IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 October 2023, accepted 18 December 2023, date of publication 19 December 2023,
date of current version 27 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3345224

RESEARCH ARTICLE

On the Performance of Online Adaptation of Robots Controlled by Nanowire Networks

PAOLO BALDINI[1], ANDREA ROLI[1,2], AND MICHELE BRACCINI[1]
[1]Department of Computer Science and Engineering (DISI), Alma Mater Studiorum—Università di Bologna, Campus of Cesena, 47521 Cesena, Italy
[2]European Centre for Living Technology, 30123 Venice, Italy

Corresponding author: Paolo Baldini (p.baldini@unibo.it)

ABSTRACT In order to complete useful tasks in complex and changing contexts, robots need to be able to adapt their behavior or actions. This requires the ability to find effective approaches to the situation at hand, and to maintain them until they keep being effective. The final goal can be summarized by the will of maintaining a high performance during the whole life of the robot, regardless of the changes that may intervene during its activity. In this work, we evaluate by the point of view of the resulting life-long performance two methodologies for the adaptation of robots controlled by immutable network-based control systems: the Nanowire Networks. We demonstrate that modifying the best found solution leads to constant improvement and to overall better cumulative performance. Complementarily, we show that a less constrained approach simplifies the exploration of different behaviors but reduces life-long performance. Finally, we confirm previous results suggesting the potential of using this novel neuromorphic device (i.e., the Nanowire Network) for the control of robots.

INDEX TERMS Life-long adaptation, online adaptation, adaptive robotics, nanowire network.

I. INTRODUCTION

Since the start of handicraft humans tried to produce systems with a well-defined functioning, meaning that their response to the external stimuli should be predictable. This has remained true also to the present day. Engineers design systems with the aim to produce an expected and desired behavior. However, this becomes harder as the environment in which these systems have to act become complex. Indeed, an action can have different outcomes according to the context in which it is executed. To prevent this undesired behavior systems are often designed for constrained environments. Alternatively, their design takes in consideration the set of possible alternative situations. This is the approach classically used in the design of industrial machines.

Constraining the working environment is not always possible, and listing all the possible situations may not be feasible. This problem started to emerge clearly in the field of robotics, where autonomous embodied agents act in highly dynamic and unpredictable environments. Besides classical techniques from control theory and AI aimed at maintaining robot behavior within given ranges and safety limits, automatic design of control software for robot provides a viable and promising way for attaining robots equipped with both robustness and flexibility in dynamic environments. Among these approaches we mention those employing evolutionary computation [1], [2]. This approach tackles the problem dually, allowing the robot to experience in multiple situations and thus producing more general solutions. The result is the creation of more effective and robust systems [3].

The typical approach for the automatic generation of control systems is *offline*. This means that the controller is optimized once in a training environment and according to some metrics [4]. This is then uploaded to the robots in order to perform the learned behavior in the real world. Offline approaches are very powerful and effective, nevertheless they require some cautions. The produced controller might behave differently in the real word due to what is called ''reality-gap'' [5]. Additionally, being immutable, the controller might not work properly in face of changes in robot's body,[1] and

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

[1]Sensors and actuators naturally undergo breakdowns or aging, potentially changing the resulting behavior completely.

environmental conditions. Current works try to tackle those problems by means of different strategies, for example by using noise or considering the occurrence of failures during the training [6].

A different approach to face dynamic and unexpected situations is by using *online adaptation*. This approach leaves the robot free to act in the environment, continuously adapting its behavior to try improving its performance.[2] The robot becomes then able to autonomously face unexpected situations in a continuously changing context. According to the dynamics of the environment, the adaptation can work on different levels [7]. A first implementation may simply switch between some consolidated behaviors at the variance of some external properties [8], [9], [10]. Another approach may continuously optimize some parameters in order to tune the behavior [5], [8], [11]. Alternatively, the adaptation may operate by assembling a set of pre-defined fundamental actions.

In this work we use a different level of adaptation, characterized by the complete nescience of the agent in the first stages of its life. We can define this approach as purely adaptive, in that the robot has no initial knowledge or ability and has to develop them from scratch. This reflects the learning process as it happens in living beings, where all the knowledge is acquired by experience. The sole information the robot has comes from an evaluation function that represents the goal that we want to achieve. Biologically, this can be compared to the innate perception of pain or happiness that drives the very first behavioral adaptation in living beings [12]. From a technical point of view, we use an adaptive mechanism recently proposed for the online adaptation of robots [13], [14], [15], [16], [17]. This involves a continual reconfiguration of the robotic control system in order to adapt to changes in the environment.

The goal of this work is to assess if the proposed mechanism manages to adapt the robot maintaining better life-long performances than a random approach. Indeed, although the search of an effective behavior is important, it should not overshadow the life-long or cumulative quality of the behavior. This is especially true considering the online adaptation context in which we work, where the whole life of the agent matter. A robot able to efficiently perform a task in an instant, that however stays idle, useless or even causes damage in the other moments, is much less desirable than a mediocre robot that overall succeeds in its duty. We want a system able to adapt while perpetuating its strengths into the newly tried behaviors. In other words, we want a robot able to maximize its life-long or cumulative performance.

The online adaptation is not the only aspect treated in this work, nor its only motivation. Our long term goal is the miniaturization of robots for missions in unknown environments. This poses many constraints also on the complexity of the control systems used. Specifically it limits the available energy, cost, and size, making common computing systems unsuitable for our goal. To face this problem we decided to try exploiting novel and promising computing systems. Specifically, here we consider the use of Nanowire Networks [18]. Those devices posses many desirable characteristics, mainly being cheap to produce, small and potentially integrating computing and memory. The drawback consists in their topological immutability, necessitating external interfaces and proper stimulation to induce the desired response [19]. This work stands here, aiming at exploring the best solutions for the adaptation of robots controlled by Nanowire Networks.

This article is organized as follows. Section II summarizes previous work on the topics covered. Specifically, it focuses on previous usage of online adaptation techniques and Nanowire Networks in robotics. Section III concerns the online adaptation strategies used in our experiments. We describe their design and their functional differences. Subsequently, section IV describes Nanowire Networks and their working principle in order to give the reader the technicalities required to fully understand our work. In section V we describe the settings of the experiments and we introduce which metrics we are going to analyze. The analysis of the results and their discussion is provided in section VI. Section VII is devoted to general considerations on the results obtained and on how they influence the current state of the research. Finally, the section VIII summarizes the work done and the results obtained, with an outlook at possible future developments.

II. PREVIOUS WORKS

In the introduction we described as different levels of continuous adaptation may exist. In this work we opted for a pure online adaptive mechanism. This means that no preliminary training is performed offline, and that the robot starts from a condition of total unawareness, that in most of the cases determines the incapacity to approach the task. Additionally, it also does not know any low-level behavior, as instead happens in other works on adaptation that focus on using or reorganizing a basic set of actions [9], [10]. We can rationalize this choice by illustrating the design history of the mechanism.

Its first description appears in [13], [14], and [15], where it is used to adapt the behavior of robots controlled by Boolean Networks (BNs). BNs are network-based systems where the interaction between the nodes is controlled by Boolean functions [20]. Although their offline training is possible, the work focussed on discovering if the context in which a robot learns leads to the emergence of different behaviors from the same control system, in what it's called phenotypic plasticity.

A second appearance of this mechanism is in [16] and [17], where it is used for the adaptation of robots controlled by Nanowire Networks. In that work the motivation behind the online adaptation changes, as it is justified by the immutability and heterogeneity of the control devices. Indeed, each

[2]Differently from online learning techniques, the process of adaptation potentially never stops.

NNs is unique and cannot be exactly replicated nor modified. In an offline approach, this would require to train each existing robot independently, as a common solution cannot be identified. Additionally, it may not be always possible to obtain similar behaviors,[3] complicating the creation of teams able to face cooperative tasks. Different robots should be trained to maximize group performance, greatly increasing the complexity of learning. Indeed, since many behaviors potentially exist, each robot should be trained to perform well in most of the situations. Even assuming that it is possible, this may be fragile to the addition of unexpectedly behaving robots, leading to possibly undesired emerging behaviors.

Those properties limit the effectiveness of offline training, and favor the adoption of online adaptation. This already requires the control system of each robot to be adapted independently, and additionally allows to continuously optimize the behavior for each specific condition. The scalability of the system is therefore intrinsically increased, as the addition of a new robot would simply cause the adaptation of the neighbors. We can conclude that in the context of heterogeneous and scalable robot groups the presence of continuous adaptation, led by an intrinsic driving force (i.e., an evaluation or objective function), can be considered an essential condition for the emergence of complex desired behaviors.

Compared to previous publications, the novelty of this work consists in analyzing the behavior of the robot from the point of view of the life-long performance. Approaches considering only the final or best performance are not suitable to evaluate online adapting robots, which have to maintain high-quality behaviors for most of their life. This shed a new light on previous results, demonstrating that mechanisms able to produce similar maximal results does not always produce the same life-long performance. To the best of the authors' knowledge, this is the first time that this type of analysis is used to evaluate online adapting robots. Finally, due to the novelty of NNs in robotics, our previous work only evaluated their use in simple tasks, such as collision avoidance [17]. In this work we test the system on more complex experiments requiring not only reactive or memory based behaviors, but also a combination of the two.

III. ONLINE ADAPTATION

In this work we describe the adaptive mechanism, and we compare it with a baseline algorithm performing a random sampling of possible solutions. We call the first *local adaptation (LA)*, and the second *random adaptation (RA)*.

Both methods are designed to operate on a network-based, immutable control system. The process consists in its coupling with the sensors and actuators of the robot (see Figure 1). The main idea is that we can inject the signals from the sensors to specific points of the network in order to produce a perturbation suitable to control some actuators. Conversely, taking the signals from specific points of the network to control the actuators can be seen as a change in the interpretation of network state. The mechanisms do not modify the NN, but only its sensory and possibly motor couplings. The couplings between the robot and the NN represent a *solution* or a *configuration*.

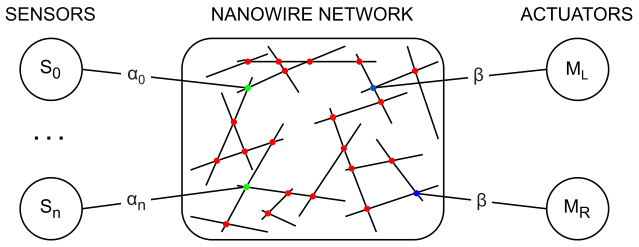

FIGURE 1. Schematic representation of the control architecture. $[S_0, \ldots, S_n]$ represent robot sensors, while M_L, M_R represent left and right motors. $[\alpha_0, \ldots, \alpha_n]$ are the weighting factors, different for each sensor signal. β represents the influence of the motor resistance in the electrical equivalent system.

A. LOCAL ADAPTATION

Local Adaptation (LA) is based on the controlled perturbation of the best solution identified. The idea relies on the assumption that minor modifications do not completely change the behavior and instead allow the exploration of similar solutions. If this exploration discovers a better configuration, this becomes the new origin from which to start the next step of the adaptation (see Algorithm 1).

Algorithm 1 Pseudocode of the Local Adaptation (LA). The lines in red are the main difference between LA and RA

```
let N be the set of nanowires of the NN
best_C ← new couplings
best_p ← evaluate_performance(best_C)
for 1 to number of steps do
  if best_p < threshold then
    C ← new couplings
  else
    C ← perturb_couplings(best_C, N)
    C ← perturb_weights(best_C)
  end if
  p ← evaluate_performance(C)
  if p ≥ best_p then
    best_C ← C
    best_p ← p
  end if
end for
```

In practice, the strategy consists in modifying the coupling between the sensors[4] and the robot controller (i.e., the NN).

[3]Different NNs may not contain enough complexity to achieve every possible behavior.

[4]Modifying the coupling between the network and the actuators is also possible, but it is not considered in this work.

When the adaptation occurs, a random subset of the couplings is chosen and reconfigured towards different points of the network (see Algorithm 2). The possible reconfiguration points must be free (i.e., not already coupled to a sensor) and more than two junctions distant from the outputs (i.e., the points of the couplings with the actuators). The size of the modified set determines the difference between the two configurations, how much the algorithm is able to explore, and how much it tries to refine the current configuration.

Algorithm 2 Pseudocode of *perturb_couplings*. To simplify the logic, we consider C as the set of nanowires the inputs/outputs connect to. This means that in this pseudocode changing c only means changing the nanowire a coupling connects to

Input: C : *couplings*, N : *nanowires*
Output: C
$O \leftarrow \{c \in C \mid is\_output(c)\}$
$I \leftarrow \{c \in C \mid is\_input(c)\}$
$N_1 \leftarrow \{n \in N \mid n \notin C,\ and\ distance(n, O) > 2\}$
$pick\_randomly\ I_1 \mid I_1 \subset N_1\ and\ 1 \leq |I_1| \leq 4$
$pick\_randomly\ I_2 \mid I_2 \subset I\ and\ |I_1| = |I_2|$
$C \leftarrow C \setminus I_2$
$C \leftarrow C \bigcup I_1$

The adaptation also influences the weights of the couplings between the sensors and the controller (see Algorithm 3). This means that the strength of the signal transported may be amplified or attenuated according to the weight of the coupling. This allows to balance the influence of specific inputs in the computation. Some use cases include the attribution of a greater/lower importance to some sensors, or the homogenization of inputs with different range of values.

Algorithm 3 Pseudocode of *perturb_weights*

Input: W : *weigths*, σ
Output: W
$W \leftarrow \{w + gaussian(0, \sigma) : w \in W\}$
$W \leftarrow \{max(0, w) : w \in W\}$

One of the risks in using LA is that the initial solution from which the adaptation starts might be excessively far from a good one, i.e., it may require too many modifications to become satisfying. In this context, the method may fail in trying to escape some very bad local optima. In the attempt to mitigate this problem, LA uses a threshold to postpone the local adaptation until a minimum acceptable solution is found. We can imagine this strategy as the application of a bootstrap phase. This works by checking if the performance of the robot is greater than a given value. Basically, it allows exploring randomly by modifying the totality of the couplings, starting then a finer optimization as soon as a good enough solution is found. The choice of the threshold value is important, in that it strongly impacts the direction of the adaptation. A too high value would prevent the local optimization, starting a random sampling of solutions.[5] A too low value would make the initial exploration limited, maintaining the risk of optimizing ill-couplings as in the LA implementation without threshold. Finally, since this parameter is related to the performance of the robot, this has to be tailored on the specific task. In this work, we selected the threshold value through the qualitative analysis of limited set of experiments. Nevertheless, it is also possible to calibrate this parameter using some optimization techniques.

B. RANDOM ADAPTATION

In order to compare the quality of the guided adaptation performed by LA, we decided to set up a baseline algorithm generating a random configuration at each adaptation step (see Algorithm 4). We call it Random Adaptation (RA). This approach is able to potentially produce any possible configuration, therefore allowing in principle to find the global best solution in any situation. This makes RA potentially more powerful than LA, that may instead get trapped in some local maxima. Nevertheless, this approach may prove to be inefficient as it lacks a strategy to drive the adaptation. The result is that RA may waste time trying many poor configurations.

Algorithm 4 Pseudocode of the Random Adaptation (RA). The line in red is the main difference between LA and RA

let N be the set of nanowires of the NN
$best\_C \leftarrow new\ couplings$
$best\_p \leftarrow evaluate\_performance(best\_C)$
for 1 **to** *number of steps* **do**
  $C \leftarrow new\ couplings$
  $p \leftarrow evaluate\_performance(C)$
  if $p \geq best\_p$ **then**
    $best\_C \leftarrow C$
    $best\_p \leftarrow p$
  end if
end for

Technically, the adaptation process is similar to LA, with the couplings between the sensors and the NN being continuously modified. The algorithm also applies a weighting, amplifying or attenuating the signals. The difference with LA consists in the size of the modified set of couplings, that in RA contains all of them. This means that using RA all the couplings will be reconfigured towards random points of the network, still maintaining the constraints of unique sensor input per node and distance from the actuators. Additionally, also the weight attributed to each coupling will be sampled completely at random.

IV. NANOWIRE NETWORKS

The long-term goal of this work is the creation of cheap, miniaturized, autonomous robots. This limits the use of

[5]This is the approach used by the Random Adaptation (RA), as described later.

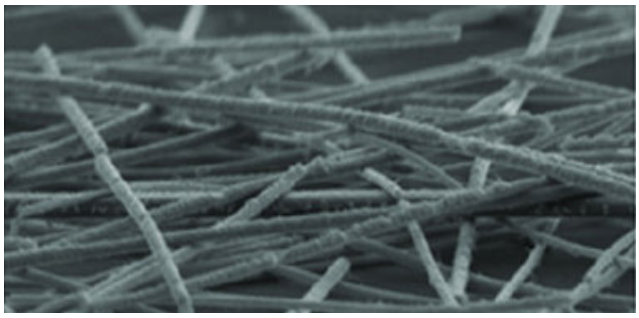

FIGURE 2. Microscopy image of a Nanowire network. Picture taken from [18] by courtesy of the authors.

computationally powerful controllers and supports the adoption of novel computing devices. Specifically, in this work we consider a recently proposed system that exhibits complex dynamics: the Nanowire Network (see Figure 2). Those are electrical devices composed of nanometric silver wires [18]. The production process is relatively simple, only consisting in the drop-cast of wires into an insulating substrate [21].

The NN resulting from the production process can be statically represented as a nanoscale circuit of resistors. The junctions between wires represent the resistances, with their conductance decreasing as the distance between the two wires increases. Nevertheless, this static representation is limited. Indeed, NNs present a complex dynamic in time, with the resistivity of each junction changing according to its recent stimulation. When two adjacent wires are subject to a voltage difference, the silver particles that compose them move towards the junction. This process creates a bridge that shortens the distance and therefore reduces the resistivity of the junction. The accumulation of the particles happens in time, causing a smooth variation of the conductivity. When a sufficient amount of particles is accumulated, the decrease in resistivity stops in a stable state. Vice versa, if the voltage difference across the wires is removed, the bridge starts to dissolve. This leads the system back to its original resistive state. In this work we consider the voltage stimulation as the only way to modify the internal state of the NN.

The interesting dynamics of the NNs has recently attracted attention, especially from the area of computer science. The main points of interest reside in the possibility to store information directly into the computing device,[6] and to change its functioning accordingly [22]. This possibility is thought to allow overcoming the costs related to information transfer, as it is needed in systems based on the Von Neumann architecture [23]. Additionally, the device presents some similarities with the neural system [24], [25]. The dynamic of the junctions has been indeed compared with the strengthening of the synaptic connection between two stimulated neurons, in what is commonly known as Hebbian theory [26], [27]. This suggests that the use of NNs may be effective in AI systems. Recent works used NNs as reservoirs in Reservoir Computing in order to solve some classical AI tasks, such as digits recognition in MNIST and Mackey-Glass time-series prediction [19]. Another aspect that supports the current interest on NNs is the expected reduced power consumption in performing complex operations, a property common to many analog computing devices [23].

[6]The NN can be used as a short term memory by exploiting its memristive properties.

V. EXPERIMENTAL SETTINGS

The aim of this work is to assess the overall quality of the behavior during the life of an adaptive robot. To achieve this result, we evaluate the robotic system in two experiments. The first is the *"T-maze"*, that assesses the ability to exploit the network plasticity. The second is the *"foraging"*, that checks the system in a well-known minimally cognitive task in the field of robotics [28], [29].

The setup consists in the generation of 150 unique NNs *replicas*. These networks are the core of the robotic control system and are not externally modifiable. The only way to influence them is through the adaptation of their couplings with the robot sensors, as described in section III. This is done through the use of the two adaptive mechanisms: LA and RA. The adaptation to each of the two tasks is guided by a specific evaluation function, that acts as a driving force to adapt the behavior [12]. This calculates the instantaneous performance according to the local perception of the robot at discrete time-steps (i.e., after each action). Before the adaptation, the overall performance obtained by the coupling is computed as a sum for the foraging, and as an average for the T-maze. Each set of couplings between the robot and the NN is called *configuration* or *solution*, and is tested during an *epoch*. At the end of each epoch the solution is adapted according to the obtained performance. The amount and duration of the epochs is task-specific and depends on its complexity:

Task	Epochs count	Epoch duration [s]
T-maze	300	50
foraging	500	150

The performance of the robot in the task drives the adaptation process. In order to be effectively autonomous, the evaluation policy has to be defined as an intrinsic value of the robot itself. In other words, the robot must autonomously understand how well it performed. This policy must take in consideration only the local perception and activity, namely the state of the sensors and actuators. This means that no external reward mechanism should be used.

To assess the quality of LA we compare its performance with that of RA. The evaluation focuses on the overall performances during life, and also considers the best results found during the whole life. The goal is to assess how effective the two proposed techniques are in exploring the space of possible solutions while maintaining an acceptable behavior.

We conduct all the experiments using the Webots simulator and the Python programming language. The model of the robot employed is the E-puck [30]. All the code and results are available online [31], [32], [33].

A. T-MAZE

We designed the T-maze task to assess whether the intrinsic dynamics of the NN can be effectively exploited by the adaptation. In the T-maze task, the robot has to travel a corridor till its end and then turn either left or right (see Figure 3). The information on the correct direction is provided by coloring the initial part of the corridor. The intuition is that to perform the correct turning the system has to take advantage of the junction plasticity to create some sort of memory.

More precisely, in the T-maze task we require the robot to reach the correct end-point of a T-shaped maze. This is divided in different areas:

- the start;
- the transition;
- the ending points.

Each of them is characterized by a well-defined color, allowing the robot to assess the quality of its behavior. The start of the maze periodically alternates between two different shades: white and black. The ending points are static and of different colors, again black and white. Finally, the transition area is gray, and it is intended for interrupting the signal associated to the correct end-point.

The robot perceives its surroundings through the use of a ground sensor, a negated ground sensor, and 8 proximity sensors. The formers enable perceiving the color of the ground; the latter to identify obstacles such as walls. At the beginning of an evaluation epoch, the robot is placed at the start point. The color of the ground determines the correct destination to reach. Specifically, the robot has to reach the area of the opposite shade (i.e., white for a black start, and vice versa). At half of the epoch duration the robot position is reset to the start, and the color of the ground changes to the opposite shade (i.e., *white* $\rightarrow$ *black* or *black* $\rightarrow$ *white*), effectively changing the destination. This design allows to test the ability of the robot to reach both the destinations. Additionally, it permits to understand if the correct destination has been reached just by looking at the ground color through the sensors. The evaluation policy can therefore be defined just according to the perception, effectively allowing to embed it into the robot itself and avoiding the presence of an omniscient rewarding mechanism. It is defined as follows:

$$score_i = 2 \cdot (1 - \alpha_i) \cdot (1 - \beta_i) - \alpha_i \tag{1}$$

where:

- $\alpha \in \{0, 1\}$ is 1 if the ground color is the same as the starting one, 0 otherwise;
- $\beta \in \{0, 1\}$ is 1 if the ground is gray, 0 otherwise;
- i is the index of the evaluation step.

The given policy aims to reward behaviors which rapidly escape the starting area and reach the correct destination. Specifically, hovering an area of the same color of the start causes the performance to decrease, while staying on an opposite shade causes the performance to increase. Traversing the transition area does not directly affect the score, however this introduces a bias favoring the movement of the robot towards the end of the corridor.

Finally, for the T-maze LA makes use of the adaptation threshold as described in the Online Adaptation section. This allows to explore randomly for few epochs until a minimum acceptable solution is found. Preliminary investigations suggest indeed that its use speedups the discovery of a successful behavior [16], [17]. For this task, we chose the threshold value according to a qualitative analysis of some preliminary tests. The goal is avoiding a relentless random exploration, and instead maintaining it relatively brief. We decided to use a threshold value of 15, which induced the longest bootstrap phase in the experiments to last 30 epochs. Considering 300 epochs of adaptation, this means that the longest bootstrap phase took 10% of the whole adaptation. This is important to correctly compare the two adaptive mechanisms and to avoid LA to behave like RA.

B. FORAGING

The foraging is a classical robotic task that requires the agent to find and bring preys to the nest. The arena is divided in two areas:

- the nest;
- the hunt-field.

The nest is where the captured preys shall be brought, while the hunt-field is where the preys are initially located.

In our implementation, we use a circular arena with an external hunt-field and an internal nest (see Figure 3). Each area is characterized by a different color that the robot can distinguish through its ground sensors. A light source is positioned at the center of the arena, providing a landmark that the robot can perceive through its light sensors.

Thin white plates homogeneously distributed in the arena represent the preys that the robot has to collect. They are passive, meaning that they do not have any active behavior and can be moved only by the robot. The latter is equipped with eight light sensors and two ground sensors, one of which is negated. Two wheeled motors and a gripper are the actuators that allow the robot to influence the environment. The firsts are controlled by continuous values, while the state of the latter changes if the control signal from the NN exceeds a given threshold.[7]

The evaluation policy should drive the adaptation in order to obtain behaviors with a capture in the hunt-field and the subsequent release in the nest. To achieve this result, we can

[7] We choose a threshold of 0.5 for an output in range [0, 1].

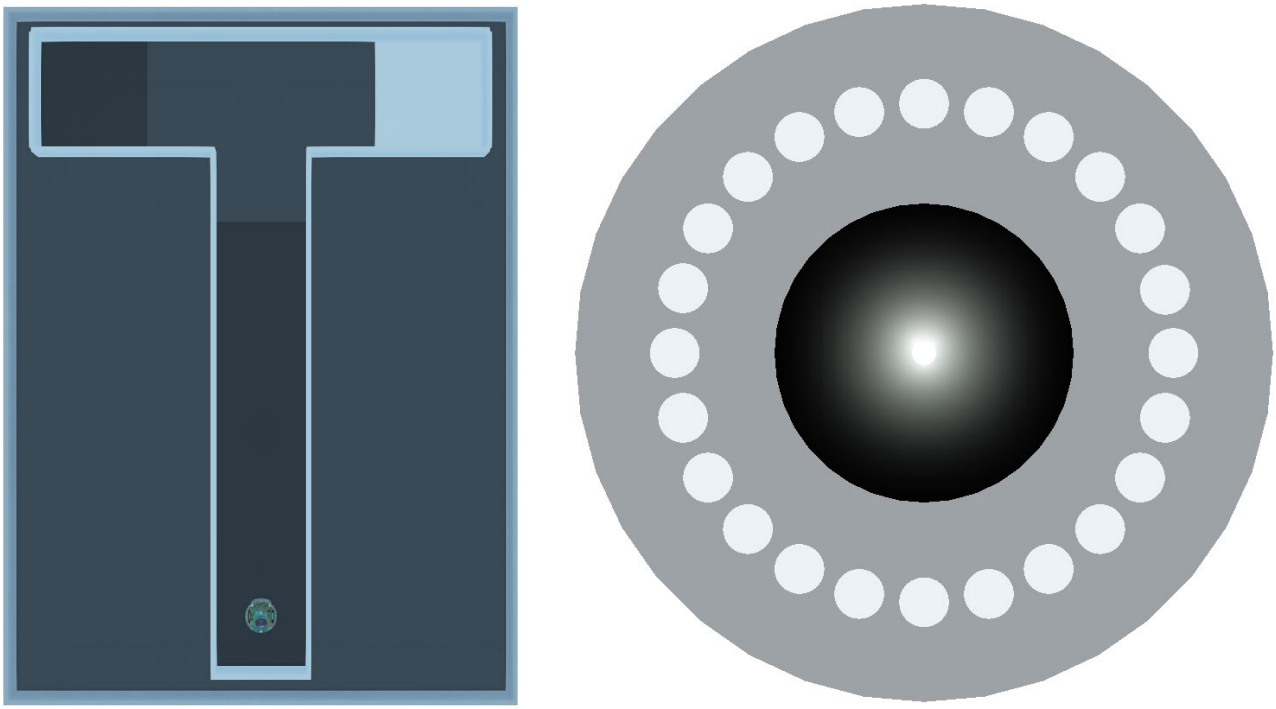

FIGURE 3. The arena of the t-maze (left) and foraging (right) tasks.

define the evaluation function as following:

$$score_i = \begin{cases} 50 & if \;\; A_i \wedge \neg\Theta_i \wedge f(P_i) \\ 100 & if \;\; A_i \wedge \Theta_i \wedge f(N_i) \wedge g(P_i) \\ -50 & if \;\; A_i \wedge \Theta_i \wedge f(H_i) \wedge g(P_i) \\ 0 & otherwise \end{cases} \quad (2)$$

where:

- $A \in \{False, True\}$ is *True* if the gripper state changed, *False* otherwise;
- $\Theta \in \{False, True\}$ is *True* if the gripper is currently open, *False* otherwise;
- $f(x)$ is a function that returns *True* if the color of the ground before the gripper action was equal to the parameter x, *False* otherwise;
- $g(x)$ is a function that returns *True* if the color of the ground after the gripper action is equal to the parameter x, *False* otherwise;
- P, H, N are respectively the colors perceived when hovering a *prey*, *hunt-field*, and *nest*;
- i is the index of the evaluation step.

The given policy defines a capture as the closure of the gripper while the robot is on a white ground (i.e., the prey). Differently, it identifies a release by looking at the color of the floor before and after the gripper opens. If, after the release, the color is different from white, then there has not been any release. It is important to underline that the robot does not know if the gripper is open, and neither if it is carrying a prey. Indeed, there is no feedback loop going from the actuators to the control system, and assumptions can be made only by looking at how its actions influence the environment. The alternative is for the robot to autonomously develop an internal memory, exploiting the plasticity of the network to save information about its state.

Another aspect that emerges from the evaluation function is that a release in the hunt-field is penalized by the same amount used to reward a capture. This prevents both increments and decrements of the evaluation function value in case of a *capture-release* sequence outside the nest. Direct and interleaved deliveries to the nest will therefore have the same final performance value. The only difference may consist in the total delivery time, possibly allowing additional captures to the fastest behavior. This choice prevents the discarding of acceptable solutions, although allowing the perpetuation of possibly inefficient ones. On the long run, the expected result is the optimization of the length of inefficient sequences in order to increase the delivery speed.

It is important to highlight that a capture in the nest is not possible. Indeed, the white plate (i.e., the prey) almost immediately vanishes after the release. The permanence time before disappearing is just sufficient for evaluation function calculation, and does not allow a second capture.

Finally, for the foraging task LA does not make use of the adaptation threshold. Indeed, preliminary analyses did not show any advantage in using this bootstrap phase in this task.

VI. RESULTS

In this work we are interested in understanding how the two adaptive mechanisms impact the overall performance of the robot during its life, and how well they explore the space of possible solutions.

A preliminary analysis consists in assessing which percentage of the replicas succeeded in achieving a minimum acceptable result during their life, and at which step they did it. We can measure this by a Run Length Distribution (RLD) that provides the percentages of successful replicas along the epochs. For the T-maze task we consider the number of correctly reached destinations, and for the foraging task the amount of correctly delivered preys. We expect the performances of the two mechanisms to be initially similar, and to deviate in favor of LA as soon as it finds some good enough solutions to optimize.[8] The results of the foraging task seem to confirm this expectation, with the difference between LA and RA that increases with the simulation time (see Figure 4). Nevertheless, the results of the T-maze are not strong enough to confirm this thesis. Indeed, in this task the difference in performance of the two mechanisms remains relatively small.

The analysis of the RLD also shows that the distance between the performance obtained by LA and RA does not diverge only according to the epoch, but also according to the chosen threshold of acceptability. Taken the foraging task and increasing the threshold by one delivery at a time, we can see that the gap between the performance of the two mechanisms also increases (see Figure 4). This suggests that the improvement of LA over RA directly correlates to the number of deliveries required, and therefore to the complexity of the task. In the T-maze this correlation is not visible. Indeed, 100% of the replicas succeed in at least one reach, and the percentage difference for the two reaches is too small to draw any conclusion. Overall, we can expect LA to perform better when the task is complex (i.e., when the set of acceptable solutions reduces), provided that it is granted with enough time for the local search to start being effective.

The analysis of the RLD allows to understand the effectiveness of the mechanisms according to the complexity of the task and the available time (i.e., the number of epochs considered). Nevertheless, it considers only if a replica reaches a given threshold before a specific epoch. This completely ignores if the performance was just by luck or if the robot effectively learned and improved a successful behavior. In order to address this question, we analyze the average performance obtained by all the replicas with the two mechanisms at each epoch. To have a clearer view of the trend, we average the performance in a 25 epochs window.[9] The results show a higher life-wise performance for LA in both the tasks (see Figure 5).

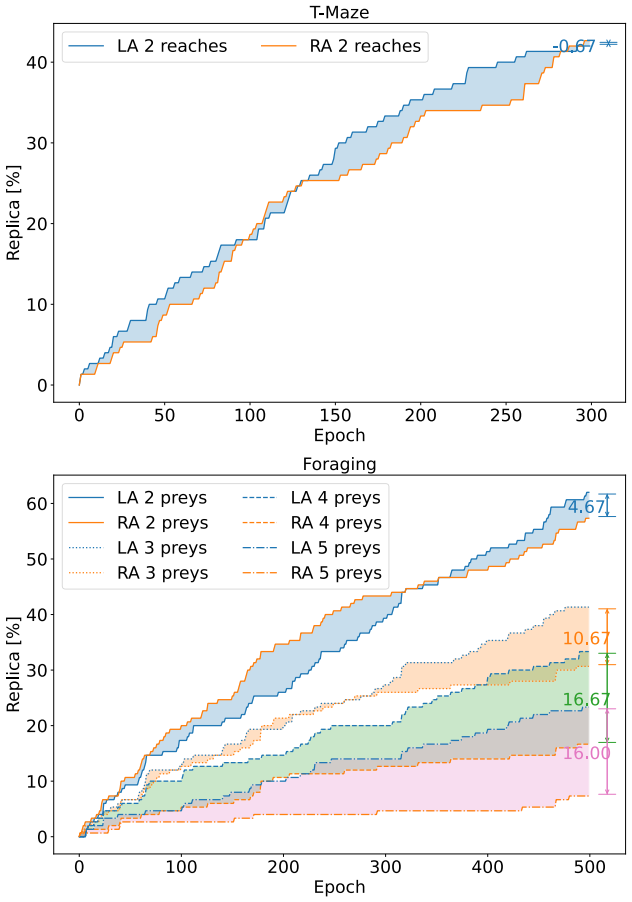

FIGURE 4. Run length distribution indicating the percentage of replicas that succeeded in reaching both the target destinations (T-maze), and in delivering the specified amount of preys (foraging). On the right side of each plot the difference between the final result of the two mechanisms is highlighted. On the top we plot the T-maze results, on the bottom the foraging ones. Both the results exclude the base-case of just 1 target reach and 1 prey delivered in that all the replicas managed to achieve these results with both the mechanisms in the very first stages of the experiment.

Additionally, while the performance of RA remains low and stable during all the simulation, in the foraging task the performance of LA shows an increasing trend. This means that the performance difference of the two mechanisms is not constant, but increases with time in favor of LA. This trend is of primary importance, as the online adaptation aims to produce systems that maintain and improve the discovered behaviors.

The analysis of the average performance in time allows to understand the overall quality of the behaviors found by the two mechanisms. Nevertheless, this measure may be sensible to replicas outliers. Indeed, a subset of robots might have learned an extremely well performing behavior, increasing the average of the whole group. Alternatively, few very poorly performing replicas might affect the results in the opposite direction. To verify this possibility we have to move to a higher level of detail, verifying the performance distribution of the replicas. We consider the cumulative amount of reaches for the T-maze and of deliveries for the foraging. In both

[8] In the very first epochs the behavior of LA and RA is comparable due to the presence of the threshold.

[9] The trend remains the same even with narrower and wider windows.

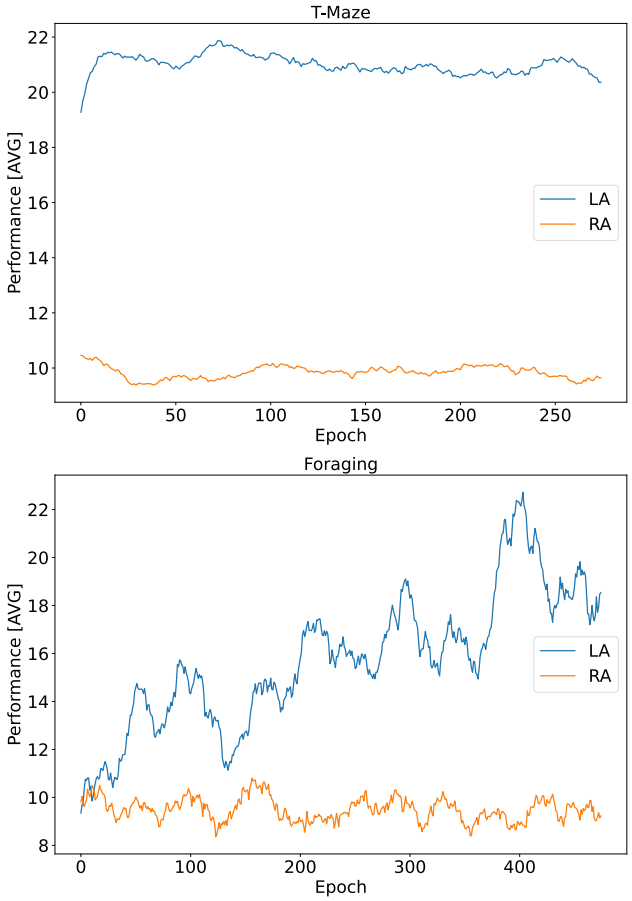

FIGURE 5. Average performance over all the replicas at each epoch. The result is again averaged over a window of 25 epochs in order to more clearly show the pattern. On the top we plot the T-maze results, on the bottom the foraging ones.

the tasks the replicas adapted with LA performed better, producing significantly different statistical distributions[10] (see Figure 6). In the T-maze this difference is particularly high, with the worst score obtained by LA being higher than the median of RA. The results of the foraging tasks are less impressive, but we can still observe a substantial amount of replicas with high performance. Specifically, more than 25% of the replicas earned higher scores than the best obtained by using RA. These results indicate that the better overall performance identified earlier is not due to outliers, but to a factual difference in the quality of the generated behaviors.

Another interesting aspect to analyze is the relation between the maximum and the cumulative score for each replica. This measure helps to understand the weight of the best epoch in the life-wise performance. Additionally, it gives supplementary information to compare the two adaptive mechanisms. In the T-maze it is not possible to identify any particular relationship between these two measures (see Figure 7). Indeed, some replicas obtaining relatively low epoch-wise scores manages to obtain better cumulative performances than others performing better on a single epoch. The foraging results seem to suggest instead a tendency for direct correlation, with replicas obtaining high epoch-wise scores also displaying better life-wise performances. The T-maze results may depend on the complexity of the task, requiring a finer tuning and therefore complicating the replication of the successful behaviors. This means that the perturbation of a good solution is less frequently of the same or higher quality and more luck is needed to improve the performance. With an analogy to search landscapes [34], it is likely that the landscape in our definition of the T-maze task is rather rugged, i.e. characterized by the fact that configurations that are close do not produce similar evaluations. Obviously, since RA does not perturb the best solution this argument is valid only for LA. Additionally, the objective function of this task heavily depends upon the reach of the target destinations, creating two disjoint clusters of results. This means that when the adaptation finds an effective configuration it experiences a substantial increment in performance. On the other hand, it implies that we cannot rely on small improvements to drive the adaptation to the desired behavior,[11] and that instead more luck is needed. Producing good foraging behaviors may require less refinement, increasing the cumulative performance and allowing an easier propagation of good behaviors when using LA.[12] Also, in this task the objective function causes a more continuous performance trend allowing the adaptation to gradually improve the behavior. Finally, it is possible that in this task the best score has a higher effect on the cumulative performance. This impact is in any case limited, as in the opposite case we would expect the cumulative performances of RA and LA to be much more similar when compared on replicas with similar best score. Instead, it is clearly visible that the cumulative performances of LA are much higher than the ones produced by RA, creating two mostly disjoint clusters. This implies that a single effective solution is not enough to strongly influence the life-wise performance. Overall, we can conclude that finding a successful solution simplifies the improvement of the overall performance, at least when using LA. This mostly depends on the level of tuning required by the task and on the chosen objective function. We can also conclude that the presence of a high life-wise performance does not imply that the adaptation produced a good performing solution. Indeed, the ability to effectively propagate mediocre behaviors may still lead to obtain a high cumulative performance.

VII. DISCUSSION

The goal of this work is to assess how two adaptive methodologies perform during the whole life of a robot. Our

[10] We performed the paired Wilcoxon test obtaining p-values of 3×10^{-26} for the T-maze and 10^{-18} for the foraging.

[11] The adaptation cannot follow a gradient of performance to improve.

[12] Differently from RA, LA operates by perturbing the best solution found and can therefore replicate successful behaviors.

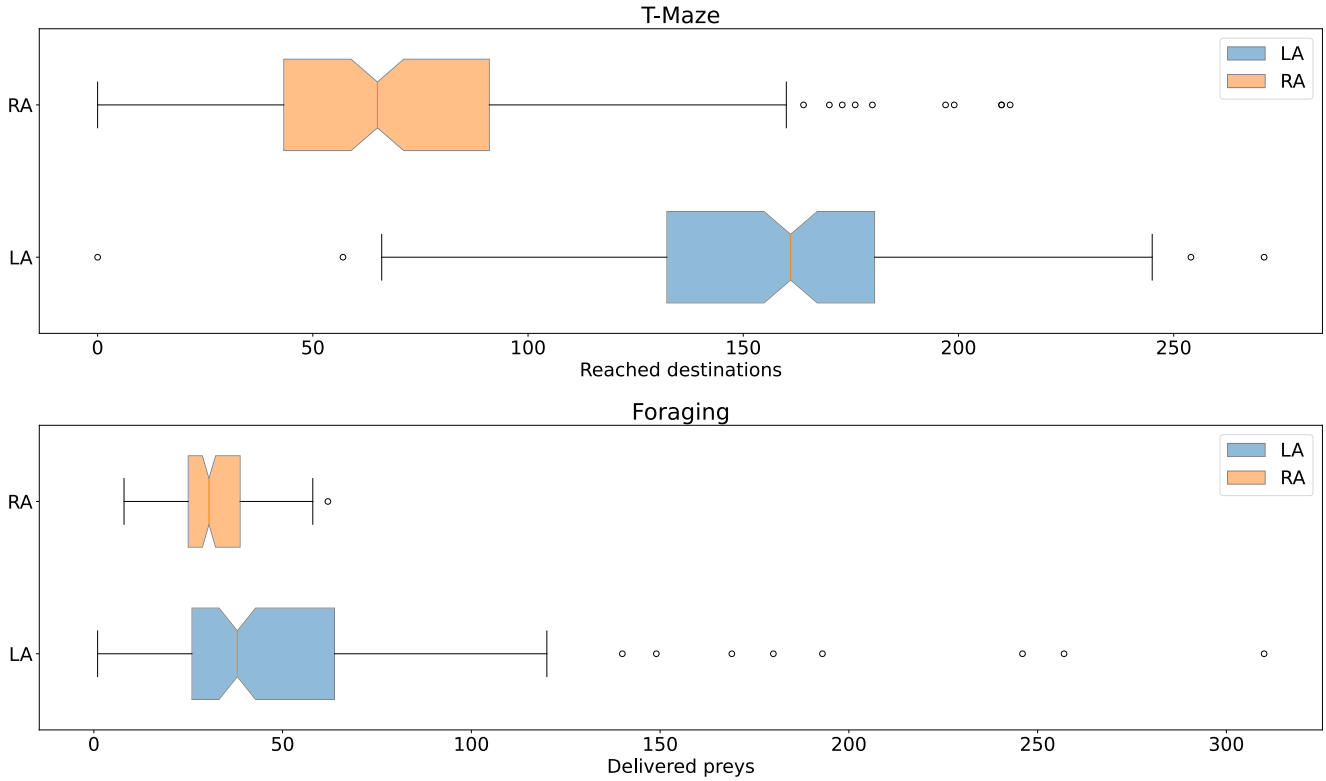

FIGURE 6. Distribution of the life-long performance of each replica in the T-maze and foraging tasks by use of the two adaptive mechanisms. The orange boxes show the distribution of the T-maze scores, the blue boxes the distribution of the foraging scores. On the ordinate axis (i.e., the y-axis) is indicated which adaptive methodology generated the data. The top axis shows the values of the T-maze distribution, the bottom one the scores of the foraging distribution.

results show that RA sometimes allows finding solutions that outperform the ones of LA. We believe this to be due to LA stalling in a local maxima, preventing the adaptation to better results. However, when not blocked in the surrounding of a poor solution the perturbation of a successful configuration often leads to better performance, both epoch- and life-wise. This is due to the search in the neighborhood of the best configuration found, raising the probability to find another acceptable solution and increasing the life-wise performance. Contrarily, the almost unconstrained search granted by random adaptation often leads to the test of poor behaviors, lowering the cumulative score of the robot.

Overall, RA explores the solution space without any constraints, yet not being able to produce outstanding configurations except for lucky trials. LA searches a way to continuously improve its best solution, often landing to quite good configurations and leading to a higher life-wise performance. This characteristic of the performance evolution over time is of primary importance, as the adaptation of real robots is strongly interested in systems that keep obtaining good results while attempting to improve their performance.

Most current work about online adaptation and learning start with some high level assumptions on the structure of the controller. The robot often knows a set of fundamental actions that enable it to obtain the desired behavior. Alternatively, the robot may already know the logic of the algorithm to run, limiting the adaptation to the runtime optimization of some parameters. In our research we work at a lower level of abstraction, where the robot starts the adaptation completely unaware of any action or logic. The only knowledge permitted is that of a function intrinsic to the robot allowing to understand how well it is behaving. This approach is more general and potentially allows the emergence of novel behaviors. Indeed, the robot is not constrained by some fundamental actions, and can create them according to its needs. We were amazed to observe the emergence of some unexpected behaviors during the run of some preliminary experiments. In the T-maze task, the robot started to exploit the elastic collisions supported by the simulator to obtain better scores. The adaptation discovered that a rapid impact against a perimetral wall produces a bounce. This collision lifts the robot, allowing the ground sensor to perceive a signal equivalent to the white destination. This behavior does not effectively impact the emergence of the desired result, as the performance increases by spending more time on the white area. However, it initially drove the adaptation of some robots unable to reach the white destination. Also in the foraging task we observed

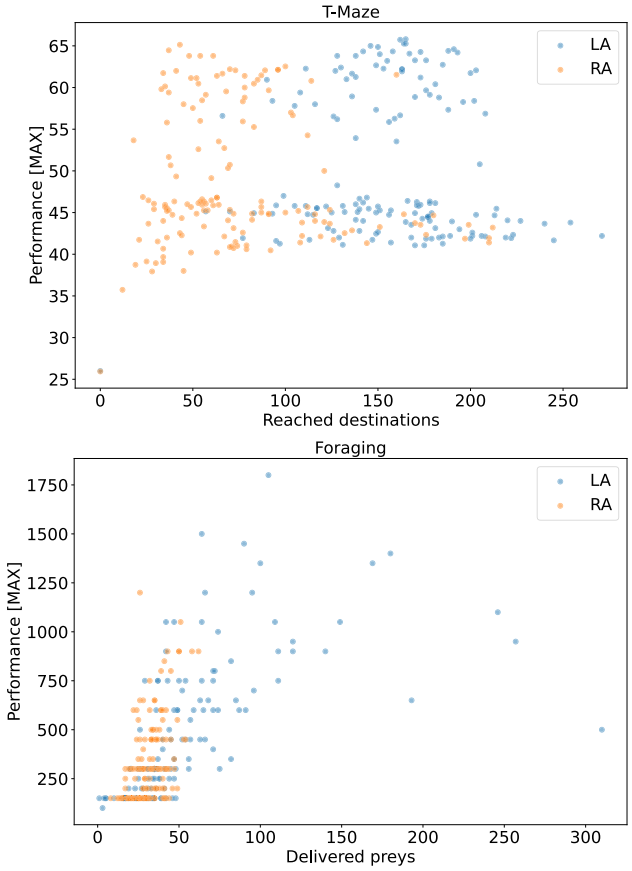

FIGURE 7. Correlation between maximum and cumulative performance obtained by each replica adapted by LA and RA in the two tasks. The abscissa axes shows the cumulative performance during the life, the ordinate axes the maximum score obtained. The results of the T-maze task cluster into two main groups, discriminating the instances that managed to reach both the target destinations.

some unexpected behaviors. The robot learned to exploit the minimal time interval between the release on the nest and the vanishing of the prey to perform an ulterior capture and release action. This initially leads to performances otherwise impossible to obtain. Obviously, although interesting, all these exploits have been disallowed before the start of the final experiments, in order not to affect the quality of the analysis.

Both the mechanisms used in this work present some strengths. RA allows a more homogeneous exploration, and prevent stagnation on local maxima. LA makes it possible to obtain better results both during life and in the single epochs. In the online adaptation of robots we aim at all those features. To attain this goal, we plan to work on the definition of a new adaptive methodology able to maximize the life-long performance while maintaining the ability to explore. We believe that this mechanism should be able to perturb the best solution dynamically, according to its score. Alternatively, we may allow perturbing also worse configurations. This would enlarge the set of solutions that may be reached by the adaptation, possibly enhancing diversification.

The ability to adapt to changes and to create novelty makes our approach potentially powerful. Nevertheless, there are some aspects that can be improved. The initial ignorance and therefore the need to generate the actions from scratch complicates the creation of good behaviors. Additionally, the reconfiguration of a subset of couplings towards random points of the network may cause a sudden drop in performance. This is the case when we reconfigure couplings that strongly influence the resulting behavior. To improve these aspects we plan to devise some heuristics allowing a more controlled adaptation. The idea is that some information like the distance of the reconfiguration point can be used to modulate the adaptation, possibly being more aggressive in the first stages and less when approaching a good solution. This may even help to solve the previously stated problem of getting stuck in the neighborhood of a local maxima. The expected result would be the increment in the adaptation speed and in the life-long performance.

Due to the limited availability of Nanowire Networks devices, all the experiments of this work were performed in simulation. This allows to test a wide amount of different topologies, and therefore to avoid biases due to ill-instances, i.e., devices very (un)suitable to perform the desired task. Howbeit, there are plans to create a hardware implementation of the proposed control system. We conceive a NN coupled with a Microelectrode Array (MEA) injecting signals to different areas of the device as it is done with neurons and tissues [35], [36]. Although conceptually different from stimulating a single nanowire, in reality the behavior is expected to be similar. Indeed, neighbor nanowires naturally exhibit negligible differences in their electric potential. This means that stimulating a single nanowire or its neighborhood has a comparable effect. Alternatively, recent works and different but similar architectures (e.g., multiterminal memristive networks) show that is possible to target single nanowires [18], [37]. This opens the way to the development of a package enriched with pins connected to single nanowires. The sensory inputs should be preprocessed to avoid damaging the device, and then feed in it. In the same way, the outputs should be adjusted to effectively control the actuators. The adaptation mechanism can be initially obtained through a proper usage of multiplexing and memories, and eventually substituted by more advanced solutions such as self-assembling wires [38]. We expect the implementations of LA and RA to be very similar, just differing in the use of the best configuration saved in the memory.

Although not in the scope of this work, the obtained results suggest that the proposed mechanisms may be applied to produce heterogeneous self-organizing multiagent systems. This potentially allows the creation of more resilient and advantageous networks of devices performing tasks for which they were not initially developed. Specifically, we envision IoT devices equipped with NNs for advanced and not pre-defined AI tasks at low power consumption and costs. In future,

entire factories may adapt their production processes according to single constituents, automatically learning how to balance the work-load according to a general evaluation function.

VIII. CONCLUSION

In this work we analyze the life-long performance obtained by the adaptation of robotic behaviors with two different mechanisms. The assumption is that an online adaptation should try to maximize the performance during the whole process. We observe that a mechanism adapting the best found solution (i.e., LA) usually performs better than a mechanism exploring at random (i.e., RA). Specifically, the obtained performance is higher both comparing the best behavior and the whole life of the individual. However, we occasionally witnessed some randomly adapted solutions performing better in a specific instant. We attribute this divergence to the adaptation of the best solution stagnating in some local maxima. The random exploration of the whole space of solutions prevents this problem, but it does not perpetuate the valid behaviors and leads to poor life-wise performances. Overall, we highlight the ability of the two mechanisms to discover novel and unexpected solutions, demonstrating the flexibility of the approach.

The obtained results suggest that a future development of the local adaptive mechanism should consider the possibility to occasionally perform a completely random search, or in any case a destructive action, to try to escape from local maxima. This includes also the possibility to dynamically modulate the perturbation of the best solution, *à la* Variable Neighborhood Search [39]. Finally, we aim to improve the life-long performance by devising a mechanism to more efficiently reuse the knowledge acquired by the robot during its life.

REFERENCES

[1] D. Floreano and F. Mondada, "Automatic creation of an autonomous agent: Genetic evolution of a neural network driven robot," in *Proc. 3rd Int. Conf. Simulation Adapt. Behav., From Animals Animats*, vol. 3. Cambridge, MA, USA: MIT Press, 1994, pp. 421–430.

[2] S. Nolfi and D. Floreano, *Evolutionary Robotics*. Cambridge, MA, USA: MIT Press, 2000.

[3] G. Francesca, M. Brambilla, A. Brutschy, L. Garattoni, R. Miletitch, G. Podevijn, A. Reina, T. Soleymani, M. Salvaro, C. Pinciroli, F. Mascia, V. Trianni, and M. Birattari, "AutoMoDe-chocolate: Automatic design of control software for robot swarms," *Swarm Intell.*, vol. 9, nos. 2–3, pp. 125–152, Sep. 2015.

[4] M. Birattari, A. Ligot, D. Bozhinoski, M. Brambilla, G. Francesca, L. Garattoni, D. Garzón Ramos, K. Hasselmann, M. Kegeleirs, J. Kuckling, F. Pagnozzi, A. Roli, M. Salman, and T. Stützle, "Automatic off-line design of robot swarms: A manifesto," *Frontiers Robot. AI*, vol. 6, p. 59, Jul. 2019.

[5] R.A. Brooks, "Artificial life and real robots," in *Proc. 1st Eur. Conf. Artif. Life*, 1992, pp. 3–10.

[6] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," in *Proc. Eur. Conf. Artif. Life*, Granada, Spain, Jun. 1995, pp. 704–720.

[7] M. Birattari, "Personal communication," IRIDIA, CoDE, Université libre de Bruxelles, Brussels, Belgium, Tech. Rep., 2023.

[8] A. Ram, R. C. Arkin, K. Moorman, and R. J. Clark, "Case-based reactive navigation: A method for on-line selection and adaptation of reactive robotic control parameters," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 27, no. 3, pp. 376–394, Jun. 1997.

[9] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multirobot cooperation," *IEEE Trans. Robot. Autom.*, vol. 14, no. 2, pp. 220–240, Apr. 1998.

[10] S. Yamada and J. Saito, "Adaptive action selection without explicit communication for multirobot box-pushing," *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 31, no. 3, pp. 398–404, Aug. 2001.

[11] D. Floreano and J. Urzelai, "Evolutionary robots with on-line self-organization and behavioral fitness," *Neural Netw.*, vol. 13, nos. 4–5, pp. 431–443, Jun. 2000.

[12] W. R. Ashby," *Design for a Brain: The Origin of Adaptive Behaviour*, 2nd ed. Seattle, WA, USA: Butler & Tanner, 1954.

[13] M. Braccini, A. Roli, and S.A. Kauffman, "Online adaptation in robots as biological development provides phenotypic plasticity," 2020, *arXiv:2006.02367*.

[14] M. Braccini, A. Roli, and S. Kauffman, "A novel online adaptation mechanism in artificial systems provides phenotypic plasticity," in *Proc. Italian Workshop Artif. Life Evol. Comput. (WIVACE)*, vol. 1722, J. J. Schneider, M. S. Weyland, D. Flumini, R. M. Füchslin, Eds. Cham, Switzerland: Springer, 2022, pp. 121–132.

[15] M. Braccini, A. Roli, E. Barbieri, and S. Kauffman, "On the criticality of adaptive Boolean network robots," *Entropy*, vol. 24, no. 10, p. 1368, Sep. 2022.

[16] P. Baldini, "Online adaptation of robots controlled by nanowire networks," M.S. thesis, Dept. Comput. Sci. Eng. (DISI), Campus Cesena, Alma Mater Studiorum—Università di Bologna, Cesena, Italy, 2022.

[17] P. Baldini, M. Braccini, and A. Roli, "Online adaptation of robots controlled by nanowire networks: A preliminary study," in *Proc. Italian Workshop Artif. Life Evol. Comput.*, vol. 1780, C. De Stefano, F. Fontanella, and L. Vanneschi, Eds. Cham, Switzerland: Springer, 2023, pp. 171–182.

[18] G. Milano, G. Pedretti, M. Fretto, L. Boarino, F. Benfenati, D. Ielmini, I. Valov, and C. Ricciardi, "Brain-inspired structural plasticity through reweighting and rewiring in multi-terminal self-organizing memristive nanowire networks," *Adv. Intell. Syst.*, vol. 2, no. 8, Aug. 2020, Art. no. 2000096.

[19] G. Milano, G. Pedretti, K. Montano, S. Ricci, S. Hashemkhani, L. Boarino, D. Ielmini, and C. Ricciardi, "In materia reservoir computing with a fully memristive architecture based on self-organizing nanowire networks," *Nature Mater.*, vol. 21, no. 2, pp. 195–202, Feb. 2022.

[20] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, Mar. 1969.

[21] G. Milano, E. Miranda, and C. Ricciardi, "Connectome of memristive nanowire networks through graph theory," *Neural Netw.*, vol. 150, pp. 137–148, Jun. 2022.

[22] K. Fu, R. Zhu, A. Loeffler, J. Hochstetter, A. Diaz-Alvarez, A. Stieg, J. Gimzewski, T. Nakayama, and Z. Kuncic, "Reservoir computing with neuromemristive nanowire networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[23] D. V. Christensen et al., "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Comput. Eng.*, vol. 2, no. 2, p. 022501, May 2022.

[24] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, Apr. 2010.

[25] G. Milan and C. Ricciardi, "Nanowire memristor as artificial synapse in random networks," in *Intelligent Nanotechnology*. Amsterdam, The Netherlands: Elsevier, 2023, pp. 219–246.

[26] G. Milano, S. Porro, I. Valov, and C. Ricciardi, "Recent developments and perspectives for memristive devices based on metal oxide nanowires," *Adv. Electron. Mater.*, vol. 5, no. 9, Sep. 2019, Art. no. 1800909.

[27] D.O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. London, U.K.: Psychology Press, 1949.

[28] T. Ziemke and M. Thieme, "Neuromodulation of reactive sensorimotor mappings as a short-term memory mechanism in delayed response tasks," *Adapt. Behav.*, vol. 10, no. 3, pp. 185–199, Jul. 2002.

[29] A. F. T. Winfield, *Towards an Engineering Science of Robot Foraging*. Berlin, Germany: Springer, 2009, pp. 185–192.

[30] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proc. 9th Conf. Auto. Robot Syst. Competitions*, 2009.

[31] P. Baldini, "Mandrab/online-MNW-robot-learning: Article—On the performance of online adaptation of robots controlled by nanowire network," Dept. Comput. Sci. Eng. (DISI), Campus Cesena, Alma Mater Studiorum—Università di Bologna, Cesena, Italy, Tech. Rep., Jul. 2023, doi: 10.5281/zenodo.8160748.

[32] P. Baldini, "Mandrab/NN_simulator: V1.0.0 [cpu]," Dept. Comput. Sci. Eng. (DISI), Campus Cesena, Alma Mater Studiorum—Università di Bologna, Cesena, Italy, Tech. Rep., Jul. 2023, doi: 10.5281/zenodo.8162579.
[33] P. Baldini, "Dataset—On the performance of online adaptation of robots controlled by nanowire networks," Dept. Comput. Sci. Eng. (DISI), Campus Cesena, Alma Mater Studiorum—Università di Bologna, Cesena, Italy, Tech. Rep., Jul. 2023, doi: 10.5281/zenodo.8160789.
[34] H. H. Hoos and T. Stützle," *Stochastic Local Search: Foundations and Applications*. San Mateo, CA, USA: Morgan Kaufmann, 2005.
[35] K. C. Cheung, "Implantable microscale neural interfaces," *Biomed. Microdevices*, vol. 9, no. 6, pp. 923–938, Oct. 2007.
[36] P. Aaser, M. Knudsen, O. H. Ramstad, R. van de Wijdeven, S. Nichele, I. Sandvig, G. Tufte, U. S. Bauer, Ø. Halaas, S. Hendseth, A. Sandvig, and V. Valderhaug, "Towards making a cyborg: A closed-loop reservoir-neuro system," in *Proc. 14th Eur. Conf. Artif. Life ECAL*, Sep. 2017, pp. 430–437.
[37] G. Milano, M. Luebben, Z. Ma, R. Dunin-Borkowski, L. Boarino, C. F. Pirri, R. Waser, C. Ricciardi, and I. Valov, "Self-limited single nanowire systems combining all-in-one memristive and neuromorphic functionalities," *Nature Commun.*, vol. 9, no. 1, p. 5151, Dec. 2018.
[38] M. Dueweke, U. Dierker, and A. Hübler, "Self-assembling electrical connections based on the principle of minimum resistance," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 54, no. 1, pp. 496–506, Jul. 1996.
[39] P. Hansen and N. Mladenović, "Variable neighborhood search: Principles and applications," *Eur. J. Oper. Res.*, vol. 130, no. 3, pp. 449–467, May 2001.

PAOLO BALDINI was born in Cesena, Emilia Romagna, Italy, in 1997. He received the bachelor's and master's degrees in computer science and engineering from the Alma Mater Studiorum—University of Bologna, Italy, in 2019 and 2022, respectively, where he is currently pursuing the Ph.D. degree.

In 2020, he participated in the Erasmus+ Program, attending lectures from the European Master on Advanced Robotics (EMARO) with the Warsaw University of Technology, Poland. His research interests include the development of biologically inspired mechanisms for the autonomous adaptation of robotic behaviors and the field of complex systems and in the interaction between computer science and other research areas, like physics and electronics. His last works consisted in experimenting with novel memristive devices and reservoir computing techniques in a robotic context.

ANDREA ROLI received the Laurea degree in electronic engineering and the Ph.D. degree in computer science from the University of Bologna. He is currently an Assistant Professor ("Ricercatore Confermato") with the Department of Computer Science and Engineering, University of Bologna. His main research interests include complex systems and artificial intelligence, with a focus on biological models, biorobotics, collective intelligence, gene regulation models for robot control, and information theory techniques for the analysis of complex systems dynamics. He teaches courses in computer science basics, artificial intelligence, and complex systems.

MICHELE BRACCINI was born in Cesena, Italy, in 1991. He received the M.S. degree (cum laude) in computer science and engineering, in March 2016, and the Ph.D. degree in computer science and engineering from the University of Bologna, Italy, in 2020.

He is currently a Research Fellow with the University of Bologna. His research interests include the study of adaptation mechanisms for the control of (swarms of) robots inspired by phenotypic plasticity and symbiotic phenomena. Also, his studies concern the development of simplified models of interaction to create new types of signals and thus novelty and creation of meaning in artificial contexts. During the Ph.D. degree, he worked on computational models of gene regulatory networks for studying the process of cell differentiation and their applications as control software for robots.

Open Access funding provided by 'Alma Mater Studiorum - Università di Bologna' within the CRUI CARE Agreement