Introducing the Fractional Order Robotic Darwinian PSO

Micael S. Couceiro[1,3], Fernando M. L. Martins[2]
Rui P. Rocha[1] and Nuno M. F. Ferreira[3]

[1]*Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal,*
{micaelcouceiro, rprocha}@isr.uc.pt
[2]*Instituto de Telecomunicações (Covilhã), Coimbra College of Education, 3030-329 Coimbra, Portugal*
fmlmartins@esec.pt
[3]*Department of Electrotechnics Engineering, Coimbra Institute of Engineering, 3030-199 Coimbra, Portugal*
{micael, nunomig}@isec.pt

Abstract. The Darwinian Particle Swarm Optimization (*DPSO*) is an evolutionary algorithm that extends the Particle Swarm Optimization using natural selection to enhance the ability to escape from sub-optimal solutions. An extension of the *DPSO* to multi-robot applications has been recently proposed and denoted as Robotic Darwinian *PSO* (*RDPSO*), benefiting from the dynamical partitioning of the whole population of robots, hence decreasing the amount of required information exchange among robots. This paper further extends the previously proposed algorithm using fractional calculus concepts to control the convergence rate, while considering the robot dynamical characteristics. Moreover, to improve the convergence analysis of the *RDPSO*, an adjustment of the fractional coefficient based on mobile robot constraints is presented and experimentally assessed with 2 real platforms. Afterwards, this novel fractional-order *RDPSO* is evaluated in 12 physical robots being further explored using a larger population of 100 simulated mobile robots within a larger scenario. Experimental results show that changing the fractional coefficient does not significantly improve the final solution but presents a significant influence in the convergence time because of its inherent memory property.

Keywords: Fractional Calculus; Evolutionary Algorithm; Swarm Robotics; Foraging; Convergence Analysis.
PACS: 87.85.St; 02.60.Lj.

INTRODUCTION

One of the most well-known bio-inspired algorithms from swarm intelligence is the Particle Swarm Optimization (*PSO*), which basically consists of a technique loosely inspired by birds flocking in search of food [1]. More specifically, it encompasses a number of particles that collectively move on the search space to find the optimal solution. A problem with *PSO* algorithm is that of becoming trapped in sub-optimal solutions. Therefore, the *PSO* may work perfectly on one problem but may fail on another. In order to overcome this problem, many authors have suggested extended versions of the *PSO*, such as the Darwinian Particle Swarm Optimization (*DPSO*) [2], to enhance the ability to escape from sub-optimal solutions (*cf.*, [3]). An extension of the *DPSO* to multi-robot applications has been recently proposed and denoted as Robotic Darwinian *PSO* (*RDPSO*), benefiting from the dynamical partitioning of the whole population of robots [4] and [5]. Hence, the *RDPSO* allows decreasing the amount of required information exchange among robots and therefore is scalable to large populations of robots.

From Optimization to Robotics

The navigation of groups of robots, especially swarm robots, has been one of the fields that has benefited from biological inspiration [6]. However, real multi-robot systems present several constraints that need to be considered. For instance, the development of robot teams for surveillance or rescue missions require that robots have to be able to maintain communication among them without the aid of a communication infrastructure. One of the first adapted versions of the *PSO* to handle real world constraints, such as obstacles, is presented by Min *et al.* [7]. Similarly to the *RDPSO*, this approach adjusts the velocity and direction of the mobile robot in real time, thus allowing the robot to reach its goal while avoiding obstacles in the way. Each robot runs an entire swarm and the global best particle is considered the best solution. Unfortunately, contrarily to our previous work [4], simulation results presented the comparison with Artificial Potential Field (*APF*) algorithms with only one robot. Also, simulation experiments lack some information such as the distance the robot needs to travel (since the time which the mobile robot spends in reaching the goal is presented). Another similar approach was developed by Pugh & Martinoli [8] where an adapted version of the *PSO* to distributed unsupervised robotic learning in groups of robots with only local information is presented. The main difference between this algorithm and classical *PSO* is that each robot (*i.e.*, particle) only takes into consideration the information of the robots within a fixed radius r (omnidirectional communication). The authors analyzed how the performance was affected if the standard *PSO* neighborhood structure was adapted to a more closely model, which is possible in a real robot group with limited communication abilities. Experimental results obtained using *Webots* simulator showed that the adapted version of the *PSO* maintained good performance for groups of robots of various sizes when compared to other bio-inspired methods. However, contrarily to the previously presented *RDPSO* algorithm [5], all bio-inspired methods used, including the adapted *PSO*, tends to get trapped in local solutions. Furthermore, and contrarily to the experimental results with real platforms shown in our previous work [9], the authors does not use multi-hop connectivity and does not apply any kind of algorithm to enforce communication between robots. Similarly, Hereford and Siebold [10] presented an embedded version of the *PSO* in swarm platforms. As in *RDPSO*, there is no central agent to coordinate the robots movements or actions. Despite the potentialities of the physically-embedded *PSO*, experimental results were carried out using a population of only three robots, performing a distributed search in a scenario without local solutions. Also, contrarily to our previous work [11], collision avoidance and fulfillment of mobile ad hoc network (*MANET*) connectivity were not considered.

Statement of Contribution

In our previous work [12], the fractional calculus (*FC*) concept was used to improve the convergence rate of the traditional *DPSO* presented by Tillett *et al.* [2]. Experimental results showed that, although the speed of convergence of the *Fractional Order DPSO* (*FODPSO*) depends on the fractional order α, the proposed algorithm outperformed the traditional *DPSO* and *PSO*, as well as the *FOPSO* previously presented in the literature [13]. This work further extends the *RDPSO* previously presented in [4] and [5] using fractional calculus to control the convergence rate of robots toward the optimal solution. Section two introduces some preliminary concepts to pave the way for section three that generalizes the *RDPSO* to a fractional order, denoting it as *Fractional Order RDPSO* (*FORDPSO*). Experimental results for the *FORDPSO* are presented in section four with both physical and simulated platforms. Finally, in section five outlines the main conclusions.

PRELIMINARIES

For better clarity in the presented paper, this section introduces the fractional calculus and gives a brief overview of the working principles behind the *RDPSO* algorithm previously proposed in [4] and further extended in [5].

Fractional Calculus

Fractional calculus (*FC*) has attracted the attention of several researchers [14], being applied in various scientific fields, such as engineering, computational mathematics, fluid mechanics, among others.

FC can be considered as a generalization of integer-order calculus, thus accomplishing what integer-order calculus cannot. As a natural extension of the integer (*i.e.*, classical) derivatives, fractional derivatives provide an excellent instrument for the description of memory and hereditary properties of processes. The concept of *Grünwald–Letnikov* fractional differential, is presented by the following definition.

Definition 1 [15] *Let* Γ *be the gamma function defined as:*

$$\Gamma(k) = (k-1)! \tag{1}$$

The signal $D^{\alpha}[x(t)]$ *given by*

$$D^{\alpha}[x(t)] = \lim_{h\to 0}\left[\frac{1}{h^{\alpha}}\sum_{k=0}^{+\infty}\frac{(-1)^{k}\Gamma(\alpha+1)x(t-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)}\right], \tag{2}$$

is said to be the ***Grünwald–Letnikov fractional derivative of order*** $\boldsymbol{\alpha}$, $\alpha \in \mathbb{C}$, *of the signal* $x(t)$.

An important property revealed by (2) is that while an integer-order derivative just implies a finite series, the fractional-order derivative requires an infinite number of terms. Therefore, integer derivatives are "local" operators while fractional derivatives have, implicitly, a "memory" of all past events. However, the influence of past events decreases over time.

The formulation in (2) inspires a discrete time calculation presented by the following definition.

Definition 2 [15] *The signal $D^{\alpha}[x(t)]$ given by*

$$D^{\alpha}\big[x[t]\big] = \frac{1}{T^{\alpha}}\sum_{k=0}^{r}\frac{(-1)^{k}\Gamma[\alpha+1]x[t-kT]}{\Gamma[k+1]\Gamma[\alpha-k+1]}, \quad (3)$$

where T is the sampling period and r is the truncation order, is the ***approximate discrete time Grünwald–Letnikov fractional difference of order α****, $\alpha \in \mathbb{C}$, of the discrete signal $x[t]$.*

The series presented in (3) can be implemented by a rational fraction expansion which leads to a superior compromise in what concerns the number of terms versus the quality of the approximation. Nevertheless, since this study focuses on the convergence of robots toward a given solution considering past events, the simple series approximation is adopted.

That being said, it is possible to extend an integer discrete difference, *i.e.*, classical discrete difference, to a fractional-order one, using the following definition.

Definition 3 [16] *The classical integer "direct" discrete difference of signal $x[t]$ is defined as follows:*

$$\Delta^{d}x[t] = \begin{cases} x[t] & , d = 0 \\ x[t] - x[t-1] & , d = 1, \\ \Delta^{d-1}x[t] - \Delta^{d-1}x[t-1], & d > 1 \end{cases} \quad (4)$$

where $d \in \mathbb{N}_0$ is the order of the integer discrete difference. Hence, one can extend the integer-order $\Delta^{d}x[t]$ assuming that the fractional discrete difference satisfies the following inequalities:

$$d - 1 < \alpha < d. \quad (5)$$

The features inherent to fractional calculus make this mathematical tool well suited to describe many phenomena, such as irreversibility and chaos, because of its inherent memory property. In this line of thought, the dynamic phenomena of a robot's trajectory configure a case where fractional calculus tools fit adequately.

RDPSO

Since the *RDPSO* approach is an adaptation of the *DPSO* to real mobile robots, four general features were proposed: *i*) a novel "punish"-"reward" mechanism to emulate the deletion and creation of robots; *ii*) an obstacle avoidance algorithm to avoid collisions; *iii*) an enforcing multi-hop network connectivity algorithm to ensure that the *MANET* remains connected throughout the mission; *iv*) a novel methodology to establish the initial planar deployment of robots preserving the connectivity of the *MANET* while spreading out the robots as most as possible. The *RDPSO* is then modelled based on the following definition.

Definition 4 [5] *The behavior of robot n is described by the following discrete equations at each discrete time, or iteration, $t \in \mathbb{N}_0$:*

$$v_n[t+1] = wv_n[t] + \sum_{i=1}^{4}\rho_i r_i(\chi_i[t] - x_n[t]), \quad (6)$$

$$x_n[t+1] = x_n[t] + v_n[t+1]. \quad (7)$$

wherein parameters w and ρ_i, $w, \rho_i > 0$ with $i = 1,2,3,4$, assign weights to the inertial influence, the local best (i.e., cognitive component), the global best (i.e., social component), the obstacle avoidance component and the enforcing communication component when determining the new velocity. Coefficients r_i, $i = 1,2,3,4$, are random vectors wherein each component is generally a uniform random number between 0 and 1. The variables $v_n[t]$ and $x_n[t]$ represent the velocity and position vector of robot n, respectively, and $\chi_i[t]$ denotes the best position of the cognitive, social, obstacle and MANET components.

The cognitive $\chi_1[t]$ and social components $\chi_2[t]$ are common in *PSO* algorithm, where $\chi_1[t]$ represents the local best position and $\chi_2[t]$ represents the global best position of robot n. The obstacle avoidance component $\chi_3[t]$ is represented by the position of each robot that optimizes a monotonically decreasing or increasing function $g(x_n[t])$ that describes the distance to a sensed obstacle (*cf.*, [4]). In real-world scenarios, obstacles need to be taken into account and the value of ρ_3 depends on several conditions related with the main objective (*i.e.*, minimize a cost function or maximize a fitness function) and the sensing information (*i.e.*, monotonicity of $g(x_n[t])$). The *MANET* component $\chi_4[t]$ is represented by the position of the nearest neighbor increased by the maximum communication range d_{max} toward robot's current position. A higher ρ_4 may enhance the ability to maintain the network connected ensuring a specific range or signal quality between robots.

Besides all these components, the *RDPSO* is represented by multiple swarms, *i.e.*, several groups of robots that, altogether, form the population. Each swarm individually follows equations (6) and (7) in the solution search and some punish-reward rules governs the whole population of robots based on the concept of *social exclusion* (for more details refer to [4]). In what concerns the *socially excluded robots*, instead of searching for the objective function's global optimum like the other robots in the active swarms, they basically randomly wander in the scenario. This approach improves the algorithm, making it less susceptible of becoming trapped in a local optimum. However, excluded robots are always aware of their individual solution and the global solution of the socially excluded group. Also, having multiple swarms enables a distributed approach, because the network that was previously defined by the whole population of robots is now divided into multiple smaller networks (one for each swarm), thus decreasing the number of nodes (*i.e.*, robots) and the information exchanged between robots of the same network. In other words, robots interaction with other robots is confined to local interactions inside the same group (swarm), thus making *RDPSO* scalable to large populations of robots.

FRACTIONAL ORDER RDPSO

This section presents the extension of the *RDPSO* algorithm using fractional calculus to control the convergence rate of robots. Considering the inertial influence in (6) as $w = 1$, one would obtain:

$$v_n[t+1] = v_n[t] + \sum_{i=1}^{4} \rho_i r_i (\chi_i[t] - x_n[t]). \qquad (8)$$

This expression can be rewritten as:

$$v_n[t+1] - v_n[t] = \sum_{i=1}^{4} \rho_i r_i (\chi_i[t] - x_n[t]). \qquad (9)$$

Hence, $v_n[t+1] - v_n[t]$ corresponds to the discrete version of the fractional difference of order $\alpha = 1$, *i.e.*, the first order integer difference $\Delta^d v_n[t+1]$. Assuming $T = 1$ and based on Definition 2, yields to the following equation:

$$D^{\alpha}\big[v_n[t+1]\big] = \sum_{i=1}^{4} \rho_i r_i (\chi_i[t] - x_n[t]). \qquad (10)$$

Based on *FC* concept and Definition 3, the order of the velocity derivative can be generalized to a real number $0 < \alpha < 1$, thus leading to a smoother variation and a longer memory effect. Therefore, considering the discrete time fractional differential presented on Definition 2, one can rewrite equation (8) as:

$$v_n[t+1] = -\sum_{k=1}^{r} \frac{(-1)^k \Gamma[\alpha+1] v[t+1-kT]}{\Gamma[k+1]\Gamma[\alpha-k+1]} + \sum_{i=1}^{4} \rho_i r_i (\chi_i[t] - x_n[t]). \qquad (11)$$

The *RDPSO* is therefore a particular case of the fractional order *RDPSO* (*FORDPSO*) for $\alpha = 1$ (without "memory").

Memory Complexity

Adding memory to the *RDPSO* algorithm allows improving the convergence rate of robots since each robot will have the information about its preceding actions. Nevertheless, the computational requirements increase linearly with r, *i.e.*, the *FORDPSO* present a $\mathcal{O}(r)$ memory complexity *per* robot. Moreover, it is noteworthy that these kinds of optimization or foraging algorithms present a higher performance as the number of robots increase. Hence, robots should be as simple and low-cost as possible (*i.e.*, swarm robots) which are usually memory limited.

Therefore, the truncation of equation (11) will depend on the requirements of the application and the features of the robot. For instance, for the *eSwarBot* (educative Swarm Robot) platforms previously presented in [17], a $r = 4$ leads to results of the same type than for $r > 4$. Although one could consider the processing power as the main reason to use a limited number of terms, the kinematical features of the platform and the mission requirements also needs to be considered in such a way that one can present the following result.

Proposition 1: *Let δ and v_{max} be the encoders-wheel resolution of robots and the maximum allowed travelled distance between iterations, respectively. If τ is the minimum natural number that verifies the following inequality:*

$$-\frac{(-1)^{\tau}\Gamma[\alpha+1]v_{max}}{\Gamma[\tau+1]\Gamma[\alpha-\tau+1]} < \delta. \qquad (12)$$

Then the FORDPSO equation (11) should be truncated based on δ and v_{max}, in $r = \tau - 1$.

Proof: Let us consider the example of a differential drive robot (*e.g.*, *eSwarBot*). A differential drive robot consists of two independently driven wheels and, usually, a free wheel for stability (*e.g.*, caster wheel). For navigation purposes, the driven wheels are usually equipped with encoders that provide odometry measures. Hence, the major odometry parameter of such mobile robot to drive forward is the radius of the wheels R_{wheel} and the number of pulses from revolution of the wheel $N_{pulses/rev}$.

The kinematical equation of a differential drive robot, while moving forward, can be defined as:

$$pulses = N_{pulses/rev} \times \frac{dist}{2\pi \times R_{wheel}}, \tag{13}$$

where $pulses$ is the number of pulses necessary for the robot to travel a distance of $dist$. Defining $pulses = 1$ we can obtain the minimum distance that a robot can travel at each iteration, *i.e.*, the resolution δ. Hence, an increment of the distance lower than δ would be unfeasible for the robot to travel. Also, one may observe through equation (11), that the relevance of past events, *i.e.*, the $v[t+1-kT]$ term, reduces over time. In other words, from a given term $r = \tau - 1$, the relevance of all previous events before it would be irrelevant as the robot is unable to travel with such accuracy.

■

To clarify the previous result, let us consider the following example.

Example: *Considering the eSwarBot platform, a resolution of $\delta = 2.76$ mm is obtained for a single pulse, taking into account that $R_{wheel} = 21.09$ mm and the combination between encoders-wheel provides $N_{pulses/rev} = 48$ pulses/revolution. Let us consider a maximum travelled distance between two iterations of $v_{max} = 100$ mm, i.e., the robot cannot travel more than 100 mm without any update of the information. Fig. 1 presents the computation of each term of equation (10). As one may observe, a term of $r = 4$ would be enough to represent the FORDPSO dynamics in such conditions as the 5th term returns an increment of 2.73 mm. In other words, the algorithm would present similar results for $r \geq 4$.*

As *eSwarBots* would be the robotic platforms used throughout this work, one will only consider the first $r = 4$ terms of the fractional discrete difference in (11), yielding:

$$v_n[t+1] = \alpha v_t^n + \frac{1}{2}\alpha v_{t-1}^n + \frac{1}{6}\alpha(1-\alpha)v_{t-2}^n + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)v_{t-3}^n + \sum_{i=1}^{4}\rho_i r_i(\chi_i[t] - x_n[t]). \tag{14}$$

Next section presents the convergence analysis of the *FORDPSO* based on the dynamical characteristic of robots.

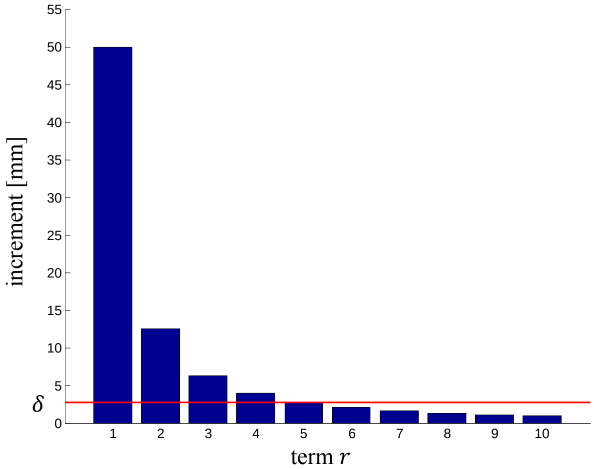

FIGURE 1. Convergence of the robot toward the solution changing the differential derivative r.

Convergence Analysis

Equation (14) represents a stochastic procedure that describes the discrete-time motion of a robot. One way to analyze the convergence of the algorithm consists on adjusting the parameters based on physical mobile robots constraints when facing a better solution. In other words, robots need to softly reduce its velocity (*i.e.*, decelerate) when converging to a given solution. That state is usually unaddressed in the literature while analyzing the traditional *PSO* and its main variants, since virtual agents (*i.e.*, particles) are not constrained by such behaviors.

Let us then suppose that a robot is traveling at a constant velocity such that $v_n[t-k] = v$ with $k \in \mathbb{N}_0$ and it is able to find its equilibrium point in such a way that $x_n[t] = \chi_i$, $i = 1,2,3,4$. In other words, the best position of the cognitive, social, obstacle and *MANET* components are the same. As a result, the robot needs to decelerate until it stops, *i.e.*, $v > v_n[t+1] \geq \cdots \geq v_n[t+k] \geq \cdots \geq 0$.

Consequently, equations (7) and (14) can be rewritten as:

$$0 \leq v\left(\alpha + \frac{1}{2}\alpha + \frac{1}{6}\alpha(1-\alpha) + \frac{1}{24}\alpha(1-\alpha)(2-\alpha)\right) < v, \tag{15}$$

thus resulting in

$$0 < \alpha \leq 0.632. \tag{16}$$

Therefore, one can conclude that $\alpha = 0.632$ is the boundary of the attraction domain, *i.e.*, the *RDPSO* is stable for $0 < \alpha \leq 0.632$ and unstable for $0.632 < \alpha \leq 1$. As a result of the above analysis, the fractional coefficient can be parameterized in such a way that the system's convergence can be controlled by taking into account obstacle avoidance and *MANET* connectivity, without resorting to the definition of any arbitrary or problem-specific parameters.

Nonetheless, to further improve the analysis of robots' behavior and the influence of *FC* in the algorithm, some experiments using two physical *eSwarBots* were carried out. As described in [18] and [19], a swarm behavior can be divided into two activities: *i*) exploitation; and *ii*) exploration. The first one is related with the convergence of the algorithm, thus allowing a good short-term performance. However, if the exploitation level is too high, then the algorithm may be stuck on local solutions. The second one is related with the diversification of the algorithm which allows exploring new solutions, thus improving the long-term performance. However, if the exploration level is too high, then the algorithm may take a long time to find the global solution. As first presented by Shi and Eberhart [20], the trade-off between exploitation and exploration in the classical *PSO* has been commonly handled by systematically adjusting the inertia weight. A large inertia weight improves exploration activity while exploitation is improved using a small inertia weight.

a)

b)

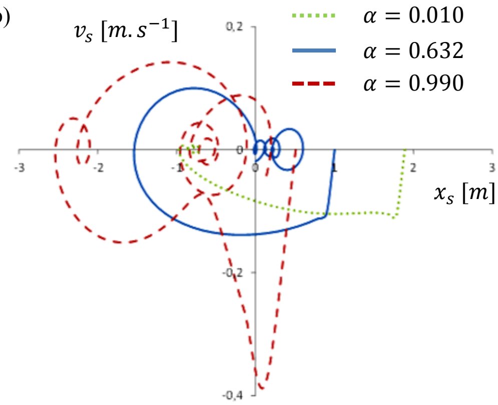

FIGURE 2. Evaluation of the fractional coefficient influence. a) Experimental setup; b) Center-of-mass trajectories in phase space of a swarm of 2 robots.

Since the *FORDPSO* presents a *FC* strategy to control the convergence of the robotic team, the coefficient α needs to be defined in order to provide a high level of exploration while ensuring the global solution of the mission. In order to understand the relation between the fractional coefficient α and the *FORDPSO* exploitation/exploration capabilities, the center-of-mass trajectory in phase space of a swarm of two physical robots, for various values of α, while fixing $\rho_i = 0.5$, will be analyzed. Both robots were randomly placed in the vicinity of the solution in (0,0) with a fixed distance of 0.5 meters between them (Fig. 2a). The solution is defined by an illuminated spot which is sensed using overhead light sensors (*LDR*) (*cf*, EXPERIMENTAL RESULTS section for the experimental setup description).

As it may be perceived (Fig. 2b), the swarm behavior is susceptible to variations in the value of α. Figure 2 depicts that when α is too small, *i.e.*, $\alpha = 0.010$, the exploitation level is too high being likely to get stuck in a local solution. However, the intensification of the algorithm convergence is improved – it presents a quick, almost linear, convergence. When α is at the boundary of the attraction domain, *i.e.*, $\alpha = 0.632$, the trajectory of the swarm is cyclical and presents a good balance between exploitation and exploration. In this case, robots exhibit a level of diversification adequate to avoid local solutions and a considerable level of intensification to converge to the global solution, *i.e.*, it presents a spiral convergence toward a nontrivial attractor. When α is too high, *i.e.*, $\alpha = 0.990$, despite the cyclical trajectory of the swarm toward the global solution, the swarm presents an oscillatory behavior. This results in a high exploration level being more unstable and sometimes unable to converge, *i.e.*, it presents a difficult convergence.

Based on those preliminary experiments, next section evaluates this novel *FORDPSO* using 12 *eSwarBots* being further explored using a larger population of 100 simulated mobile robots within a larger scenario.

EXPERIMENTAL RESULTS

To further validate the claims around the *FORDPSO*, this section provides experimental results obtained using both real and simulated robots.

Real Robots

In this section, it is explored the effectiveness of using the *FORDPSO* on swarms of real robots, while performing a collective foraging task with local and global information under communication constraints. Since the *FORDPSO* is a stochastic algorithm, every time it is executed it may lead to different trajectory convergence. Therefore, a set of 20 trials of 3 minutes each was considered. A minimum (s_{min}), initial (s_{init}) and maximum (s_{max}) number of 1, 2 and 3 swarms were used for a population of $N = 12$ robots.

The experiments were carried out in a 2.55 meters to 2.45 meters scenario. The experimental environment was an enclosed arena that contained two sites (Fig. 3a). Each site was represented by an illuminated spot uniquely identifiable by controlling the brightness of the light. The brighter site (optimal solution) was considered better than the dimmer one (sub-optimal solution), and so the goal of the robots was to collectively choose the brighter site.

The intensity values $F(x, y)$ represented in Fig. 3b were obtained sweeping the whole scenario with a single robot in which the light sensor was connected to a 10-bit analog input, thus offering a resolution of approximately 5 mV. To improve the interpretation of the algorithm performance, results were normalized in a way that the objective of robotic teams is to find the optimal solution of $f(x, y) = 1$.

The maximum communication distance between robots d_{max} was defined as 1.5 meters. At each trial, robots were manually deployed on the scenario in a spiral manner while preserving the maximum communication distance d_{max} (as previously presented in [5]).

In order to evaluate the impact of fractional calculus in the convergence of the algorithm, the original *RDPSO* ($\alpha = 1$) was compared to the fractional order *RDPSO* with $\alpha = 0.632$ (*cf.*, Convergence Analysis Section). Based on our previous work [9] and the presented considerations, Table 1 summarizes the *FORDPSO* configuration.

TABLE 1. *FORDPSO* parameters.

Parameter	Value	
	1st Set	2nd Set
α	1	0.632
Number of Trials	20	
Time per Trial $[sec]$	180	
N_T	12	
s_{min}	1	
s_{init}	2	
s_{max}	3	
d_{max} $[mm]$	1500	
v_{max} $[mm]$	100	
ρ_1	0.1	
ρ_2	0.3	
ρ_3	0.79	
ρ_4	0.79	

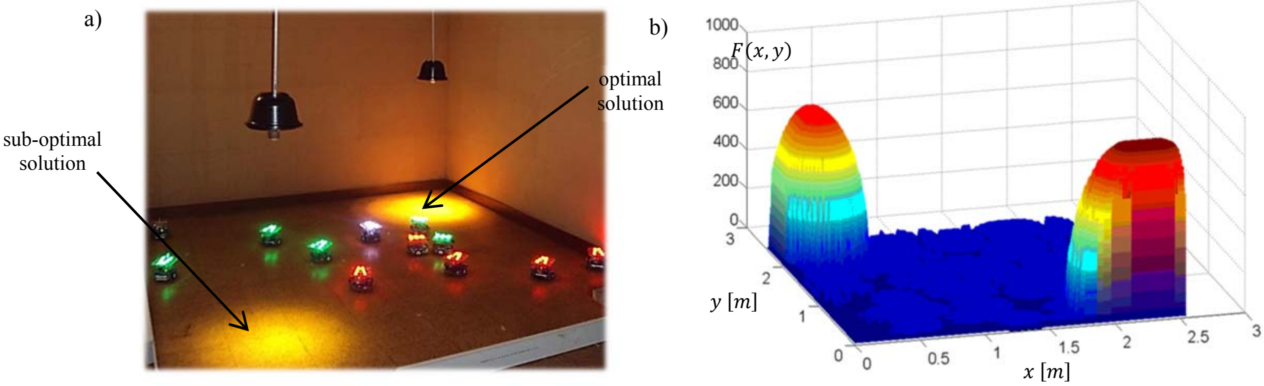

FIGURE 3. Experimental setup. a) Enclosed arena with 2 swarms (different colors); b) Virtual representation of the target distribution.

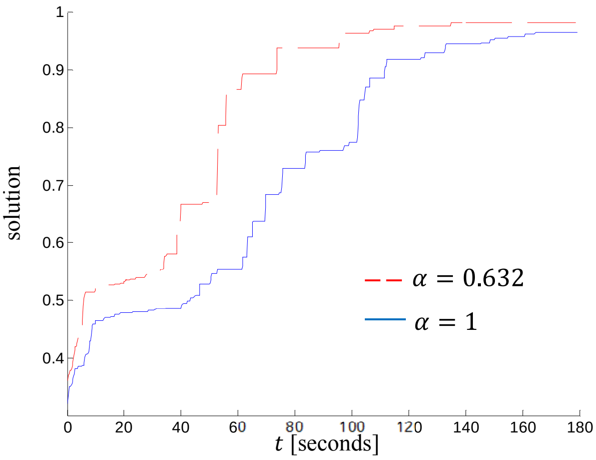

FIGURE 4. Performance of the *FORDPSO* with a population of $N = 12$ robots changing α.

Since these experiments represent a foraging task, it is necessary to evaluate both the completeness of the mission and the time needed to complete it. Therefore, Fig. 4 depicts the convergence of both *RDPSO* and *FORDPSO*. The median of the best solution in the 20 experiments was taken as the final output for both α. As one may observe, the decrease of α from 1 to 0.632 improves the convergence rate of the algorithm also marginally improving the median value of the solution at the end of the mission, *i.e.*, $t = 180$ seconds.

However, analyzing swarm algorithms within small populations of 12 robots may not represent the required collective performance (*cf.*, [21]). Also, it may not be enough to assess the *FORDPSO* performance within the small proposed scenario. Hence, next section presents computational experiments using a larger population of simulated robots within a larger scenario.

Simulated Robots

The use of simulated robots instead of the physical ones was necessary to further evaluate the influence of fractional calculus in the *RDPSO* algorithm. The experiments were carried out in a simulated scenario of 300×300 meters with obstacles randomly deployed at each trial, in which a 2-dimensional benchmark Gaussian functions was defined where *x* and *y*-axis represent the planar coordinates in meters (Fig. 5).

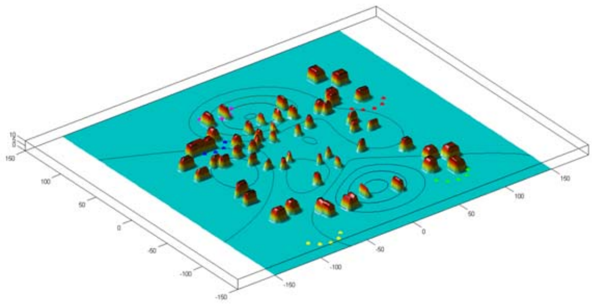

FIGURE 5. Virtual scenario based on a Gaussian distribution endowed with obstacles.

Once again, in order to improve the interpretation of the algorithm performance, results were normalized in a way that the objective of robotic teams was to maximize the function, thus finding the optimal solution of 1, while avoiding obstacles and ensuring the *MANET* connectivity. A set of 100 trials of 500 iterations each was considered for $N = 100$ robots. Also, a minimum (s_{min}), initial (s_{init}) and maximum (s_{max}) number of 2, 5 and 8 swarms were used. The maximum travelled distance between iterations was set as 0.5 meters, *i.e.*, $v_{max} = 0.5$ while the maximum communication distance between robots was set to $d_{max} = 30$ meters.

TABLE 2. *FORDPSO* parameters.

Parameter	Value 1st Set	Value 2nd Set
α	1	0.632
Number of Trials	100	
Number of Iterations	500	
N_T	100	
s_{min}	2	
s_{init}	5	
s_{max}	8	
$d_{max}\ [mm]$	30000	
$v_{max}\ [mm]$	500	
ρ_1	0.1	
ρ_2	0.3	
ρ_3	0.79	
ρ_4	0.79	

Once again, in order to evaluate the impact of fractional calculus in the convergence of the algorithm, the original *RDPSO* ($\alpha = 1$) was compared to the fractional order *RDPSO* with $\alpha = 0.632$. Table 2 summarizes the *FORDPSO* configuration applied in the simulation experiments.

Fig. 6 depicts the convergence of both *RDPSO* and *FORDPSO*. The median of the best solution in the 100 experiments was taken as the final output for both α.

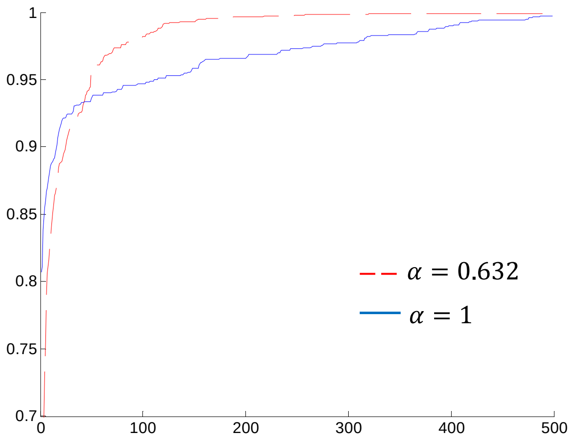

FIGURE 6. Performance of the *FORDPSO* with a population of $N = 100$ robots changing α.

Once again, one may observe that using the integer-order *FORDPSO*, *i.e.*, $\alpha = 1$, present worse results than the the fractional-order $\alpha = 0.632$ (Fig. 6). In other words, one can conclude that, despite both *RDPSO* and *FORDPSO* reveal a similar behavior, the combination between *FC* and Darwin's principles contributes to an improved convergence dynamics.

CONCLUSION

This paper introduces fractional calculus to control the convergence rate of the Robotic Darwinian Particle Swarm Optimization (*RDPSO*) previously proposed. Both memory complexity and convergence analysis of this novel extension, denoted as fractional-order *RDPSO* (*FORDPSO*), are carefully considered based on real robot physical constraints.

Experimental results show that the algorithm converges in most situations regardless on the fractional-order. Nevertheless, the fractional extension of the algorithm presents a considerably superior performance in both time and mission completeness.

One of the future improvements will be the extension of the *FORDPSO* with adaptive mechanisms since robots may need to dynamically change their behavior during the search mission, based on contextual information. Therefore, the *FORDPSO* should be further extended in order to control the swarm susceptibility to the main mission, obstacle avoidance and communication constraint, by systematically adjusting its parameters.

ACKNOWLEDGMENTS

This work was supported by a *PhD* scholarship (*SFRH/BD /73382/2010*) granted to the first author, the research project *CHOPIN* (*PTDC/EEA-CRO/119000/2010*) by the Portuguese Foundation for Science and Technology (*FCT*), the Institute of Systems and Robotics (*ISR*), the Institute of Telecommunications (*IT*) and RoboCorp. Furthermore, this work was made possible by the support and assistance of Rui Mendes that provided the indoor sports pavilion of the Coimbra College of Education (*ESEC*) in which the experiments were carried out. The authors would also like to thank Carlos Figueiredo and Miguel Luz for their cooperation and advice that were vital to fulfill the experiments and other works of technical nature.

REFERENCES

1. J. Kennedy and R. Eberhart, "A new optimizer using particle swarm theory," in *in proceedings of the IEEE Sixth International Symposium on Micro Machine and Human Science*, Nagoya , Japan, 1995, pp. 39-43.
2. J. Tillett, T. M. Rao, F. Sahin, R. Rao, and S. Brockport, "Darwinian Particle Swarm Optimization," in *in proceedings of the 2nd Indian International Conference on Artificial Intelligence*, 2005, pp. 1474-1487.
3. M. S. Couceiro, N. M. F. Ferreira, and J. A. T. Machado, "Fractional Order Darwinian Particle Swarm Optimization," in *3th Symposium on Fractional Signals and Systems, FSS'2011*, Coimbra, Portugal, 2011.
4. M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "A Novel Multi-Robot Exploration Approach based on Particle Swarm Optimization Algorithms," in *in proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR2011*, Kyoto, Japan, 2011, pp. 327-332.
5. M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, "Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Swarms," in *in proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR2011*, Kyoto, Japan, 2011, pp. 284-289.
6. E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press, 1999.
7. H. Q. Min, J. H. Zhu, and X. J. Zheng, "Obstacle Avoidance," in *Proc. Int. Conf. on Machine Learning and Cybernetics*, 2005, pp. 2950-2956.

8. J. Pugh and A. Martinoli, "Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, 2007.
9. M. S. Couceiro, F. M. L. Martins, R. P. Rocha, and N. M. F. Ferreira, "Analysis and Parameter Adjustment of the RDPSO - Towards an Understanding of Robotic Network Dynamic Partitioning based on Darwin's Theory," *International Mathematical Forum, Hikari, Ltd.*, vol. **7**, no. 32, pp. 1587-1601, 2012.
10. J. Hereford and M. Siebold, "Multi-robot search using a physically-embedded Particle Swarm Optimization," *International Journal of Computational Intelligence Research*, vol. **4**, no. 2, p. 197–209, 2008.
11. M. S. Couceiro, F. M. L. Martins, F. Clemente, R. P. Rocha, and N. M. F. Ferreira, "Towards a Further Understanding of the Robotic Darwinian PSO," in *Computational Intelligence and Decision Making - Trends and Applications, From Intelligent Systems, Control and Automation: Science and Engineering Bookseries*. Springer Verlag, 2012, pp. 17-26.
12. M. S. Couceiro, R. P. Rocha, N. M. F. Ferreira, and J. A. T. Machado, "Introducing the Fractional Order Darwinian PSO," *Signal, Image and Video Processing, Fractional Signals and Systems Special Issue, Springer*, 2012.
13. E. J. S. Pires, J. A. T. Machado, P. B. M. Cunha, and L. Mendes, "Particle swarm optimization with fractional-order velocity," *Journal on Nonlinear Dynamics*, vol. **61**, p. 295–301, 2010.
14. J. Sabatier, O. P. Agrawal, and J. A. T. Machado, *Advances in Fractional Calculus - Theoretical Developments and Applications in Physics and Engineering*. Berlin: Springer, 2007.
15. J. A. T. Machado, et al., "Some Applications of Fractional Calculus in Engineering," *Hindawi Publishing Corporation Mathematical Problems in Engineering*, vol. **1**, no. 34, 2010.
16. P. W. Ostalczyk, "A note on the Grünwald–Letnikov fractional-order backward-difference," *Physica Scripta*, vol. **136**, p. 014036, 2009.
17. M. S. Couceiro, C. M. Figueiredo, J. M. A. Luz, N. M. F. Ferreira, and R. P. Rocha, "A Low-Cost Educational Platform for Swarm Robotics," *International Journal of Robots, Education and Art*, 2011.
18. K. Yasuda, N. Iwasaki, G. Ueno, and E. Aiyoshi, "Particle Swarm Optimization: A Numerical Stability Analysis and Parameter Adjustment Based on Swarm Activity," *IEEJ Transactions on Electrical and Electronic Engineering, Wiley InterScience*, vol. **3**, pp. 642-659, 2008.
19. Y. Wakasa, K. Tanaka, and Y. Nishimura, "Control-Theoretic Analysis of Exploitation and Exploration of the PSO Algorithm," in *IEEE International Symposium on Computer-Aided Control System Design, IEEE Multi-Conference on Systems and Control*, Yokohama, Japan, 2010.
20. Y. Shi and R. Eberhart, "Fuzzy adaptive particle swarm optimization," *Proc. IEEE Congr. Evol. Comput.*, vol. **1**, p. 101–106, 2001.
21. G. Beni, "From swarm intelligence to swarm robotics," in *Proceedings of the Swarm Robotics Workshop*, Heidelberg, Germany, 2004, pp. 1-9.