in the
meta-heuristic optimizer hyperparameters.

So, the main contributions of the present work are the following:

- 170 1. The proposed robust tuning approach for the OMR controller formulated
as a nonlinear dynamic optimization problem where the solution is through
a robust meta-heuristic optimizer. The main features of the proposal are:
i) the obtained controller gains produce control performance indexes as

insensible as possible to uncertainty variations without knowing the uncertainty variation bounds, and *ii*) it is not required the careful selection of the controller gain bounds to achieve real experimental results (laboratory tests). Those features can be useful for practical purposes.

2. The experimental verification of the robust tuning approach with a physical prototype where the comparative statistical evidence with a non-robust tuning approach shows the effectiveness of the proposal.

The rest of the paper is organized as follows. Section 2 presents the details of the proposed robust tuning approach. The meta-heuristic operation is explained in Section 3. The experimental testbed and the results are discussed in Section 4. Finally, the conclusions are drawn in Section 5.

2. General overview of the robust tuning approach of controllers through dynamic optimization

The proposed Robust Tuning Approach for Controller Gains (RTACG) of mechatronic systems is formulated as an offline nonlinear dynamic optimization problem, where the control design objectives, and also their variations under the presence of uncertainties, are minimized to guarantee a closed-loop system performance as insensible as possible to such changes. In order to apply the RTACG, it is important to consider the following assumptions: 1) There is a valid dynamic model of the plant to be controlled where the numerical simulation largely replicates reality. 2) The dynamic model is represented as (nonlinear) differential equations. 3) The plant must be fully actuated. 4) There is a control system parameterized in the corresponding gains that stabilize the plant. 5) The task (desired trajectory) must be smooth and time-parameterized. 5) The uncertainty parameter must be implicitly or explicitly included in the objective function to make it robust. 6) The uncertainty variations and their bounds are not known.

The RTACG starts from the formulation of a Non-Robust Tuning Approach, which is stated as follows:

$$\min_{p^* \in R^{n_p}} \hat{J}(p) \quad (1)$$

$$\hat{J} = \int_0^{t_f} \sum_{i=1}^{n_j} \mu_i \hat{\mathcal{L}}_i(x, p, \xi, u, t) dt \quad (2)$$

subject to:

$$\frac{dx}{dt} = \mathcal{F}(x, p, \xi, u, t) \quad (3)$$

$$x(0) = x_0$$

$$g_j(p, t) \leq 0 \quad j = 1, \dots, n_g \quad (4)$$

$$h_k(p, t) = 0 \quad k = 1, \dots, n_h \quad (5)$$

$$p_{min} \leq p \leq p_{max} \quad (6)$$

The non-robust objective function \hat{J} (2) involves n_j control performance indexes. The i -th control performance index is represented by the positive semidefinite function $\hat{\mathcal{L}}_i$, which must be of at least class C^1 . Each performance index is weighted by the term μ_i such that the multi-objective optimization problem is transformed into a single-objective optimization problem by using the weighted sum approach to fulfill one trade-off of the Pareto front [61]. The plant, represented in the state space vector $x \in R^{n_x}$, is modeled through nonlinear differential equations (3) with the initial condition x_0 , where the control system is defined in $u = \mathcal{F}_u(x, p, \xi, t) \in R^{n_u}$. The static and dynamic constraints, inherent in the controller tuning, are included in (4) and (5) for the inequality and equality ones, respectively. The controller gains are grouped in the vector $p \in R^{n_p}$ with the lower and upper limits p_{min} and p_{max} , respectively. The uncertain parameter vector $\xi = \bar{\xi} + \zeta \in R^{n_\xi}$ consists of terms whose values are not previously known by the designer, and varies ζ units from its nominal known value $\bar{\xi}$. The Non-Robust Tuning Approach handles the parameters associated with the optimization problem in their nominal values without considering the variations ζ , i.e., $\xi = \bar{\xi}$. Nevertheless, the obtained solution from the Non-Robust Tuning Approach might be sensitive to changes in the nominal values of the design variables, the design parameters, and the state vector.

Then, the next step in the RTACG is to know the rate of change of the non-robust objective function $\Delta\hat{J}$ concerning the variations in the uncertainty vector $\Delta\xi$, which can be obtained by using the first-order Taylor expansion of the non-robust objective function \hat{J} around the nominal value of the uncertainty vector $\bar{\xi}$, i.e.,

$$\Delta\hat{J} = \Gamma \Big|_{\xi=\bar{\xi}} \Delta\xi \quad (7)$$

where $\Gamma \Big|_{\xi=\bar{\xi}} = \left[\frac{\partial\hat{J}}{\partial\xi_1}, \dots, \frac{\partial\hat{J}}{\partial\xi_{n_\xi}} \right] \Big|_{\xi=\bar{\xi}} \in R^{1 \times n_\xi}$, and the j -th gradient $\frac{\partial\hat{J}}{\partial\xi_j} \Big|_{\xi_j=\bar{\xi}_j} \in R$ is expressed as

$$\begin{aligned} \frac{\partial\hat{J}}{\partial\xi_j} \Big|_{\xi=\bar{\xi}} &= \int_0^{t_f} \sum_{i=1}^{n_j} \mu_i \left(\left(\frac{\partial\hat{\mathcal{L}}_i}{\partial x} + \frac{\partial\hat{\mathcal{L}}_i}{\partial u} \frac{\partial u}{\partial x} \right) \frac{\partial x}{\partial\xi_j} + \frac{\partial\hat{\mathcal{L}}_i}{\partial u} \frac{\partial u}{\partial\xi_j} + \frac{\partial\hat{\mathcal{L}}_i}{\partial\xi_j} \right) \Big|_{\xi=\bar{\xi}} dt \\ &= \int_0^{t_f} \sum_{i=1}^{n_j} \mu_i \tilde{J}_{i,j} \Big|_{\xi=\bar{\xi}} dt \end{aligned} \quad (8)$$

The j -th gradient $\frac{\partial\hat{J}}{\partial\xi_j} \Big|_{\xi_j=\bar{\xi}_j}$ requires the sensitivity equation $\frac{\partial x}{\partial\xi_j} \Big|_{\xi=\bar{\xi}} \in R^{n_x}$ of the state vector x . This can obtain by deriving both sides of the dynamics equation in (3) with respect to the j -th uncertainty ξ_j .

$$\begin{aligned} \frac{\partial\dot{x}}{\partial\xi_j} \Big|_{\xi=\bar{\xi}} &= \frac{\partial\mathcal{F}(x, p, \xi, u, t)}{\partial\xi_j} \Big|_{\xi=\bar{\xi}} \\ \frac{d}{dt} \frac{\partial x}{\partial\xi_j} \Big|_{\xi=\bar{\xi}} &= \left(\left(\frac{\partial\mathcal{F}}{\partial x} + \frac{\partial\mathcal{F}}{\partial u} \frac{\partial u}{\partial x} \right) \frac{\partial x}{\partial\xi_j} + \frac{\partial\mathcal{F}}{\partial u} \frac{\partial u}{\partial\xi_j} + \frac{\partial\mathcal{F}}{\partial\xi_j} \right) \Big|_{\xi=\bar{\xi}} \end{aligned} \quad (9)$$

Then, if the terms of the gradient vector $\Gamma \Big|_{\xi=\bar{\xi}} \approx 0$ in (7) are minimized, the variations in the non-robust objective function \hat{J} , due to the changes in the uncertainty parameter vector ξ , are decreased.

Finally, the RTACG, formulated as an offline nonlinear dynamic optimization problem, consists of finding the robust control gains included in p^{*r} that simultaneously minimize the aggregate function that involves n_j control performance indexes $\hat{\mathcal{L}}_i(x, p, \xi, u, t)$, and also the square of the $n_j \times n_\xi$ gradients of such indexes concerning the uncertainty vector ξ . The optimization problem

240 is formally stated in (10)-(17), where the i -th control performance index is weighted by the term μ_i , while its gradient with respect to the j -th uncertainty parameter is weighted by the term $\tilde{\mu}_{i,j}$ considering $\bar{\mu}_{i,j} = \tilde{\mu}_{i,j}\mu_i^2$.

$$\min_{p^*r \in R^{n_p}} \bar{J}(p) \quad (10)$$

$$\bar{J} = \hat{J} + \sum_{j=1}^{n_\xi} \tilde{\mu}_{i,j} \left. \frac{\partial \hat{J}}{\partial \xi_j} \right|_{\xi=\bar{\xi}}^2 \quad (11)$$

$$= \int_0^{t_f} \sum_{i=1}^{n_j} \left(\mu_i \hat{\mathcal{L}}_i(x, p, \xi, u, t) + \sum_{j=1}^{n_\xi} \bar{\mu}_{i,j} \tilde{J}_{i,j}^2 \Big|_{\xi=\bar{\xi}} \right) dt \quad (12)$$

subject to:

$$\frac{dx}{dt} = \mathcal{F}(x, p, \xi, u, t) \quad (13)$$

$$x(0) = x_0$$

$$\left. \frac{d}{dt} \frac{\partial x}{\partial \xi_j} \right|_{\xi=\bar{\xi}} = \left(\left(\frac{\partial \mathcal{F}}{\partial x} + \frac{\partial \mathcal{F}}{\partial u} \frac{\partial u}{\partial x} \right) \frac{\partial x}{\partial \xi_j} + \frac{\partial \mathcal{F}}{\partial u} \frac{\partial u}{\partial \xi_j} + \frac{\partial \mathcal{F}}{\partial \xi_j} \right) \Big|_{\xi=\bar{\xi}} \quad (14)$$

$$\frac{\partial x}{\partial \xi_j} = \mathbf{0} \quad \forall j = 1, \dots, n_\xi$$

$$g_j(p, t) \leq 0 \quad j = 1, \dots, n_g \quad (15)$$

$$h_k(p, t) = 0 \quad k = 1, \dots, n_h \quad (16)$$

$$p_{min} \leq p \leq p_{max} \quad (17)$$

It is important to note that RTACG minimizes the control performance indexes, as well as their changes due to the uncertainty vector, and considers the sensitivity state vector into the formulation. The main advantages of the proposal are: *i*) the obtained controller gains produce closed-loop system response characteristics (related to the control performance indexes) as insensible as possible to changes in the uncertainties, improving the execution of the task. *ii*) Unlike the case of the Non-Robust Approach presented in [48], the proposed RTACG does not require a careful selection of the controller gain bounds to provide satisfactory results in a real experiment (laboratory tests). This is because

the sensitivity state vector produces the information of the environment with the uncertainty vector, such that the obtained gains are more in line with the real scenario. *iii*) The uncertainty variation bounds are indirectly adjusted by the gradient information of the control performance indexes. So, by minimizing those indexes reduces the uncertainty variations such that the selection of their bounds is not required in the proposed RTACG.

2.1. Application of the proposed RTACG in Omnidirectional Mobile Robots

One of the main issues related to the N-RTACG in a real environment, is that the variations from their nominal values of the design variables, design parameters, or the state vector can significantly influence the feasibility and the closed-loop performance of the obtained tuning in the offline dynamic optimization problem. Such variations induce a high probability of violating constraints and yielding sudden changes in the performance functions related to the control performance indexes [62]. The proposed Robust Tuning Approach for Controller Gains (RTACG) incorporates protection against possible sources of uncertainties introduced in the system or the design environment. In many cases, the uncertainty variation bounds are often unknown, so the proposal reduces their effects without considering such bounds by incorporating the gradient of the control performance indexes.

For the particular case of the controller tuning for the OMR presented in [48], when derivative control gains in the N-RTACG are included as the control design variables, those tend to their upper value at the end of the optimization process. Real experimental tests are required to set the corresponding limits in such gains [30]. When those gains increase, the OMR velocity estimation also increments the noise and provokes vibrations in the OMR [48]. This leads the search space in the N-RTACG to be necessarily reduced (shrinkage of the control gain bounds) to avoid the increment of noise while estimating the velocity. It does not considerably affect the trajectory tracking. For this reason, the OMR Cartesian velocity $\bar{\xi} = [x_4, x_5, x_6]^T \in R^3$ is proposed in the present work as the nominal value of the uncertain parameter vector $\xi \in R^3$ to find control gains

that diminish the changes of the trajectory tracking under the noise effect in such velocity.

The necessary elements to formulate the proposed RTACG are stated in this section for the particular study case related to the control of an OMR under 285 uncertainties in the linear and angular velocity of the OMR Cartesian space.

The OMR dynamics consider the following assumptions to keep the mathematical derivation as simple as possible and without real loss of generality [63]:
 1) Rigid cart with non-deformable wheels are considered. 2) The movement 290 of the OMR is in a horizontal plane. 3) The friction is not set down. The schematic diagram of the OMR is shown in Fig. 1, where the OMR dynamics is represented as,

$$\dot{x} = f(x, p, \xi) + g(x, p, \xi)u \in R^6 \quad (18)$$

and its terms are detailed in (19)-(20) with $\lambda = \frac{1}{2mr^2+3J}$, $\kappa^\pm = \sin x_3 \pm \sqrt{3} \cos x_3$, and $\eta^\pm = \cos x_3 \pm \sqrt{3} \sin x_3$.

$$f(x, p, \xi) = \begin{bmatrix} x_4 & x_5 & x_6 & -3J\lambda x_6 x_5 & 3J\lambda x_4 x_6 & 0 \end{bmatrix}^T \in R^6 \quad (19)$$

$$g(x, p, \xi) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\lambda r \kappa^- & 2r\lambda \sin x_3 & -\lambda r \kappa^+ \\ \lambda r \eta^+ & -2r\lambda \cos x_3 & \lambda r \eta^- \\ \frac{Lr}{3JL^2+I_z r^2} & \frac{Lr}{3JL^2+I_z r^2} & \frac{Lr}{3JL^2+I_z r^2} \end{bmatrix} \in R^{6 \times 3} \quad (20)$$

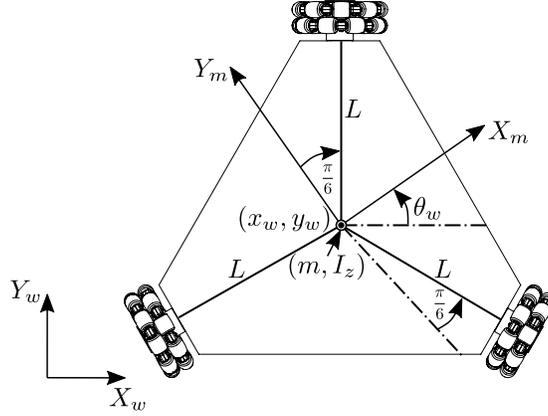


Figure 1: Schematic diagram of the OMR.

295 The state vector is denoted by $x = [x_1, x_2, x_3, x_4, x_5, x_6]^T = [x_w, y_w, \phi_w, \dot{x}_w, \dot{y}_w, \dot{\phi}_w]^T \in R^6$. The first three elements of x correspond to the linear/angular position of the OMR, and the latter to their velocities. The desired state vector is represented by $\bar{x} = [\bar{x}_d, \bar{y}_d, \bar{\phi}_d, \dot{\bar{x}}_d, \dot{\bar{y}}_d, \dot{\bar{\phi}}_d]^T \in R^6$.

The task-space PD controller given in (21), is selected to track the trajectory.

$$u = \check{J}^T (k_p e + k_d \dot{e}) \in R^3 \quad (21)$$

300 The state error vector and its velocity are defined as $e = [\bar{x}_d - x_1, \bar{y}_d - x_2, \bar{\phi}_d - x_3]^T \in R^3$, $\dot{e} = [\dot{\bar{x}}_d - x_4, \dot{\bar{y}}_d - x_5, \dot{\bar{\phi}}_d - x_6]^T \in R^3$, respectively. The control gains are grouped in the matrices $k_p = \text{diag}(k_{p1}, k_{p2}, k_{p3}) \in R^{3 \times 3}$ and $k_d = \text{diag}(k_{d1}, k_{d2}, k_{d3}) \in R^{3 \times 3}$. The matrix $\check{J} \in R^{3 \times 3}$, related to the Jacobian one of the omnidirectional mobile robot dynamics, is displayed in (22).

$$\check{J} = \begin{bmatrix} -\frac{\kappa^-}{3} & \frac{2}{3} \sin x_3 & -\frac{\kappa^+}{3} \\ \frac{\eta^+}{3} & -\frac{2}{3} \cos x_3 & \frac{\eta^-}{3} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \quad (22)$$

305 The N-RTACG is proposed in [48], where the dynamic optimization problem formulation is given in (23)-(28).

$$\min_{p^* \in R^6} \hat{J}(x, p, \xi) \quad (23)$$

where

$$\hat{J} = \mu_1 \hat{J}_1 + \mu_2 \hat{J}_2 \quad (24)$$

$$\hat{J}_1 = \int_0^{t_f} \hat{\mathcal{L}}_1(x, p, \xi) dt \quad (25)$$

$$\hat{J}_2 = \int_0^{t_f} \hat{\mathcal{L}}_2(x, p, \xi) dt \quad (26)$$

subject to:

$$\frac{dx}{dt} = f(x, p, \xi) + g(x, p, \xi)u \quad (27)$$

$$x(0) = x_0$$

$$p_{min} \leq p \leq p_{max} \quad (28)$$

In the N-RTACG, two non-robust performance indexes, related to the linear/angular position accumulated error \hat{J}_1 of the OMR (25), and the accumulated energy consumption \hat{J}_2 of the OMR motors (26), are taken into account in an aggregate function \hat{J} (24). The functions $\hat{\mathcal{L}}_1 = e_1^2(t) + e_2^2(t) + L^2 e_3^2(t)$ and $\hat{\mathcal{L}}_2 = u_1^2(t) + u_2^2(t) + u_3^2(t)$, related to the control performance indexes, consider the sum of the squared position errors and the sum of the squared motor wheel torques at each time instant t , respectively. The terms $\mu_1 = 0.95$ and $\mu_2 = 0.05$ in (24) involve an *a priori* preference selected for the N-RTACG, where $L = 0.2870m$. The design variable vector $p = [k_{p1}, k_{p2}, k_{p3}, k_{d1}, k_{d2}, k_{d3}]^T$ includes the gains of the task-space Proportional-Derivative (PD) controller. The dynamic constraint consists of the closed-loop control system dynamics (27) with the task-space PD controller (21). A parameterized hypocycloid trajectory (29)-(31) is the task to be executed by the robot in the $X_w - Y_w$ plane, considering a cosine movement in its orientation. A smooth path $\varphi(t) = (126E - 5)t^5 - (42E - 5)t^6 + (5.4E - 5)t^7 - (0.315E - 5)t^8 + (0.70E - 8)t^9$ is included in the trajectory for taking the robot from its start position to the initial position of the trajectory. For more details of the N-RTACG, please consult

[48].

$$\bar{x}_d(t) = \begin{cases} \varphi(t) & \forall t \leq 10 \\ 0.8181 \cos(0.1047t) + 0.1818 \cos(0.4712t) & \forall t > 10 \end{cases} \quad (29)$$

$$\bar{y}_d(t) = \begin{cases} 0 & \forall t \leq 10 \\ 0.8181 \sin(0.1047t) - 0.1818 \sin(0.4712t) & \forall t > 10 \end{cases} \quad (30)$$

$$\bar{\phi}_d(t) = \begin{cases} 0.4363\varphi(t) & \forall t \leq 10 \\ 0.4363 \cos(0.0523t) & \forall t > 10 \end{cases} \quad (31)$$

325 In the corresponding study case, the rate of change of the squared position error sum $\hat{\mathcal{L}}_1$, with respect to the j -th uncertainty ξ_j , is considered in the proposed Robust Tuning Approach for Controller Gains (RTACG). Then, the gradient $\left. \frac{\partial \hat{\mathcal{L}}_1}{\partial \xi_j} \right|_{\xi=\bar{\xi}}$ is computed as

$$\begin{aligned} \tilde{J}_{1,j} \Big|_{\xi=\bar{\xi}} &= \left. \frac{\partial \hat{\mathcal{L}}_1}{\partial \xi_j} \right|_{\xi=\bar{\xi}} \\ &= -2 \left(e_1 \frac{\partial x_1}{\partial \xi_j} + e_2 \frac{\partial x_2}{\partial \xi_j} + L^2 e_3 \frac{\partial x_3}{\partial \xi_j} \right) \Big|_{\xi=\bar{\xi}} \end{aligned} \quad (32)$$

330 where the state sensitivity vector $\left. \frac{\partial x}{\partial \xi_j} \right|_{\xi=\bar{\xi}}$ regarding the j -th uncertainty ξ_j is given by

$$\begin{aligned} \left. \frac{\partial \dot{x}}{\partial \xi_j} \right|_{\xi=\bar{\xi}} &= \left. \frac{\partial (f(x, p, \xi) + g(x, p, \xi)u)}{\partial \xi_j} \right|_{\xi=\bar{\xi}} \\ \left. \frac{d}{dt} \frac{\partial x}{\partial \xi_j} \right|_{\xi=\bar{\xi}} &= \left(A \frac{\partial x}{\partial \xi_j} + \frac{\partial g}{\partial x_3} u \frac{\partial x_3}{\partial \xi_j} + B \right) \Big|_{\xi=\bar{\xi}} \end{aligned} \quad (33)$$

where $\frac{\partial g}{\partial x_3} \in R^{6 \times 3}$, $A = \frac{\partial f}{\partial x} + g \frac{\partial u}{\partial x} \in R^{6 \times 6}$, $B = g \frac{\partial u}{\partial \xi_j} + \frac{\partial f}{\partial \xi_j} \in R^6$.

Formally, the offline dynamic optimization problem for the RTACG is formulated in (34)-(38). This finds the robust control gains p^{*r} that simultaneously minimize the position errors (25), their change rate (32), and the energy consumption (26), grouped in the weighted performance function (35), subject to
335 the differential equations related to the dynamics of the closed-loop system (36),

as well as the sensitivity vector (37), and also, the bounds in the control gain design variable vector (38).

$$\min_{p^*r \in R^6} \bar{J}(x, p, \xi) \quad (34)$$

$$\bar{J} = \underbrace{\sum_{i=1}^2 \mu_i \hat{J}_i}_{N-RTACG} + \underbrace{\int_0^{t_f} \sum_{j=1}^3 \bar{\mu}_{1,j} \tilde{J}_{1,j}^2 \Big|_{\xi=\bar{\xi}} dt}_{RTACG} \quad (35)$$

subject to:

$$\frac{dx}{dt} = f(x, p, \xi) + g(x, p, \xi)u \quad (36)$$

$$x(0) = x_0$$

$$\frac{d}{dt} \frac{\partial x}{\partial \xi_j} \Big|_{\xi=\bar{\xi}} = \left(A \frac{\partial x}{\partial \xi_j} + \frac{\partial g}{\partial x_3} u \frac{\partial x_3}{\partial \xi_j} + B \right) \Big|_{\xi=\bar{\xi}} \quad (37)$$

$$\frac{\partial x}{\partial \xi_j} = \mathbf{0} \quad \forall j = 1, \dots, n_\xi$$

$$p_{min} \leq p \leq p_{max} \quad (38)$$

340 Note: The MATLAB® code associated with the optimization of the RTACG approach is shown in <https://www.dropbox.com/sh/qy0pjb5qyu9kin/AAD2YRTkrZficjXIuWXd6yq0a?dl=0> for making more explicit the implementation details.

3. Differential Evolution (DE)

In the previous section, the proposed RTACG is formulated as a dynamic optimization problem. In this formulation, the OMR dynamic behavior (dynamic system) is represented through nonlinear differential equations. The transcription method is necessary to convert the original continuous-time formulation (infinite-dimensional optimization problem) into a discrete-time formulation (discrete-dimensional optimization problem) to apply nonlinear programming or metaheuristic algorithms in the solution of the problem [64]. In the case

350

presented in this paper, the control signal is parameterized using the task-space PD controller. The OMR dynamics and the system sensitivity equations are converted into a finite set of states through their solutions by the Euler method. It is important to mention that using the Euler method in the solution of the nonlinear differential equations required by the dynamic optimization problem (using the transcription method), the satisfaction (feasibility) of the dynamic constraints given in (18) and (37) are fulfilled. So, Differential Evolution (DE) can be used to solve the RTACG.

DE is a meta-heuristic technique inspired in the natural evolution according to the Neo-Darwinian theory [65]. DE considers three evolutionary operations: mutation, crossover, and selection, as is detailed in the next subsections.

As the RTACG is a nonlinear dynamic optimization problem, the self-adaptive Differential Evolution from [66] is used to solve it. The algorithm outperforms the original DE [65], and the Fuzzy Adaptive Differential Evolution algorithm (FADE) [67] in high dimensional benchmark problems. This is because the algorithm has a self-adaptation mechanism in the F and CR parameters to handle the dynamic variation of the fitness landscape over time. This adaptation improves the tracking of the promising solution movement in the search space and finds enhanced competitive solutions in the benchmark optimization problems [68, 69]. Hence, the self-adaptive differential evolution increases the reliability of the obtained solution.

The general operation of the self-adaptive Differential Evolution is observed in Algorithm 1 for its variant DE/Rand/1/Bin. First, a population with NP individuals (candidate solutions to the optimization problem) is randomly generated within the search space. Each individual corresponds to the control design variable vector p^r in the RTACG. For each generation, G , every individual \vec{x}_i^G in the population combines with a mutant which is generated by the difference of other different individuals at given rates F and CR to create an offspring denoted by \vec{u}_i^G .

Then, the individual \vec{x}_i^G competes with its offspring \vec{u}_i^G using a fitness indicator to determine which one must persist in the next generation as \vec{x}_i^{G+1} . The

value of the objective function with the computation of the OMR dynamics and state sensitivity dynamics is used to construct the fitness indicator. This process is repeated during G_{max} generations. Finally, the best individuals are found in the population when the maximum generation G_{max} is reached. Therefore, the robust control gains p^{*r} are found by selecting the best individual in the last generation.

Algorithm 1 Self-adaptive Differential Evolution Rand/1/Bin.

1: **Input:** $NP, G_{max}, p_{min}, p_{max}$.
2: **Output:** p^{*r}
3: **Begin**
4: $G \leftarrow 1$
5: **for** $i \leftarrow 1$ **to** NP **do**
6: Generate a random individual \vec{x}_i^G in search space $\forall i = 1, \dots, NP$.
7: Simulate dynamics (36) and sensitivity vector (37) for \vec{x}_i^G based on Algorithm 2.
8: Evaluate $\bar{J}(\vec{x}_i^G)$.
9: **end for**
10: Randomly initialize $F_i^0 \in (0, 1)$ and $CR_i^0 \in (0, 1) \forall i = 1, \dots, NP$.
11: **while** $G \leq G_{max}$ **do**
12: **for** $i \leftarrow 1$ **to** NP **do**
13: Calculate self-adaptative mechanism in the F_i^G (40) and CR_i^G (42) parameters.
14: Generate an offspring with operations: Mutation (39) and Crossover (41) for \vec{u}_i^G .
15: Replace \vec{u}_i^G outside the boundary with a random value in $[p_{min}, p_{max}]$ [70].
16: Simulate dynamics (36) and sensitivity vector (37) for \vec{u}_i^G based on Algorithm 2.
17: Evaluate $\bar{J}(\vec{u}_i^G)$.
18: **if** $\bar{J}(\vec{u}_i^G)$ is better than $\bar{J}(\vec{x}_i^G)$ **then**
19: $\vec{x}_i^{G+1} \leftarrow \vec{u}_i^G$
20: **else**
21: $\vec{x}_i^{G+1} \leftarrow \vec{x}_i^G$
22: **end if**
23: **end for**
24: $G \leftarrow G + 1$
25: **end while**
26: Initialize p^{*r} with the best solution in the last population.
27: **End**

Algorithm 2 Numerical simulation of the OMR dynamics and the sensitivity.

- 1: **Input:** The design parameter vector p presented in \vec{x}_i^G or \vec{u}_i^G .
 - 2: **Output:** The state vector $x(t)$, and the sensitivity vector $\frac{\partial x(t)}{\partial \xi_j} \forall j = 1, \dots, n_\xi$.
 - 3: **Begin**
 - 4: Set the initial condition $x(0) = x_0$ to the differential equation that describes the OMR dynamics.
 - 5: Set the initial condition $\frac{\partial x(0)}{\partial \xi_j} = \mathbf{0} \forall j = 1, \dots, n_\xi$ to the sensitivity differential equation.
 - 6: Divide the time horizon $t = [t_0 = 0, t_1 = \Delta t, t_2 = 2\Delta t, \dots, t_f = N\Delta t]$ in $N + 1$ intervals with a step size (integration time) of Δt .
 - 7: **for** $i \leftarrow 0$ **to** $N - 1$ **do**
 - 8: Calculate the trajectory (29)-(31) in the time t_i .
 - 9: Evaluate the task-space PD controller (21) in the state vector $x(t_i)$.
 - 10: Solve the nonlinear differential equation of the OMR dynamics (18) with the Euler's method for the time $t_i + \Delta t$.

$$x(t_i + \Delta t) = x(t_i) + \Delta t \frac{dx(t_i)}{dt}$$
 - 11: Solve the nonlinear differential equation of the sensitivity dynamics (33) with the Euler's method for the time $t_i + \Delta t$.

$$\frac{\partial x(t_i + \Delta t)}{\partial \xi_j} = \frac{\partial x(t_i)}{\partial \xi_j} + \Delta t \frac{d}{dt} \frac{\partial x(t_i)}{\partial \xi_j}$$
 - 12: **end for**
 - 13: **End**
-

Mutation

Individuals in the population are mutated to generate another population of NP mutant individuals \vec{v}_i^G . The mutation process makes changes to the original information of individuals (number of design variables D) using the difference of other individuals, as seen in the equation (39). The terms r_1 , r_2 and r_3 are the indexes of randomly selected individuals such that $r_1 \neq r_2 \neq r_3 \neq i$, $j = 1, \dots, D$.

$$v_{i,j}^G = \hat{x}_{r_1,j}^G + F_i^G (\hat{x}_{r_2,j}^G - \hat{x}_{r_3,j}^G) \quad (39)$$

395 The scale factor F_i^G is a positive parameter that controls the magnitude of the difference $(\hat{x}_{r_2,j}^G - \hat{x}_{r_3,j}^G)$. The self-adaptation mechanism of the F_i^G parameter [71] is adopted and shown in (40), where $F_l=0.1$, $F_u=0.9$ and $\psi=0.3$.

$$F_i^G = \begin{cases} F_l + rand(0,1)F_u & \text{if } rand(0,1) < \psi \\ F_i^{G-1} & \text{otherwise} \end{cases} \quad (40)$$

Crossover

Crossover is the process by which a population of NP descendant individuals is generated. This process is made by combining the genetic material of the original individual $(\hat{x}_{i,j}^G)$ with the material of the mutant one $(v_{i,j}^G)$. This work uses the uniform crossover, as seen in the equation (41).

$$\hat{u}_{i,j}^G = \begin{cases} v_{i,j}^G & \text{if } rand(0,1) < CR_i^G \text{ or } j = j_{rand} \\ \hat{x}_{i,j}^G & \text{otherwise} \end{cases} \quad (41)$$

The self-adaptation mechanism proposed in [71] is also included in the parameter CR_i^G , where $\omega=0.7$ is selected. This mechanism leads to better individuals with a higher chance of surviving and producing descendants, improving the exploration/exploitation of the search space, and this is shown below:

$$CR_i^G = \begin{cases} rand(0,1) & \text{if } rand(0,1) < \omega \\ CR_i^{G-1} & \text{otherwise} \end{cases} \quad (42)$$

Selection

Finally, the individuals that persist in the next-generation $G + 1$ are chosen between each of the generated solution \vec{u}_i^G and the original solution \vec{x}_i^G according to the fitness indicator.

4. Results

In this section, the proposed approach called RTACG is applied to the controller of an OMR. A comparative study of the proposal is carried out with a

control tuning approach reported in the recent specialized literature. The Non-
415 Robust Tuning Approach for Controller Gains presented in [48] is considered in
the comparison to highlight the effectiveness of the proposal under the effect of
Gaussian noise in the Cartesian velocity estimation of the OMR.

First, the empirical analysis on the efficiency of the evolutionary algorithm
(self-adaptive Differential Evolution) is given in Section 4.1 to search for the
420 best solution for both approaches (RTACG and N-RTACG). Then, once the
controller gains for both approaches are obtained, comparative results in sim-
ulation of the closed-loop system are described in Section 4.2. Finally, those
gains are set in a laboratory prototype (real-time experimentation) to perform
a comparative analysis in a real environment, and the discussion is given in
425 Section 4.3.

4.1. Discussion of the algorithm performance in the controller tuning approaches

4.1.1. Differential evolution performance with algorithm parameter variations

In the first part of the Section 4.1, the performance of the self-adaptive
differential evolution is compared with the results obtained by the original dif-
430 ferential evolution (original DE), which the latter considers fixed parameters in
CR and F. The main purpose is to show the effects of the self-adapting param-
eters CR and F in the differential evolution algorithm for the solution of the
proposed RTACG (nonlinear dynamic optimization problem), and confirm the
competitive and reliable performance of the algorithm.

435 The parameters associated to the optimization problems are set as follows:
the control design variable bounds are proposed as $p_{min} \in [0, 0, 0, 0, 0, 0]^T \in R^6$
and $p_{max} \in [3000, 3000, 3000, 100, 100, 100]^T \in R^6$. The weights in the
performance function are established as $\mu_1 = 0.95$, $\mu_2 = 0.05$, $\bar{\mu}_{1,1} = \bar{\mu}_{1,2} =$
 $\bar{\mu}_{1,3} = 1e16$. The first two weights μ_1 and μ_2 were selected according to [48], and
440 the last ones were empirically proposed based on a set of trials with different
weight values in the optimization problem. The above with the purpose of
obtaining the most suitable control performance. The differential equations
that describe the OMR dynamics and the state sensitivity vector are solved

Table 1: Kinematic and dynamic parameters of the OMR.

<i>Parameter</i>	<i>Description</i>	<i>Value</i>	<i>Units</i>
r	Wheel radius	0.0625	m
L	Wheel distance from the mass center	0.2870	m
m	Mass	16.3190	kg
J	Wheel inertia	$5.82E - 4$	$kg \cdot m^2$
I_z	Mobile robot inertia	0.5160	$kg \cdot m^2$

by the Euler’s integration method with the initial condition $x_0 = \mathbf{0} \in R^6$, the
445 integration time $\Delta t = 5ms$ and the final time $t_f = 130s$. The OMR dynamics
uses the kinematic and dynamic parameters of Table 1 [48]. On the other
hand, the same population size $NP = 50$ and the same maximum generation
number $Gmax = 700$ are selected for each algorithm. Those parameters were
empirically chosen by visualizing the generation where the performance function
450 value is steady and has a suitable performance.

Both Differential Evolution algorithms are implemented in MATLAB®2020
over a PC with a Xenon®E5-2603V2 1.8GHZ CPU and 32GB in RAM. The
computational complexity of the proposed RTACG is related to the compu-
tational complexity of the Differential Evolution algorithm, which solves the
455 problem. The computational complexity [72] depends on the performance func-
tion evaluation $\bar{J}(p)$, the maximum number of generations G_{max} , and the size
of the population NP . So, the computational complexity in the Big O notation
is $\mathcal{O}(\bar{J}(p) \cdot NP \cdot G_{max})$, and the average time to perform a single execution of
the algorithm for solving the RTACG is around twelve hours.

460 The comparative study involves one hundred executions with different fixed
values of CR and F in the original DE, and also other one hundred executions
with different fixed values of ψ and ω in the self-adaptive Differential Evolution.

The descriptive statistics of the data obtained from the best objective func-
tion values of the executions per each algorithm is showed in Table 2. The mean
465 ($mean(\bar{J})$), the standard deviation ($\sigma(\bar{J})$), the maximum and minimum values

($max(\bar{J})$ and $min(\bar{J})$) are provided in the columns of such a table. In addition, the controller gains of the best and the worst executions with respect to the performance function are displayed in Table 3.

Table 2: Descriptive statistics of the performance function \bar{J} through one hundred different executions in the original differential evolution and in the self-adaptative differential evolution.

	$mean(\bar{J})$	$\sigma(\bar{J})$	$max(\bar{J})$	$min(\bar{J})$
Original DE	$8.58049E - 3$	$1.62E - 6$	$8.589026168902798E - 3$	$8.5796547017689E - 3$
Self-adaptative DE	$8.57965E - 3$	$4.97E - 16$	$8.579654701773048E - 3$	$8.5796547017691E - 3$

Table 3: Obtained PD control gains by using the original differential evolution and self-adaptative differential evolution.

Original DE	CR	F	ω	ψ	k_{p1}	k_{p2}	k_{p3}	k_{p4}	k_{p5}	k_{p6}
<i>Best</i>	0.80	0.50	-	-	1483.5170	1483.4499	2636.8217	68.0400	67.9976	16.1084
<i>Worse</i>	1.00	0.10	-	-	1667.8055	1668.9065	1028.6443	46.8862	47.4172	10.8110
Self-adaptative DE	CR	F	ω	ψ	k_{p1}	k_{p2}	k_{p3}	k_{p4}	k_{p5}	k_{p6}
<i>Best</i>	[0.10, 0.90]	[0.10, 0.90]	0.70	0.30	1483.5170	1483.4498	2637.6843	68.0400	67.9976	16.1031
<i>Worse</i>	[0.10, 0.90]	[0.10, 0.90]	0.10	0.30	1483.5309	1483.4637	2646.0829	68.0402	67.9978	16.1483

The following highlights from the Tables 2 and 3 are: *i*) The results of the self-adaptative DE show a reliable performance through the executions with different algorithm parameters because the average of the objective function values is lower than the one in the original DE (see Table 2). Besides, the standard deviation indicates that in all executions of the self-adaptative DE, the obtained solutions have very similar performance functions. On the other hand, the design variable vector in the self-adaptative DE is less affected by the algorithm parameters than the original DE results, as observed in Table 3 with the best and the worst design variable vectors. Hence, the convergence of the self-adaptative DE is hardly affected by the changes in algorithm parameters. *ii*) The self-adaptative DE presents competitive results because, through executions with different algorithm parameters, the obtained solutions present

a similar competitive performance. While the original DE, a suitable solution is obtained through a rigorous tuning procedure of the algorithm parameters. Hence, the self-adaptative DE does not require a careful selection of the algorithm parameter to obtain a competitive solution. Also, the computational
485 time to search for a competitive solution in the RTACG is decreased with self-adaptative DE because it does not require an exhaustive tuning procedure of the algorithm parameters, i.e., whatever selection of the algorithm parameter provides a competitive solution.

4.1.2. *Performance of the self-adaptive DE in the RTACG and N-RTACG approaches.*
490

In the second part of the Section 4.1, the analysis of the self-adaptive DE/Rand/1/Bin behavior is presented when separately solving the RTACG and N-RTACG approaches. The main purpose is to obtain the best design variable vector among the thirty independent executions for each approach because of
495 the stochastic nature of the algorithm [73].

The parameters associated to the optimization problems for both approaches and those corresponding to the self-adaptive DE algorithm are set according to Section 4.1.1.

The descriptive statistics of the thirty executions (thirty samples) of the self-
500 adaptive DE/Rand/1/Bin for each optimization problem related to the RTACG and N-RTACG are presented in Table 4. Each sample for statistics is related to the best objective function (the best fitness of individuals in the last generation) at each execution. The first two columns are related to the mean values of the thirty samples, and the last ones include the standard deviation of the samples.
505 It is observed, in the standard deviation, that the performance function value through the executions converges to a similar value (mean value). This indicates that the best solutions through the executions converge to the same region of the search space, and so, sub-optimal solutions are not found. Then, the obtained solutions are referred to in this paper as a global one.

510 The performance function behavior of the best individuals through execu-

Table 4: Descriptive statistics of the obtained results through the use of self-adaptive DE/Rand/1/Bin.

$\text{mean}(\hat{J})$	/	$\text{mean}(\bar{J})$	$\sigma(\hat{J})$	/	$\sigma(\bar{J})$
$8.5761E - 3$	/	$8.5796E - 3$	$7.541E - 18$	/	$1.481E - 16$

\hat{J} : Performance function for the N-RTACG. \bar{J} : Performance function for the RTACG.

tions of the self-adaptive DE/Rand/1/Bin in both tuning approaches is shown in Fig. 2. The algorithm convergence requires around 200 generations for the RTACG and around 50 generations for the N-RTACG. The slow converge in the proposed RTACG is because of the optimization problem complexity by incorporating the sensitivities, i.e., more complexity in the controller tuning approach results in more generation in the algorithm to converge.

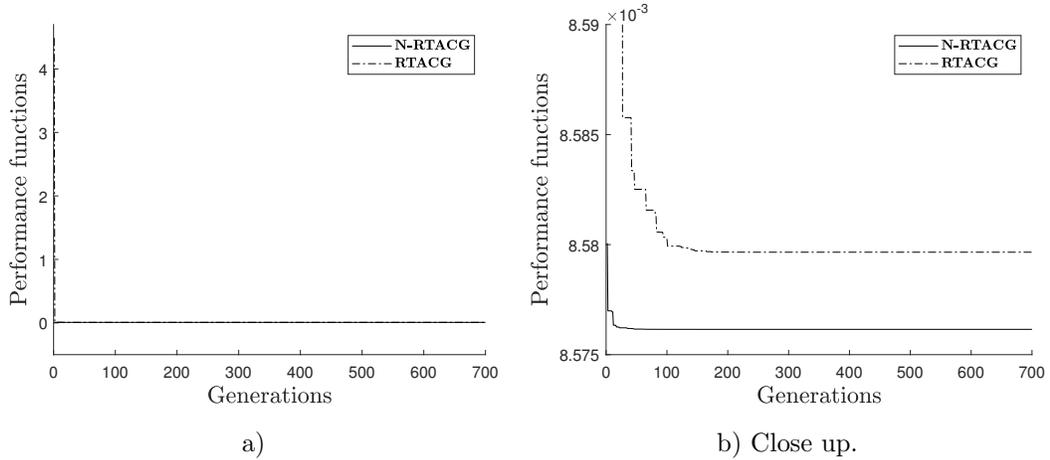


Figure 2: Behavior of the performance function through the generation in the RTACG and N-RTACG using the self-adaptive DE/Rand/1/Bin.

The robust control gains p^{*r} obtained by the RTACG, and the non-robust control gains p^* acquired by the N-RTACG are displayed in Table 5. These gains correspond to the best design variables vector of the thirty independent executions that solve the RTACG and N-RTACG problems, respectively. It is important to note that the gains k_{p_2} and k_{d_1} obtained by the N-RTACG attain their upper limits (i.e., there are active box constraints). While the robust

Table 5: Obtained gains by using RTACG and N-RTACG.

	\hat{k}_{p1}	\hat{k}_{p2}	\hat{k}_{p3}	k_{d1}	k_{d2}	k_{d3}
p^{*r}	1483.5170	1483.4498	2637.6842	68.0400	67.9976	16.1031
p^*	1148.8169	2999.9999	2576.3175	99.9999	16.2817	15.8837

control gains p^{*r} are into their bounds. This indicates that the sensitivity of the linear and angular position concerning the velocity allows finding feasible solutions into their limits, and an additional rigorous procedure to set the control gain bounds described in the N-RTACG [48], is not required.

4.2. Robustness analysis for the RTACG in simulation

In this subsection, the comparative study in numerical simulation of the closed-loop system of the controller with the OMR is presented with the use of the robust and non-robust gains obtained from Table 5. In the simulation, different disturbances in the velocities of the OMR are included in the differential equations of the OMR dynamics to analyze the changes in the control performance index (trajectory tracking) with respect to velocity variation, and so, to show the effectiveness of the RTACG in the trajectory tracking of the OMR under uncertainty variations.

Hence, the performance of the controller with the gains given in Table 5 for the proposed RTACG is compared with those acquired by the N-RTACG. The simulation results include random changes in the lineal and angular velocities of the OMR. This simulates Gaussian noise in the velocity states, and these states are considered as the uncertainty vector. Then, the uncertainty vector $\xi = \bar{\xi} + \zeta \in R^3$ considers variations $\zeta(t) = [\zeta_1(t), \zeta_2(t), \zeta_3(t)]^T \in R^3$ in the nominal velocity states $\bar{\xi} = [x_4, x_5, x_6]^T \in R^3$ of the OMR, as is observed in Fig. 3. The disturbance vector $\zeta(t)$ is proposed at each time instant t as a uniformly distributed random number vector in the interval $[\zeta_{min}, \zeta_{max}]$, which will represent Gaussian noises in the nominal velocity states. It is important to point out that the same disturbance vector $\zeta(t)$ is included in the closed-loop control simulations with the gains obtained from RTACG and N-RTACG to

present a fair comparative result.

The robustness analysis in simulation involves the quantification of the performance function, the sensitivity of the control performance index under variations in the disturbance ζ of the linear and angular velocities of the OMR, and the Root Mean Square (RMS) value of the velocities in the OMR. Three different uncertainties in the velocity signals are incorporated into the closed-loop system to analyze their effects. In the first test, the disturbance vector ζ is reduced to zero in the uncertainty vector ξ , i.e., there is no noise in the velocity signals of the OMR ($\xi = \bar{\xi}$). The i -th disturbance is randomly incremented in the interval $\zeta_i(t) \in [-5E-5, 5E-5]$ and $\zeta_i(t) \in [-5E-3, 5E-3]$ for the second and the third test, to simulate unknown uncertainties in the velocity signals of the closed-loop system.

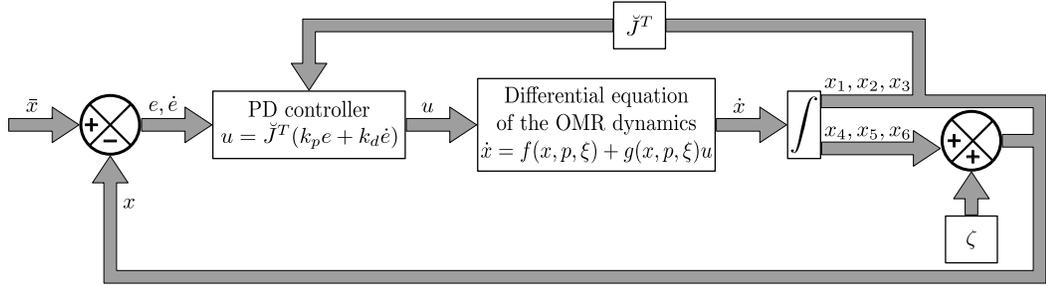


Figure 3: Schematic representation of the closed-loop system of the OMR with the task-space PD controller and the uncertainties $\zeta(t)$.

The performance function value \bar{J} and its terms are displayed in Table 6. The first column includes the minimum and maximum variations in the disturbance vector ζ . The second column shows the performance function \bar{J} evaluated both with the robust gains p^{*r} obtained by the RTACG as well as with the non-robust gains p^* given by N-RTACG. The terms of the performance function \bar{J} given by the linear/angular position error \hat{J}_1 , the energy consumption \hat{J}_2 , and the sensitivities $\tilde{J}_{1,j}$ of position error concerning the j -th velocities x_j of the OMR, are presented in the rest of the columns. The boldface numbers indicate

the lowest value (the best) between both approaches. It is observed that the use of the robust gains p^{*r} significantly reduces the overall performance function \bar{J} concerning the non-robust ones (see the second column of Table 6).

Besides, it is important to note that the proposed RTACG minimizes the sensitivities of the position errors, as is shown in the last three columns of Table 6. The decrement of such sensitivities impacts the maximum variation of the performance function \hat{J}_1 (position errors) under the effect of uncertainties in the velocity signals. With the increment of uncertainties, the robust design p^{*r} can maintain the position error \hat{J}_1 into the interval $[1.58E-7, 2.75E-6]$, while the non-robust design can maintain it into the interval $[1.85E-7, 3.30E-5]$ (see third column in Table 6). The reduction in the position error \hat{J}_1 , provided by the robust control gains p^{*r} with respect to those given by the non-robust one p^* , is around 17.08%, 16.98%, and 1100%, when the disturbances are $\zeta_i(t) = 0$, $\zeta_i(t) \in [-5E-5, 5E-5]$, and $\zeta_i(t) \in [-5E-3, 5E-3]$, respectively. This indicates that the proposed RTACG finds gains that make the linear and angular position errors as insensible as possible to variations in the velocity signals.

Table 6: Performance function value in the simulation results with the obtained gains by using the RTACG and the N-RTACG under uncertainty variations.

RTACG	ζ	$\bar{J}(p^{*r})$	$\hat{J}_1(p^{*r})$	$\hat{J}_2(p^{*r})$	$\tilde{J}_{1,1}(p^{*r})$	$\tilde{J}_{1,2}(p^{*r})$	$\tilde{J}_{1,3}(p^{*r})$
	0	8.5796E-3	1.58E-7	$1.7155E-1$	6.39E-23	4.76E-23	4.76E-23
	$[-5E-5, 5E-5]$	0.2623	1.59E-7	1.7360E-1	1.25E-17	1.26E-17	1.78E-19
	$[-5E-3, 5E-3]$	2538.43	2.75E-6	1.8258E1	1.25E-13	1.26E-13	1.78E-15
N-RTACG	ζ	$\bar{J}(p^*)$	$\hat{J}_1(p^*)$	$\hat{J}_2(p^*)$	$\tilde{J}_{1,1}(p^*)$	$\tilde{J}_{1,2}(p^*)$	$\tilde{J}_{1,3}(p^*)$
	0	4.4723	$1.85E-7$	1.7152E-1	$1.48E-16$	$1.48E-16$	$1.48E-16$
	$[-5E-5, 5E-5]$	79.1012	$1.89E-7$	$1.9219E-1$	$1.74E-16$	$7.57E-15$	$1.58E-16$
	$[-5E-3, 5E-3]$	$2.901E6$	$3.30E-5$	$2.0450E2$	$7.61E-11$	$1.37E-10$	$7.59E-11$

The sensitivities of the control performance index, related to the change rate of the sum of the squared position errors $\hat{\mathcal{L}}_1(t)$ concerning the velocities, are displayed in Fig. 4. They can be observed under the effects of different

disturbance vector ζ . This visually confirms that the RTACG obtains control gains that significantly reduce the sensitivities of the position errors to the velocities changes ($\partial \hat{\mathcal{L}}_1 / \partial x_i \forall i = 4, 5, 6$). So, the gains obtained by the proposed
590 RTACG significantly reduce the position error sensitivities in such a way that the closed-loop performance of the controller is better than the N-RTACG when uncertainties arise.

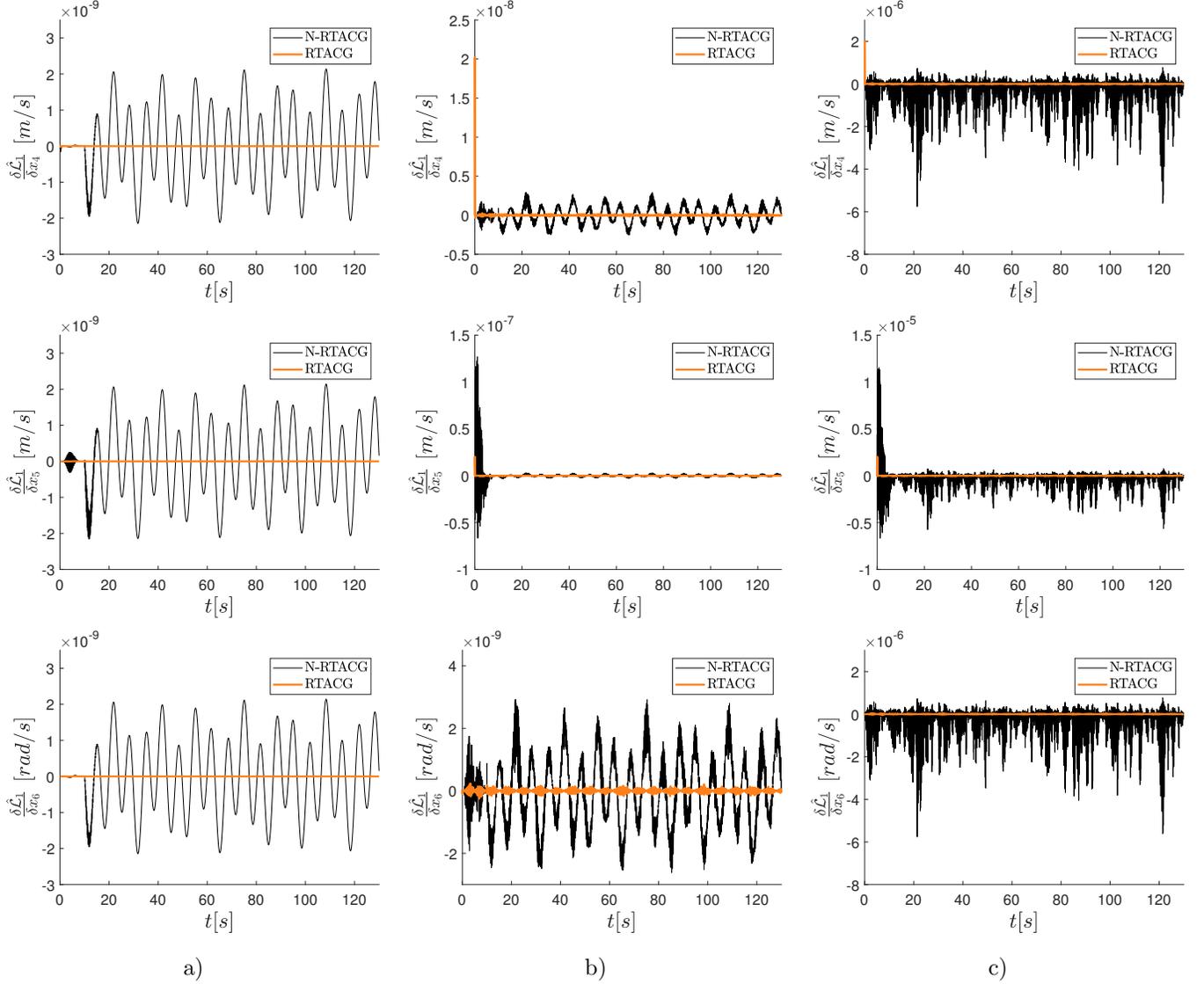


Figure 4: Sensitivity of the position error with respect to velocities incorporating different disturbances ζ by using RTACG and N-RTACG in the closed-loop system. a) $\zeta = 0$. b) $\zeta = [-5E - 5, 5E - 5]$. c) $\zeta = [-5E - 3, 5E - 3]$.

On the other hand, the RMS values of the errors in the OMR velocity \dot{e} , with the three different uncertainties in the velocity signals, are exhibited in Table

595 7. The first column relates the three different uncertainties introduced in the closed-loop system. The second to fourth column presents the RMS value of the Cartesian velocity errors of the OMR. The fifth column provides the sum of the RMS values given in the second to the fourth column. The last column $Var\%$ represents the percentage of change between the RMS velocity error when $\zeta = 0$ and the RMS velocity error with uncertainties. The boldface number indicates
600 the lowest value (the best) between both approaches. In the test where there are not disturbances $\zeta = 0 \in R^3$ ($\xi = \bar{\xi}$), the RMS values in the velocity errors of the closed-loop system are similar in both tuning approaches (see the fifth column of Table 7). Nevertheless, the robust gains can reduce the velocity
605 variation around 313% and 146% with the disturbances $\zeta_i(t) \in [-5E-5, 5E-5]$ and $\zeta_i(t) \in [-5E-3, 5E-3]$, respectively, when compared with the non-robust gains. Hence, the robust gains compensate better for the sudden changes in the velocity error given by disturbances.

Table 7: RMS values for the simulation results of the OMR velocity error with the controller gains obtained by the RTACG and the N-RTACG under different uncertainties.

	ζ	$RMS(\dot{e}_1)$ [m/s]	$RMS(\dot{e}_2)$ [m/s]	$RMS(\dot{e}_3)$ [rad/s]	$\sum_{i=1}^3 RMS(\dot{e}_i)$	$Var\%$
RTACG	0	$8.511E-5$	$7.083E-5$	$1.807E-5$	$1.740E-4$	0.0
	$[-5E-5, 5E-5]$	$9.484E-5$	$8.211E-5$	$5.848E-5$	$2.354E-4$	35.3
	$[-5E-3, 5E-3]$	$4.072E-3$	$4.072E-3$	$5.614E-3$	$1.376E-2$	7806.1
	ζ	$RMS(\dot{e}_1)$ [m/s]	$RMS(\dot{e}_2)$ [m/s]	$RMS(\dot{e}_3)$ [rad/s]	$\sum_{i=1}^3 RMS(\dot{e}_i)$	$Var\%$
N-RTACG	0	$8.402E-5$	$7.039E-5$	$1.807E-5$	$1.725E-4$	0.0
	$[-5E-5, 5E-5]$	$9.103E-5$	$2.754E-4$	$5.732E-5$	$4.238E-4$	145.7
	$[-5E-3, 5E-3]$	$3.441E-3$	$2.445E-2$	$5.489E-3$	$3.338E-2$	19250.5

Based on the previous discussion, it is confirmed that the RTACG can provide
610 controller gains that efficiently handle the uncertainties in the system. In this manner, the control performance index of the closed-loop system can tolerate more uncertainty variations without greatly degrading its quality.

4.3. Robustness analysis for the RTACG in a real environment

In this subsection, the experimental results of the gains obtained by the
615 proposed RTACG are validated through a comparative empirical analysis. This
analysis consists of implementing the gains obtained by the RTACG and the
N-RTACG (see Table 5) in a laboratory OMR prototype (real-time experimen-
tation) and statistically evaluating the controller performance under the effect of
different uncertainties included in the velocity signals. It is important to point
620 out that the different disturbance vector $\zeta(t)$ in a specific interval is included in
the closed-loop control real-time experimentation with the gains obtained from
RTACG and N-RTACG, i.e., at each experimentation different disturbance vec-
tor is considered at each time. For that reason, the descriptive and inferential
statistical analysis of thirty independent executions of the closed-loop system
625 for each approach is fulfilled to provide a formal conclusion related to the effec-
tiveness of the RTACG in the trajectory tracking of the OMR under uncertainty
variations in a real environment.

The schematic diagram of the closed-loop system for the OMR prototype is
shown in Fig. 5. The OMR includes a motherboard mini-ITX GA-D425TUD
630 with the intel® processor *Atom*TM D525, 4 GB of RAM and 250 GB hard disk,
where the control system and the position of the OMR, given by the odometry
system [74], are programmed in the Simulink® software. The Sensoray 626 PCI
board provides the data acquisition of the motor position and the out sampling
signals (related to the control signal). Three advanced motion controls, 12A8
635 series analog servo drives, amplify the control signals to drive the three brushed
DC motors of the OMR. External DC power sources supply the required voltage
to the electrical components in the OMR.

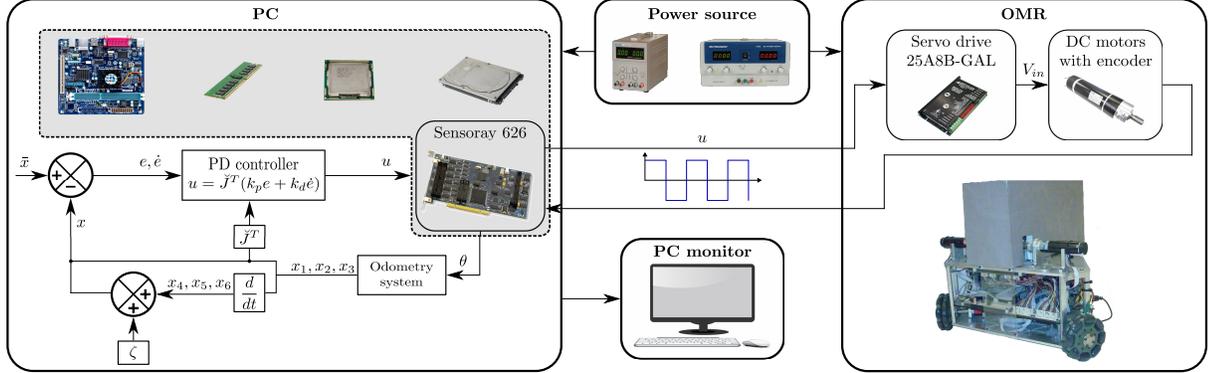


Figure 5: Schematic diagram of the closed-loop system for the experimental OMR prototype.

In the real-time experimentation, uniformly distributed random disturbances $\zeta(t) \in R^3$ are programmed and incorporated in the velocity signals. The uncertainties are included in the OMR velocity, as is observed in Fig. 5. The sampling time in the experiments is $5ms$. Three tests are considered. The first one does not include disturbances ($\zeta = 0$) and only considers those presented in the real environment, which are unknown. The last two tests contemplate two different intervals of disturbances $\zeta(t) \in [-11E - 5, 11E5]$ and $\zeta(t) \in [-1E - 3, 1E3]$, respectively. Each test consists of carrying out thirty independent executions of the closed-loop system with the gains obtained by the RTACG and the N-RTACG to obtain enough experimental evidence to perform a statistical study. Descriptive and inferential statistics [73] are used to make general conclusions about the performance of the RTACG concerning N-RTACG under the effect of different uncertainties in the real environment related to the three tests previously commented. The statistics sample is related to the position error \hat{J}_1 because the RTACG, for the particular study case, makes the position error as insensible as possible to the velocity uncertainties for the real environment.

Table 8 shows the descriptive statistics of the thirty executions. The second column indicates the mean value of the samples, the third one the standard deviation, and the last two columns represent the minimum and maximum val-

ues. The boldface number indicates the lowest value (the best) between both approaches. The results indicate that the controller gains obtained by the proposed RTACG can reduce the position error under the effect of uncertainties. Nevertheless, if the disturbance ζ is set to zero, the control performance in the position error is worse than the reported in the N-RTACG. Besides, the standard deviation presented in the third column of Table 8 indicates that, through the thirty executions of the controller in the real OMR prototype, the position errors are closer to the corresponding average than the ones in the N-RTACG. This indicates that the RTACG gives a more consistent control performance.

Table 8: Descriptive statistics of the thirty real-time executions of the controller with the gains p^{*r} and p^* obtained by RTACG and N-RTACG, respectively.

RTACG	ζ	$\text{mean}(\hat{J}_1(p^{*r}))$	$\sigma(\hat{J}_1(p^{*r}))$	$\text{Min}(\hat{J}_1(p^{*r}))$	$\text{Max}(\hat{J}_1(p^{*r}))$
	0	$7.709891E - 4$	$1.880423E - 6$	$7.675822E - 4$	$7.736226E - 4$
	$[-11E - 5, 11E - 5]$	$7.701144E - 4$	$8.311074E - 7$	$7.680850E - 4$	$7.715172E - 4$
	$[-1E - 3, 1E - 3]$	$5.045223E - 3$	$2.165748E - 5$	$5.006271E - 3$	$5.112541E - 3$
N-RTACG	ζ	$\text{mean}(\hat{J}_1(p^*))$	$\sigma(\hat{J}_1(p^*))$	$\text{Min}(\hat{J}_1(p^*))$	$\text{Max}(\hat{J}_1(p^*))$
	0	$7.638518E - 4$	$2.716012E - 6$	$7.602548E - 4$	$7.705609E - 4$
	$[-11E - 5, 11E - 5]$	$7.738168E - 4$	$1.877205E - 6$	$7.708005E - 4$	$7.785132E - 4$
	$[-1E - 3, 1E - 3]$	$7.037825E - 3$	$4.050501E - 5$	$6.986586E - 3$	$7.143789E - 3$

The results presented in Table 8 only describe the thirty samples. To make general conclusions about the performance of the RTACG, a pairwise comparison of the samples obtained by the RTACG and N-RTACG is performed using the Wilcoxon signed-rank nonparametric test. Those results are presented in Table 9, where the first column indicates the uncertainty variations, the last column represents the p-value, the third and the fourth columns are the rank-sums R+ and R-, respectively. For the nonparametric test, the two-sided alternative hypothesis is selected, which means that the median of the results given by the proposed RTACG is different from the one given by N-RTACG. The alternative hypothesis is confirmed when the p-value is less than the significance level α .

The winner (the best performance) in the comparison is chosen based on the rank-sums R+ (meaning that the first tuning approach outperformed the second one) and R- (meaning that the second tuning approach outperformed the first one). In this case, the significance level is chosen as $\alpha = 0.05$. On the contrary, when the null hypothesis is confirmed, i.e., when the p-value is larger than the significance level, then non-conclusive results are obtained. In that case, there is not enough information to guarantee that one approach outperforms the other. Based on the results presented in Table 9, it is confirmed that, with confidence around 95%, the controller with the gains obtained by the proposed RTACG can efficiently compensate different uncertainties in OMR velocities compared to the non-robust approach (N-RTACG). On the other hand, it is also confirmed that, with confidence around 95%, the controller with the gains acquired by the N-RTACG is the aptest one when there are no disturbances ($\zeta = 0$).

Table 9: Wilcoxon signed-rank sum test of the thirty real-time executions of the controller with the RTACG and N-RTACG.

ζ	Comparison	R_+	R_-	$p - value$
0	RTACG vs N - RTACG	0	465	$1.862E - 9$
$[-11E - 5, 11E - 5]$	RTACG vs N-RTACG	465	0	$1.862E - 9$
$[-1E - 3, 1E - 3]$	RTACG vs N-RTACG	465	0	$1.862E - 9$

The evolution of the control performance index, related to the sum of the squared position errors $\mathcal{L}_1(t)$ through the time, is illustrated in Fig. 8 with the three different disturbances ζ . It is observed that the control performance index $\mathcal{L}_1(t)$ in the RTACG presents a uniform amplitude through the time with a lower value of peak to peak amplitude than the N-RTACG. In particular, the maximum amplitude of such errors in the RTACG is $2.229E - 5$, $2.598E - 5$, and $4.207E - 4$ with the chosen uncertainties. While in the N-RTACG, the maximum amplitude is $3.672E - 5$, $4.367E - 5$, and $7.358E - 4$. Reductions in the position error of around 64.7%, 68.1%, and 74.9% are presented using the gains obtained by the proposed RTACG. This is attributed to the minimization

of the sensitive in the control performance index $(\partial\hat{\mathcal{L}}_1(t)/\partial x_i \forall i = 4, 5, 6)$ given
700 in the RTACG, which reduces the negative effects of the uncertainties.

The trajectory tracking in the $X_w - Y_w$ Cartesian plane and in the ϕ_w space
are shown in Fig. 6 and Fig. 7, respectively. Those figures visualize the behavior
of the real OMR prototype with the controller gains obtained by RTACG and
N-RTACG under the three different disturbances ζ . It is important to highlight
705 that, although both figures have a similar trajectory tracking using the gains of
both tuning approaches, the control performance index $\hat{\mathcal{L}}_1(t)$ in the proposed
RTACG presents a remarkable reduction around of [64.7%, 74.9%], and also,
with a more uniform variation concerning the N-RTACG (see Fig. 8).

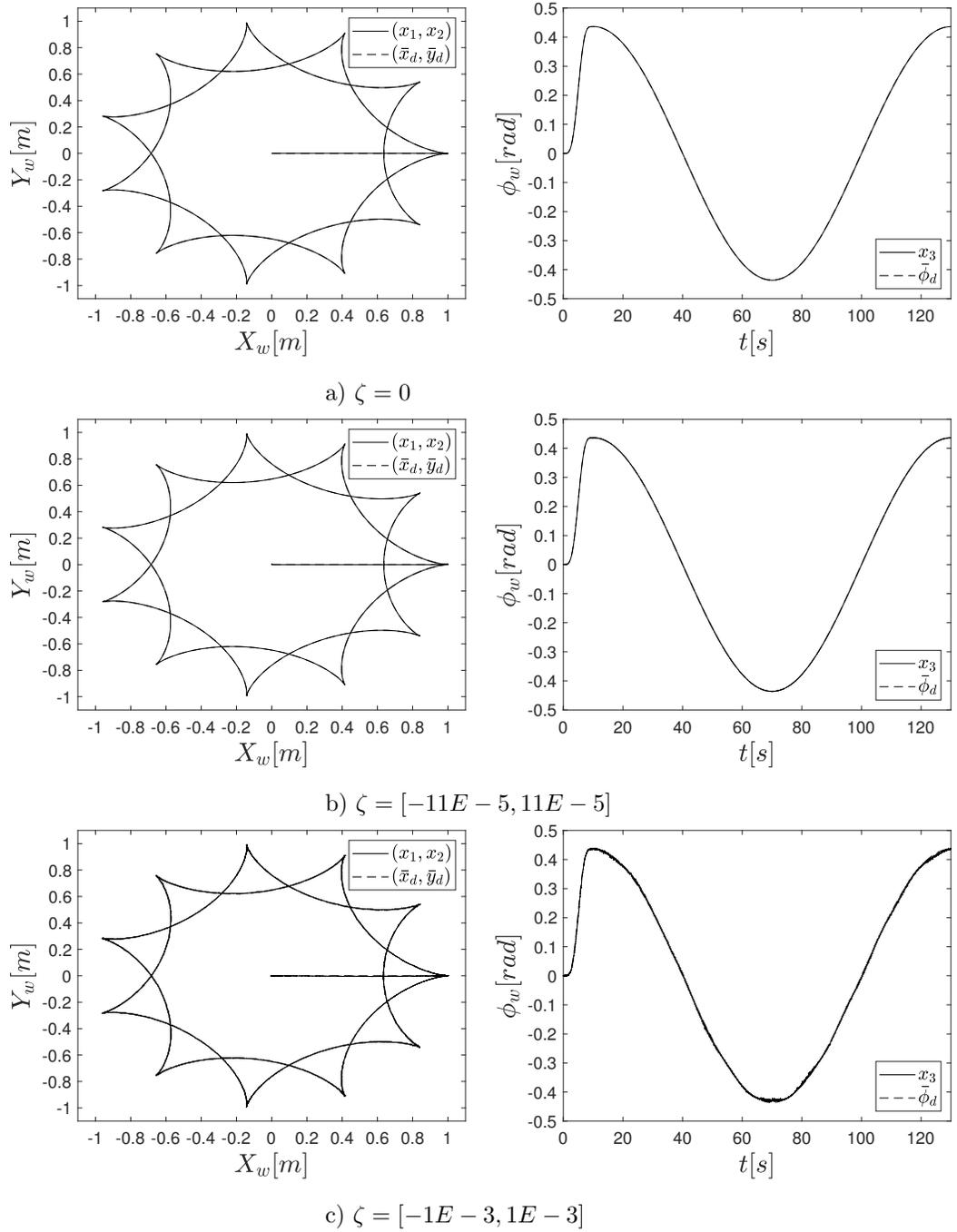


Figure 6: Experimental results of the OMR behavior in the trajectory tracking under different disturbances ζ using the obtained gains by the RTACG.

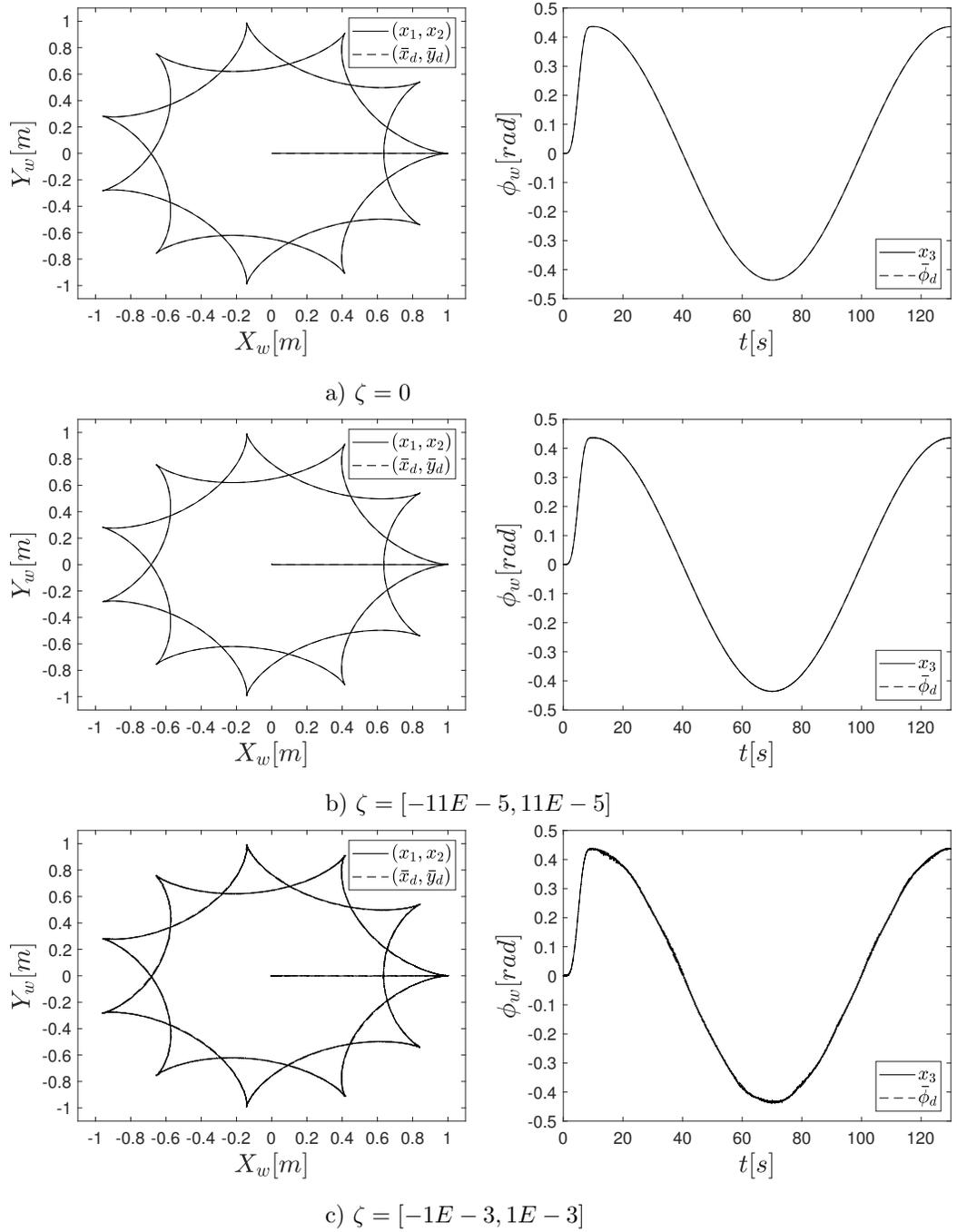


Figure 7: Experimental results of the OMR behavior in the trajectory tracking under different disturbances ζ using the obtained gains by the N-RTACG.

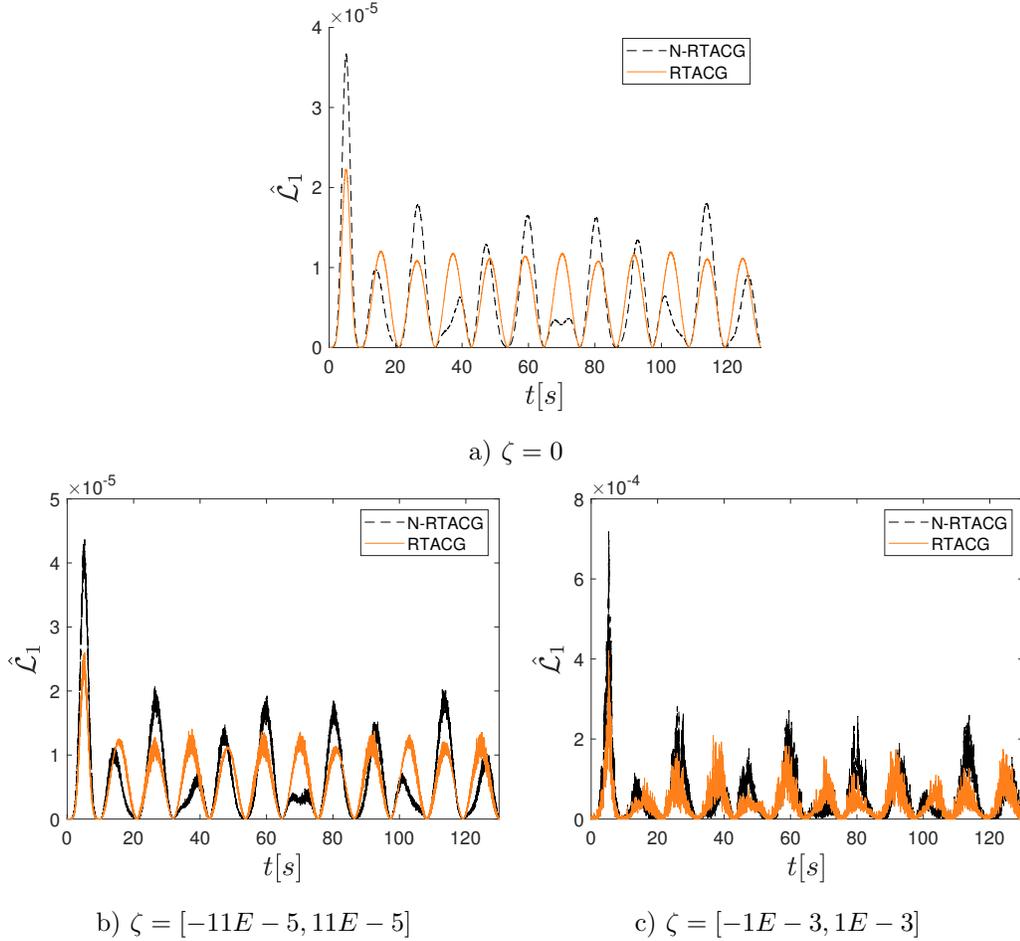


Figure 8: Experimental results of the $\hat{J}_1(t)$ function through the time with different disturbances ζ .

5. Conclusions

710 This paper proposes the Robust Tuning Approach for Controller Gains
 (RTACG) as an offline nonlinear dynamic optimization problem. The self-
 adaptative DE is used to solve the RTACG and provides a competitive and
 reliable performance without requiring an exhaustive algorithm parameter tun-
 ing.

715 The main advantages of the proposal are: *i*) the obtained controller gains

produce control performance indexes as insensible as possible to uncertainty variations without knowing the uncertainty variation bounds, and *ii*) it is not required the careful selection of the controller gain bounds to achieve real experimental results (laboratory tests). Those advantages can be useful for practical
720 purposes.

The proposal is applied to the task-space PD controller of an Omnidirectional Mobile Robot, where the position error, the energy consumption, and the sensitivity of the position error concerning the velocity variations, are minimized. The simulation and experimental results show that with the control
725 gains obtained by the proposed RTACG, the position errors are less sensitive to the effects of velocity variations (uncertainties). Hence, the closed-loop system performance is significantly improved concerning a non-robust tuning approach.

In the results of the proposed RTACG compared with a non-robust tuning approach, simulation results indicate that the improvement is around 17.08%,
730 16.98%, and 1100% using different magnitudes of uncertainties. Besides, the inferential statistics in the experimental results confirm that, with the confidence of 95%, the proposed RTACG presents a better closed-loop system performance under the effects of different uncertainties. In particular, reductions around 64.7%, 68.1%, and 74.9% in the position errors are attained in the experimental
735 results with different magnitudes of uncertainties.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

740 Acknowledgments

The authors acknowledge the support of the Secretaría de Investigación y Posgrado (SIP) under Grants 20190239, 20210374 and 20200150; and in part by the Dirección de Posgrado, Investigación e Innovación of the Tecnológico

Nacional de México through the projects No. 11013. The first author acknowl-
745 edges the support from the Mexican Consejo Nacional de Ciencia y Tecnología
(CONACyT) through a scholarship to pursue his graduate studies at CIDETEC-
IPN.

References

- [1] S. C. Park, The fourth industrial revolution and implications for innovative
750 cluster policies, *AI & Society* 33 (3) (2018) 433–445. doi:<https://doi.org/10.1007/s00146-017-0777-5>.
- [2] L. Sciavicco, B. Siciliano, *Modelling and control of robot manipulators*,
Springer-Verlag London UK, 2000.
- [3] M. W. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*,
755 New York: Wiley, 2005.
- [4] P. R. Daugherty, H. J. Wilson, *Human + machine: Reimagining work in
the age of AI*, Harvard Business Press, 2018.
- [5] C. Caudwell, C. Lacey, What do home robots want? the ambivalent power
of cuteness in robotic relationships, *Convergence* 26 (4) (2020) 956–968.
760 doi:<https://doi.org/10.1177/1354856519837792>.
- [6] L. Bai, J. Guan, X. Chen, J. Hou, W. Duan, An optional passive/active
transformable wheel-legged mobility concept for search and rescue robots,
Robotics and Autonomous Systems 107 (2018) 145 – 155. doi:<https://doi.org/10.1016/j.robot.2018.06.005>.
- 765 [7] X.-B. Jin, T.-L. Su, J.-L. Kong, Y.-T. Bai, B.-B. Miao, C. Dou, State-
of-the-art mobile intelligence: Enabling robots to move like humans by
estimating mobility with artificial intelligence, *Applied Sciences* 8 (3). doi:
<https://doi.org/10.3390/app8030379>.

- 770 [8] A. Deshpande, J. Luntz, Behaviors for physical cooperation between robots for mobility improvement, *Autonomous Robots* 23 (4) (2007) 259–274. doi:
<https://doi.org/10.1007/s10514-007-9044-9>.
- [9] H. Seraji, Reachability analysis for base placement in mobile manipulators, *Journal of Robotic Systems* 12 (1) (1995) 29–43. doi:<https://doi.org/10.1002/rob.4620120104>.
- 775 [10] X. Chen, Y. Chen, J. Chase, Mobile robots-state of the art in land, sea, air, and collaborative missions, 2009.
- [11] G. Diaz-Arango, H. Vázquez-Leal, L. Hernandez-Martinez, M. T. S. Pascual, M. Sandoval-Hernandez, Homotopy path planning for terrestrial robots using spherical algorithm, *IEEE Transactions on Automation Science and Engineering* 15 (2) (2018) 567–585. doi:<https://doi.org/10.1109/TASE.2016.2638208>.
- 780 [12] K. Alipour, P. Daemi, A. Hassanpour, B. Tarvirdizadeh, On the capability of wheeled mobile robots for heavy object manipulation considering dynamic stability constraints, *Multibody system dynamics* 41 (2) (2017) 101–123. doi:<https://doi.org/10.1007/s11044-017-9563-x>.
- 785 [13] B. Hichri, L. Adouane, J.-C. Fauroux, Y. Mezouar, I. Doroftei, Cooperative mobile robot control architecture for lifting and transportation of any shape payload, in: *Distributed Autonomous Robotic Systems*, 2016, pp. 177–191.
- [14] K. Dörfler, T. Sandy, M. Gifftthaler, F. Gramazio, M. Kohler, J. Buchli, Mobile robotic brickwork, in: *Robotic Fabrication in Architecture, Art and Design 2016*, 2016, pp. 204–217. doi:https://doi.org/10.1007/978-3-319-26378-6_15.
- 790 [15] G. A. Cardona, C. Bravo, W. Quesada, D. Ruiz, M. Obeng, X. Wu, J. M. Calderon, Autonomous navigation for exploration of unknown environments and collision avoidance in mobile robots using reinforcement
- 795

- learning, in: 2019 SoutheastCon, 2019, pp. 1–7. doi:<https://doi.org/10.1109/SoutheastCon42311.2019.9020521>.
- [16] A. V. Savkin, H. Li, A safe area search and map building algorithm for a wheeled mobile robot in complex unknown cluttered environments 36 (1) (2018) 96–118. doi:<https://doi.org/10.1017/S0263574717000194>.
- [17] M. Ciszewski, M. Giergiel, T. Buratowski, P. Małka, Modeling and Control of a Tracked Mobile Robot for Pipeline Inspection, 2020. doi:<https://doi.org/10.1007/978-3-030-42715-3>.
- [18] Z. Chen, F. Gao, Q. Sun, Y. Tian, J. Liu, Y. Zhao, Ball-on-plate motion planning for six-parallel-legged robots walking on irregular terrains using pure haptic information, Mechanism and Machine Theory 141 (2019) 136 – 150. doi:<https://doi.org/10.1016/j.mechmachtheory.2019.07.009>.
- [19] W. Chung, K. Iagnemma, Wheeled robots, 2016, pp. 575–594. doi:https://doi.org/10.1007/978-3-319-32552-1_24.
- [20] M. S. Masmoudi, N. Krichen, M. Masmoudi, N. Derbel, Fuzzy logic controllers design for omnidirectional mobile robot navigation, Applied soft computing 49 (2016) 901–919. doi:<https://doi.org/10.1016/j.asoc.2016.08.057>.
- [21] M. O. Tătar, F. Haiduc, D. Mândru, Design of a synchro-drive omnidirectional mini-robot, in: Mechatronic Systems and Materials VI, Vol. 220, 2015, pp. 161–167. doi:<https://doi.org/10.4028/www.scientific.net/SSP.220-221.161>.
- [22] C.-C. Tsai, C.-F. Hsu, C.-W. Wu, F.-C. Tai, Cooperative localization using fuzzy ddeif and broad learning system for uncertain heterogeneous omnidirectional multi-robots, International Journal of Fuzzy Systems 21 (8) (2019) 2542–2555. doi:<https://doi.org/10.1007/s40815-019-00739-2>.
- [23] W. Li, C. Yang, Y. Jiang, X. Liu, C.-Y. Su, Motion planning for omnidirectional wheeled mobile robot by potential field method, Jour-

- nal of Advanced Transportation 2017. doi:<https://doi.org/10.1007/s40815-019-00739-2>.
- 825
- [24] S. Morales, J. Magallanes, C. Delgado, R. Canahuire, Lqr trajectory tracking control of an omnidirectional wheeled mobile robot, in: 2018 IEEE 2nd Colombian Conference on Robotics and Automation (CCRA), 2018, pp. 1–5. doi:<https://doi.org/10.1109/CCRA.2018.8588146>.
- 830 [25] C. Ren, X. Li, X. Yang, S. Ma, Extended state observer-based sliding mode control of an omnidirectional mobile robot with friction compensation, IEEE Transactions on Industrial Electronics 66 (12) (2019) 9480–9489. doi:<https://doi.org/10.1109/TIE.2019.2892678>.
- [26] L. Ovalle, H. Ríos, M. Llama, V. Santibáñez, A. Dzul, Omnidirectional mobile robot robust tracking: Sliding-mode output-based control approaches, Control Engineering Practice 85 (2019) 50–58. doi:<https://doi.org/10.1016/j.conengprac.2019.01.002>.
- 835
- [27] L. Desborough, R. Miller, Increasing customer value of industrial control performance monitoring - honeywell’s experience, in: AIChE symposium series, no. 326, New York, American Institute of Chemical Engineers, 1998, 2002, pp. 169–189.
- 840
- [28] G. Liu, S. Daley, Optimal-tuning pid control for industrial systems, Control Engineering Practice 9 (11) (2001) 1185–1194. doi:[https://doi.org/10.1016/S0967-0661\(01\)00064-8](https://doi.org/10.1016/S0967-0661(01)00064-8).
- [29] M. Arahal, M. Duran, Pi tuning of five-phase drives with third harmonic injection, Control Engineering Practice 17 (7) (2009) 787–797. doi:<https://doi.org/10.1016/j.conengprac.2008.12.005>.
- 845
- [30] M. G. Villarreal-Cervantes, J. Alvarez-Gallegos, Off-line pid control tuning for a planar parallel robot using de variants, Expert Systems with Applications 64 (2016) 444 – 454. doi:<https://doi.org/10.1016/j.eswa.2016.08.013>.
- 850

- [31] E. Alcalá, V. Puig, J. Quevedo, T. Escobet, R. Comasolivas, Autonomous vehicle control using a kinematic lyapunov-based technique with lqr-lmi tuning, *Control Engineering Practice* 73 (2018) 1–12. doi:<https://doi.org/10.1016/j.conengprac.2017.12.004>.
855
- [32] K. Åström, T. Hägglund, Revisiting the ziegler–nichols step response method for pid control, *Journal of Process Control* 14 (6) (2004) 635–650. doi:<https://doi.org/10.1016/j.jprocont.2004.01.002>.
- [33] K. Nisi, B. Nagaraj, A. Agalya, Tuning of a pid controller using evolutionary multi objective optimization methodologies and application to the pulp and paper industry, *International Journal of Machine Learning and Cybernetics* 10 (8) (2019) 2015–2025. doi:<https://doi.org/10.1007/s13042-018-0831-8>.
860
- [34] J. M. S. Ribeiro, M. F. Santos, M. J. Carmo, M. F. Silva, Comparison of pid controller tuning methods: analytical/classical techniques versus optimization algorithms, in: 2017 18th International Carpathian Control Conference (ICCC), 2017, pp. 533–538. doi:<https://doi.org/10.1109/CarpathianCC.2017.7970458>.
865
- [35] A. Rodríguez-Molina, E. Mezura-Montes, M. G. Villarreal-Cervantes, M. Aldape-Pérez, Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem, *Applied Soft Computing* 93 (2020) 106342. doi:<https://doi.org/10.1016/j.asoc.2020.106342>.
870
- [36] G. Wen, S. S. Ge, C. L. P. Chen, F. Tu, S. Wang, Adaptive tracking control of surface vessel using optimized backstepping technique, *IEEE Transactions on Cybernetics* 49 (9) (2019) 3420–3431. doi:[10.1109/TCYB.2018.2844177](https://doi.org/10.1109/TCYB.2018.2844177).
875
- [37] V. H. Alves Ribeiro, G. R. Meza, Multi-objective pid controller tuning for an industrial gasifier, in: 2018 IEEE Congress on Evolutionary Compu-

- 880 tation (CEC), 2018, pp. 1–6. doi:<https://doi.org/10.1109/CEC.2018.8477957>.
- [38] H. V. Hultmann Ayala, L. dos Santos Coelho, Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator, *Expert Systems with Applications* 39 (10) (2012) 8968–8974. doi:<https://doi.org/10.1016/j.eswa.2012.02.027>.
885
- [39] B. Doicin, M. Popescu, C. Patrascioiu, Pid controller optimal tuning, in: 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2016, pp. 1–4. doi:<https://doi.org/10.1109/ECAI.2016.7861175>.
- 890 [40] R.-E. Precup, R.-C. David, E. M. Petriu, A.-I. Szedlak-Stinean, C.-A. Bojan-Dragos, Grey wolf optimizer-based approach to the tuning of pi-fuzzy controllers with a reduced process parametric sensitivity, *IFAC-PapersOnLine* 49 (5) (2016) 55–60, 4th IFAC Conference on Intelligent Control and Automation Sciences/ICONS 2016. doi:<https://doi.org/10.1016/j.ifacol.2016.07.089>.
895
- [41] G. Reynoso-Meza, J. Sanchis, X. Blasco, M. Martínez, Algoritmos evolutivos y su empleo en el ajuste de controladores del tipo pid: Estado actual y perspectivas, *Revista Iberoamericana de Automática e Informática Industrial RIAI* 10 (3) (2013) 251 – 268. doi:<https://doi.org/10.1016/j.riai.2013.04.001>.
900
- [42] G. Reynoso-Meza, X. Blasco, J. Sanchis, M. Martínez, Controller tuning using evolutionary multi-objective optimisation: Current trends and applications, *Control Engineering Practice* 28 (2014) 58 – 73. doi:<https://doi.org/10.1016/j.conengprac.2014.03.003>.
- 905 [43] T. Bäck, D. B. Fogel, Z. Michalewicz, *Handbook of evolutionary computation*, 1st Edition, IOP Publishing Ltd. and Oxford University Press, 1997.

- [44] R. C. Eberhart, Y. Shi, J. Kennedy, Swarm intelligence, 1st Edition, Morgan Kaufmann Publishers, 2001. doi:<https://doi.org/10.1016/B978-1-55860-595-4.X5000-1>.
- 910 [45] R. Sadeghian, M. T. Masouleh, Controller tuning based on optimization algorithms of a novel spherical rolling robot, Journal of Mechanical Science and Technology 30 (11) (2016) 5207–5216. doi:<https://doi.org/10.1007/s12206-016-1038-0>.
- [46] A. Aouf, L. Boussaid, A. Sakly, A pso algorithm applied to a pid controller
915 for motion mobile robot in a complex dynamic environment, in: 2017 International Conference on Engineering & MIS (ICEMIS), 2017, pp. 1–7. doi:<https://doi.org/10.1109/ICEMIS.2017.8273012>.
- [47] H. Huang, Intelligent motion controller design for four-wheeled omnidirectional mobile robots using hybrid ga-pso algorithm, in: 2011 IEEE International
920 Conference on Systems, Man, and Cybernetics, 2011, pp. 2267–2272. doi:<https://doi.org/10.1109/ICSMC.2011.6084015>.
- [48] O. Serrano-Pérez, M. G. Villarreal-Cervantes, J. C. González-Robles, A. Rodríguez-Molina, Meta-heuristic algorithms for the control tuning of omnidirectional mobile robots, Engineering Optimization 52 (2) (2020)
925 325–342. doi:<https://doi.org/10.1080/0305215X.2019.1585834>.
- [49] S. Leyffer, M. Menickelly, T. Munson, C. Vanaret, S. M. Wild, A survey of nonlinear robust optimization, INFOR: Information Systems and Operational Research 58 (2) (2020) 342–373. doi:<https://doi.org/10.1080/03155986.2020.1730676>.
- 930 [50] S. Salomon, Active Robust Optimization, Active Robust Optimization: Optimizing for Robustness of Changeable Products, 2019, pp. 57–86. doi:https://doi.org/10.1007/978-3-030-15050-1_3.
- [51] H.-G. Beyer, B. Sendhoff, Robust optimization – a comprehensive survey,

- Computer Methods in Applied Mechanics and Engineering 196 (33) (2007)
935 3190 – 3218. doi:<https://doi.org/10.1016/j.cma.2007.03.003>.
- [52] D. H. Jung, B. C. Lee, Development of a simple and efficient method for robust optimization, International Journal for Numerical Methods in Engineering 53 (9) (2002) 2201–2215. doi:<https://doi.org/10.1002/nme.383>.
- 940 [53] D. Buche, P. Stoll, R. Dornberger, P. Koumoutsakos, Multiobjective evolutionary algorithm for the optimization of noisy combustion processes, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 32 (4) (2002) 460–473. doi:<https://doi.org/10.1109/TSMCB.2002.804372>.
- 945 [54] E. J. Hughes, Constraint handling with uncertain and noisy multi-objective evolution, in: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Vol. 2, 2001, pp. 963–970. doi:<https://doi.org/10.1109/CEC.2001.934294>.
- [55] K.-H. Lee, G.-J. Park, Robust optimization considering tolerances of design
950 variables, Computers & Structures 79 (1) (2001) 77–86. doi:[https://doi.org/10.1016/S0045-7949\(00\)00117-6](https://doi.org/10.1016/S0045-7949(00)00117-6).
- [56] M. K. Shirani Faradonbeh, A. Tewari, G. Michailidis, Input perturbations for adaptive control and learning, Automatica 117 (2020) 108950. doi:<https://doi.org/10.1016/j.automatica.2020.108950>.
- 955 [57] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, J. Álvarez Gallegos, M. Aldape-Pérez, Bio-inspired adaptive control strategy for the highly efficient speed regulation of the dc motor under parametric uncertainty, Applied Soft Computing 75 (2019) 29–45. doi:<https://doi.org/10.1016/j.asoc.2018.11.002>.
- 960 [58] A. Rodríguez-Molina, M. G. Villarreal-Cervantes, M. Aldape-Pérez, Indirect adaptive control using the novel online hypervolume-based differen-

tial evolution for the four-bar mechanism, *Mechatronics* 69 (2020) 102384.
doi:<https://doi.org/10.1016/j.mechatronics.2020.102384>.

- [59] R. Moliner, R. Tanda, Herramienta para la sintonía robusta de controladores PI/PID de dos grados de libertad, *Revista Iberoamericana de Automática e Informática industrial* 13 (1) (2016) 22–31. doi:<https://doi.org/10.1016/j.riai.2015.05.003>.
965
- [60] K. J. Åström, H. Panagopoulos, T. Hägglund, Design of pi controllers based on non-convex optimization, *Automatica* 34 (5) (1998) 585–601.
970 doi:[https://doi.org/10.1016/S0005-1098\(98\)00011-9](https://doi.org/10.1016/S0005-1098(98)00011-9).
- [61] A. Osyczka, *Multicriterion optimization in engineering with FORTRAN programs*, Chichester, UK: Ellis Horwood–Wiley, 1984.
- [62] G. Karer, I. Škrjanc, Interval-model-based global optimization framework for robust stability and performance of pid controllers, *Applied Soft Computing* 40 (2016) 526–543. doi:<https://doi.org/10.1016/j.asoc.2015.11.046>.
975
- [63] C. C. de Wit, B. Siciliano, G. Bastin, *Theory of Robot Control*, 1st Edition, Communications and Control Engineering, Springer, London, 1996. doi:<https://doi.org/10.1007/978-1-4471-1501-4>.
- [64] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*, 2nd Edition, *Advances in Design and Control*, SIAM, 2010.
980
- [65] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *Journal of global optimization* 11 (4) (1997) 341–359. doi:<https://doi.org/10.1023/A:1008202821328>.
985
- [66] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: A comparative study on nu-

- merical benchmark problems, *IEEE Transactions on Evolutionary Computation* 10 (6) (2006) 646–657. doi:<https://doi.org/10.1109/TEVC.2006.872133>.
990
- [67] J. Liu, J. Lampinen, A fuzzy adaptive differential evolution algorithm, *Soft Computing* 9 (6) (2005) 448–462. doi:<https://doi.org/10.1007/s00500-004-0363-x>.
- 995 [68] P. Rakshit, A. Konar, S. Das, Noisy evolutionary optimization algorithms – a comprehensive survey, *Swarm and Evolutionary Computation* 33 (2017) 18 – 45. doi:<https://doi.org/10.1016/j.swevo.2016.09.002>.
- [69] R. W. Morrison, *Designing evolutionary algorithms for dynamic environments*, Springer Science & Business Media, 2004.
- 1000 [70] E. Juárez-Castillo, H.-G. Acosta-Mesa, E. Mezura-Montes, Empirical study of bound constraint-handling methods in particle swarm optimization for constrained search spaces, in: *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 604–611. doi:<https://doi.org/10.1109/CEC.2017.7969366>.
- 1005 [71] J. Brest, M. S. Maučec, Population size reduction for the differential evolution algorithm, *Applied Intelligence* 29 (3) (2008) 228–247. doi:<https://doi.org/10.1007/s10489-007-0091-x>.
- [72] K. R. Opara, J. Arabas, Differential evolution: A survey of theoretical analyses, *Swarm and Evolutionary Computation* 44 (2019) 546–558. doi:
1010 <https://doi.org/10.1016/j.swevo.2018.06.010>.
- [73] J. Derrac, S. García, D. Molina, F. Herrera, 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 (1) (2011) 3 – 18. doi:<https://doi.org/10.1016/j.swevo.2011.02.002>.
1015

- [74] M. G. Villarreal-Cervantes, J. F. Guerrero-Castellanos, S. Ramírez-Martínez, J. P. Sánchez-Santana, Stabilization of a (3,0) mobile robot by means of an event-triggered control, *ISA Transactions* 58 (2015) 605 – 613. doi:<https://doi.org/10.1016/j.isatra.2015.06.013>.