Proteus-I: A Flexible and Adaptable Low-Cost General-Purpose Micro-Robot Prototype for Swarm Robotics

Nureddin Moustafa
University of Cantabria
Santander, Spain
nureddin.moustafa@alumnos.unican.es

Andrés Iglesias
Toho University
Funabashi, Japan
University of Cantabria
Santander, Spain
iglesias@unican.es

Akemi Gálvez
Toho University
Funabashi, Japan
University of Cantabria
Santander, Spain
galveza@unican.es

***Abstract*—Swarm intelligence is a subfield of artificial intelligence based on the collective behavior of decentralized, self-organized systems comprised of very simple individual interacting locally with one another and with their environment. Such interactions lead to the emergence of intelligent behavior, unknown to the individual agents. One of the most remarkable applications of swarm intelligence is swarm robotics, Swarm robotics is one of the most exciting developments in artificial intelligence during the last few years is swarm robotics, where expensive and sophisticated robots can be replaced by a swarm of simple inexpensive micro-robots. In this context, this paper introduces *Proteus-I*, a flexible and adaptable low-cost general-purpose micro-robot prototype especially designed for swarm robotics. The robot has been designed to support the most popular swarm intelligence algorithms by either hardware or software. The paper describes the main components of the micro-robot along with some of the most important features to support swarm intelligence algorithms for swarm robotics.**

***Index Terms*—artificial intelligence, human-robot interaction, swarm intelligence, micro-robot prototype, swarm robotics**

I. Introduction

A. Swarm intelligence

One of the most striking discoveries in artificial intelligence during the last decades is the realization that highly sophisticated behaviors can be obtained from the aggregation of very simple behavioral patterns by unsophisticated agents collaborating together to solve a complex problem. This field, commonly known as *swarm intelligence*, is modifying drastically the previous belief that complex behaviors by social groups can only be obtained from a high individual intelligence of its members.

Nature provides a strong source of inspiration for swarm intelligence. Consider, for instance, the dynamics of natural groups such as the colonies of social insects (ants, termites, bees, fireflies), which as a group are able to construct complex nests and carry out many different sophisticated tasks totally impossible for their individuals members. Another typical example is the behavior of a flock of birds when moving all together following a common tendency in their displacements. Other examples from nature include animal herding, fish schooling, and many others.

Work funded by EU Horizon 2020 research and innovation programme (PDE-GIR project, MSCA grant agreement No 778035) and grant #TIN2017-89275-R of Spanish Agency of Research (AEI) and European Funds EFRD.

Roughly speaking, swarm intelligence can been defined as *"the property of a system whereby the collective behaviors of (unsophisticated) agents interacting locally with one another and with their environment cause coherent functional global patterns to emerge"* [8]. Under this new paradigm, there is no a centralized intelligence controlling the swarm, taking decisions, and sending orders to the swarm members about how to behave. Individual agents in a swarm intelligence system follow simple rules and have a very limited knowledge and intelligence. However, as a whole, the social group is able to carry out difficult tasks that its individual members cannot intend for. Such complex collective behaviors emerge from a small set of simple behavioral rules exploiting only low-level interactions between individuals and with the environment (stigmergy) using decentralized control and self-organization. In this way, the limited intelligence of swarm units is amplified by their (local or global) interactions. Members of the swarm have the ability to communicate with each other and with the environment, thus enhancing the global intelligence of the swarm. The interested reader is referred to [8], [12], [39] for a comprehensive overview about the field of swarm intelligence, its history, main techniques, and applications.

B. Swarm robotics

Nowadays, swarm intelligence is attracting a lot of attention from the scientific and industrial community due to its potential application in several fields. For instance, military and civil applications related to the control of unmanned vehicles have been reported in the literature [25], [26], [36]. It has also been shown that self-organizing swarm robots can potentially accomplish complex tasks and thus replace sophisticated and expensive robots by simple inexpensive drones, a research subfield usually referred to as *swarm robotics* [2], [9], [29]–

978-1-7281-2741-5/19/$31.00 ©2019 IEEE

[32], [37], [43]. As remarked by several authors [1], [5], swarm robotic systems offer several interesting advantages, such as:

- *Improved performance by parallelization*: swarm intelligence systems are very well suited for parallelization, because the swarm members can perform different actions at different locations simultaneously. This feature makes the swarm more flexible and efficient for complex tasks, as individual robots (or groups of them) can solve different parts of a complex task independently.
- *Task enablement*: groups of robots can do certain tasks that are impossible or very difficult for a single robot (e.g., collective transport of too heavy items, dynamic target tracking, cooperative environment monitoring, autonomous surveillance of large areas).
- *Scalability*: inclusion of new robots into a swarm does not require reprogramming the whole swarm. Furthermore, because interactions between robots involve only neighboring individuals, the total number of interactions within the system does not increase dramatically by adding new units, making the system highly scalable.
- *Distributed sensing and action*: a swarm of simple interconnected mobile robots deployed throughout a large search space possesses greater exploratory capacity and a wider range of sensing than a sophisticated robot. This makes the swarm much more effective in several tasks: exploration and navigation (e.g., in disaster rescue missions), nanorobotics-based manufacturing, microbotics for human body diagnosis, and many others.
- *Fault tolerance*: due to the decentralized and self-organized nature of the swarm, the failure of a single unit does not affect the completion of the given task.

All these advantages motivated a great interest in swarm robotics during the last two decades. The reader is kindly referred to Section II for further details on this issue.

C. Aims and structure of this paper

In this context, the present paper presents *Proteus-I*, a micro-robot prototype for swarm robotics. This prototype has been designed to meet four important differential features:

1. *It is a low-cost prototype*. Many approaches in robotics contain highly sophisticated components, making the robots very expensive. Opposed to these expensive alternatives, this approach is designed to be not only cost-effective but also low-cost. In this implementation, a single robotic unit costs less than 50US$.
2. *It is very flexible and adaptable*. This implementation is very flexible, in the sense that its different components can be added or removed very easily. To this aim, the present design is fully modular and the robot does not contain fixed parts. It is also very adaptable, meaning that the robot can be adapted or modified by simple removal, replacement, or addition of different components in order to meet different goals.
3. *It is a general-purpose robot*. Instead of a specialized goal-oriented robotic platform, the present design leads to a general-purpose robot. This feature is a consequence of the previous one, as we can simply modify some components (e.g., the sensors) for different purposes or applications.
4. *It is well suited for swarm robotics*. Despite its low-cost design and implementation, the robot has enough computing power to support the most important swarm intelligence techniques running locally at software level. Furthermore, the robots of the swarm are highly interconnected with one another via standard communication interfaces.

The structure of this paper is as follows: Section II describes some previous work in the field. Then, Section III describes our design and implementation of the micro-robot, including the hardware architecture and main components, its assembly and the programming framework. Finally, Section IV revisits some of the most popular swarm intelligence methods and discuss the main features of our micro-robot *Proteus-I* to support such methods for swarm robotics.

II. Previous Work

Swarm robotics has attracted much attention from the scientific community and the industry during the last decades. Pioneering research projects dating back the 80s and 90s (such as CEBOT, SWARMS or ACTRESS) made just preliminary advances, yet they set the foundations of the field. Since then, ambitious large-scale projects, such as the European-funded projects Swarm-bots (2001-2005), i-Swarm (2004-2008) and Swarmanoid (2006-2010) and initiatives from USA universities such as Harvard, MIT, Stanford, UPenn, ASU, Texas A&M and many others, have led to a boost in swarm robotics. As a result, several approaches have been described in the literature during the last few years.

One of the first approaches in the field was the robot *Khepera* [17], developed at the EPFL (Lausanne, Switzerland) in mid 90s. Subsequently versions (such as *Khepera III* [20]) were released for the following decade along with some simulation platforms. This work was later extended to a new robot, *Alice* [6], [7], developed by Gilles Caprari at the Autonomous Systems Lab at EPFL. From the very beginning, *Alice* designed to be autonomous, small (with just under 1 cubic inch, it featured a very amazingly small size for its time), and relatively inexpensive, making it affordable to construct and operate a large crowd of robots simultaneously (such as swarms of 90 robots working cooperatively, a very impressive number for that time). They were also highly configurable, with numerous extension modules available for different research purposes.

A larger (CD size) robot was *Kobot* [33], [34]. It was applied to analyze the self-organized flocking of a swarm of robots moving as a coherent group and avoiding obstacles as if they were a single "super-organism". The *e-Puck* was a popular cylindrical-shaped micro-robot designed to be robust and affordable enough to allow intensive classroom use [19]. It was quite popular for academic purposes, particularly in small amounts. However, it was a little bit big and quite expensive for large bulks, as required for crowds in swarm robotics.

Another popular micro-robot was *Jasmine*, a public open-hardware development to create a simple and cost-effective micro-robotics platform [27]. This micro-robot has been widely used in swarm robotics studies, such as playing the role of a honeybee in aggregation scenarios [10]. A differential feature of *Jasmine* with respect to other swarms of robots is that Jasmine only supports local communication, while long distance communication is neither intended nor implemented, making it a good testbed for many problems. Other micro-robots well suited for swarm robotics are *S-bot* [18], *i-Swarm robot* [35], *SwarmBot* [13], *AMiR* (Autonomous Miniature Robot) [2], *Colias* [3] and its evolution, *Colias*-Φ [4].

Motivated by the recent trend of increasing the number of robots in the swarm, several relevant works are focusing on scalability issues and the challenges they present. Crowds of up to 100 robots (and even more) has been analyzed in projects, such as the *I-swarm* and the *iRobot* swarm projects, by using *R-one*, *iRobot*, *SwarmBot* and other micro-robotics models [14]–[16], [24], [28].

A significant step in this regard was the small *kilobot*, by Michael Rubenstein, at the Self Organizing Systems Research Group at Harvard University [21]. It is a small, cheap and very simple microrobot. It can only perform three simple tasks: respond to light, measure a distance, and sense the presence of other robots. However, when combined, they can organize themselves into shapes, such as grouping into clusters based on their own color light (or that of their neighbors) or dispersing to fill a space. Kilobots are programed all at once, as a group, using infrared light. Recently, a swarm of one thousand kilobots has been reported in the literature [22]. With this huge amount, individual units are not really important; it does not even matter if one or a few robots break down, as the collective behavior of the swarm still prevails. In other words, a large robotic swarm provides a lot of flexibility and robustness, as the collective tendency of the swarm is highly immune to individual failures or breakages [23]. It also gives room to analyze some interesting configurations; for instance, a gradient formation, where a source robot generates a gradient value that is incremented as it propagates through the swarm, giving each robot a metrics of the distance value from the source, or for localization tasks, where robots determine their position in the coordinate system by communicating with already localized robots [22], [23].

III. Design and Implementation

A. Hardware architecture and components

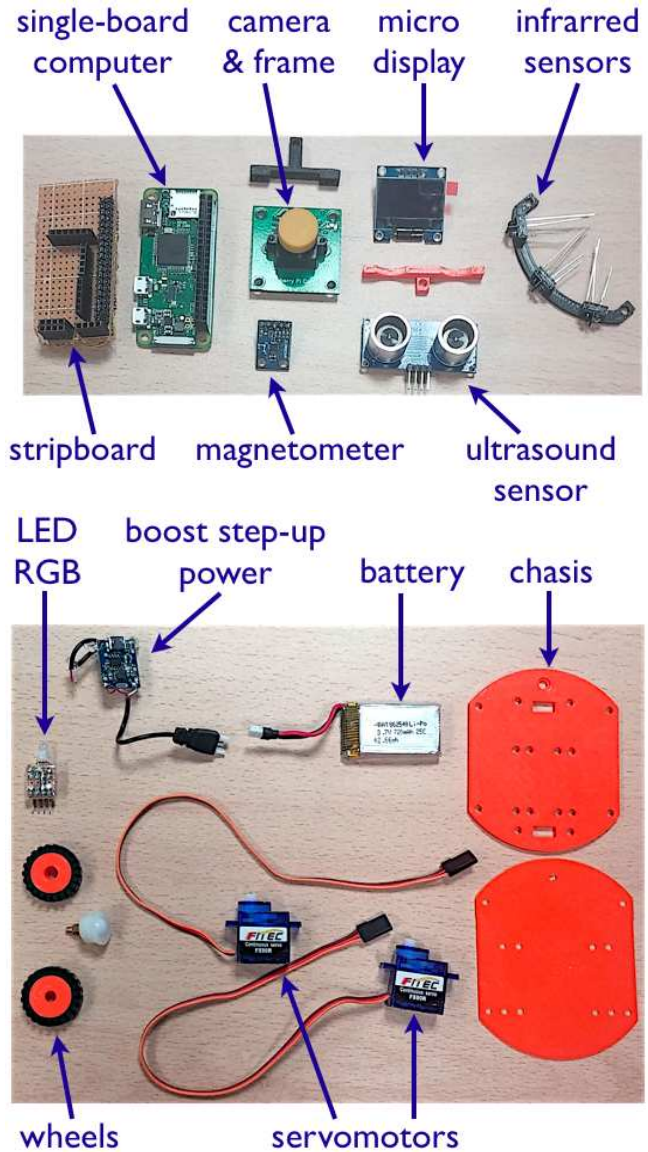

Fig. 1: Main components of the micro-robot *Proteus-I*.

Figure 1 shows the different components used in *Proteus-I*:

1) the *chassis*: the robot is mounted on a rigid chassis, shown in orange color in that figure. The chassis and other mechanical parts such as the holders have been generated by 3D printing from a 1.75 mm PLA (polylactic acid) filament by using a fully enclosed domestic desktop 3D printer. The robot moves through two side wheels driven by two continuous rotation servomotors, displayed in the picture. Unlike ordinary motors, servo motors can be individually controlled. Only the indication of the angle of rotation for moving the motor is necessary.
2) a *battery*: the chassis hosts the 3.7V battery with its board connectors and the electronics of our robotic unit.
3) a *boost step-up power*: in our implementation, we also include the module *Lithium 134n3p* charger, a built-in charge and discharge power MOS operating at an input voltage in the range 3.7V∼5.5V with output voltage 5V and providing charge and discharge management, temperature control and protection against over-temperature, output over-voltage, short circuit, heavy load over-charge and over-discharge. This component is

required for the battery to be voltage-compatible.

4) a *single-board micro-computer*: in this implementation we use the popular micro-computer *Raspberry Pi Zero W*, one of the most affordable and cost-effective micro-computers in the market, with a prices of 10US$. This micro-computer comes with a 32-bit RISC ARMv6Z architecture, featuring a Broadcom BCM2835 system on a chip application processor. Its CPU is the ARM1176JZF-S core by ARM, running at 1GHz. It also includes a graphical processor unit Broadcom Video Core IV running at 250 MHz, with support to Open GL and featuring a H.264/MPEG-4 AVC high-profile decoder and encoder with support to 1080p (high-definition video mode). The system comes with 512 MB (shared with the GPU), 1micro-USB (direct from the BCM2835 chip), a MIPi camera interface for video input, mini-HDMI at 1080p resolution and composite video for video output, 2 boards via the serial bus I^2S for audio input, a stereo audio through PWM on GPIO for audio output, and a MicroSDHC non-volatile memory card for data storage. The card also provides support for communications via Bluetooth 4.1 for very short distances (about a range of 10 meters or less), 802.11n wireless LAN for wider areas (up to 100 meters), and a FM receiver working in the range 65 MHz to 108 MHz FM bands, all through the Cypress CYW43438 wireless chip. It also has an unpopulated HAT-compatible 40-pin GPIO header, and composite video and reset headers. These wireless communication options are used for communication and data exchange over short distances among the robots of the swarm and with a central server for tracking purposes.
5) a *stripboard*: for further connectivity of electronic components. This stripboard is located close to the board chip to gain access to all micro-computer pins.
6) an *ultrasound sensor*: in this work, we use the ultrasound sensor *HC-SR04*, manufactured by *ElecFreaks*. It is an ultrasonic sensor operating at 5V DC that uses sonar to compute the distance to an object, much like bats or dolphins actually do. Each *HC-SR04* module includes an ultrasonic transmitter, a receiver and a control circuit, with 4 pins for power, trigger (transmitter), echo (receiver), and ground. Each ultrasound pulse of our sensors operates at a constant frequency of 40 kHz, sending an 8 cycle burst of ultrasound pulses. The sensor captures its echo with signals lasting in the order of milliseconds. For this sensor, the manufacturer advises to consider over 60 ms measurement cycle, in order to prevent trigger signal to the echo signal. The accuracy range of the sensor is about 3 millimeters, with a traveling range of pulses between 2–500 centimeters. The ultrasound sensor is used for collision avoidance with static and dynamic objects (including other robots in the swarm) as well as with the boundaries of the physical 3D environment.
7) a LED RGB: a hand-made diode with RGB lights used to indicate the different robot states, such as *active*, *idle*, *sleep*, and others.

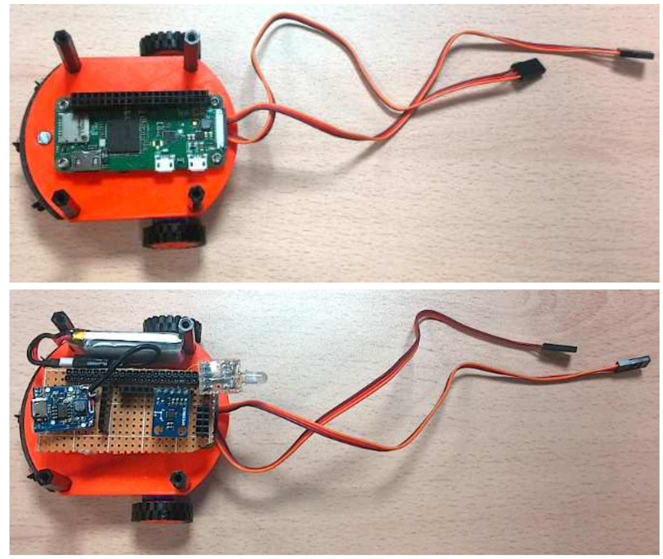

Fig. 2: Two steps of the assembly process of the micro-robot.

8) a *magnetometer*: in this work, we use the triple-axis magnetometer board *HMC-5883L*. This user-friendly compass is a 3.3V max chip with added circuitry to make it 5V-safe logic and power, so that it can be connected to either 3 or 5V microcontrollers. It uses I2C serial bus for easy interface to communicate. Its internal functioning is based on the anisotropic magnetoresistive (AMR) technology by *Honeywell*, with AMR directional sensors having a full range of ± 8 gauss an a resolution of up to 2 milligauss. The magnetometer is used in this work for global spatial orientation of the robotic units of the swarm regardless their physical environment.
9) *infrared sensors*: used for collision avoidance, scene exploration and navigation throughout the 3D environment. In this work we consider a set of three infrared sensors deployed on a semi-ring holder located in the front of the robot to cover a wider exploration area. One of the IR sensors is located in the middle, and the other two near the corners in the front. They can detect obstacles in the range of 30 centimeters in the sun light. They also come with a variable resistance to adapt the sensors to different short distances.
10) a *mini-camera*: used for image capture and navigation. Our model provides a resolution of 3280×2464 pixels, and support video capture at a frame rate of 30 FPS (frames per second) for a resolution of 1080p, 60 FPS at 720p and 90 FPS at 480p.
11) an *OLED micro-display*: it is a SDD1306 chipset with a serial connection I^2C, a size of 0.96 inches and power consumption of 20mA, it provides a resolution of 128×64 pixels, with a vision angle of 160 degrees. It is mainly used to display relevant information for tracking purposes and also as a user-interface with the board.

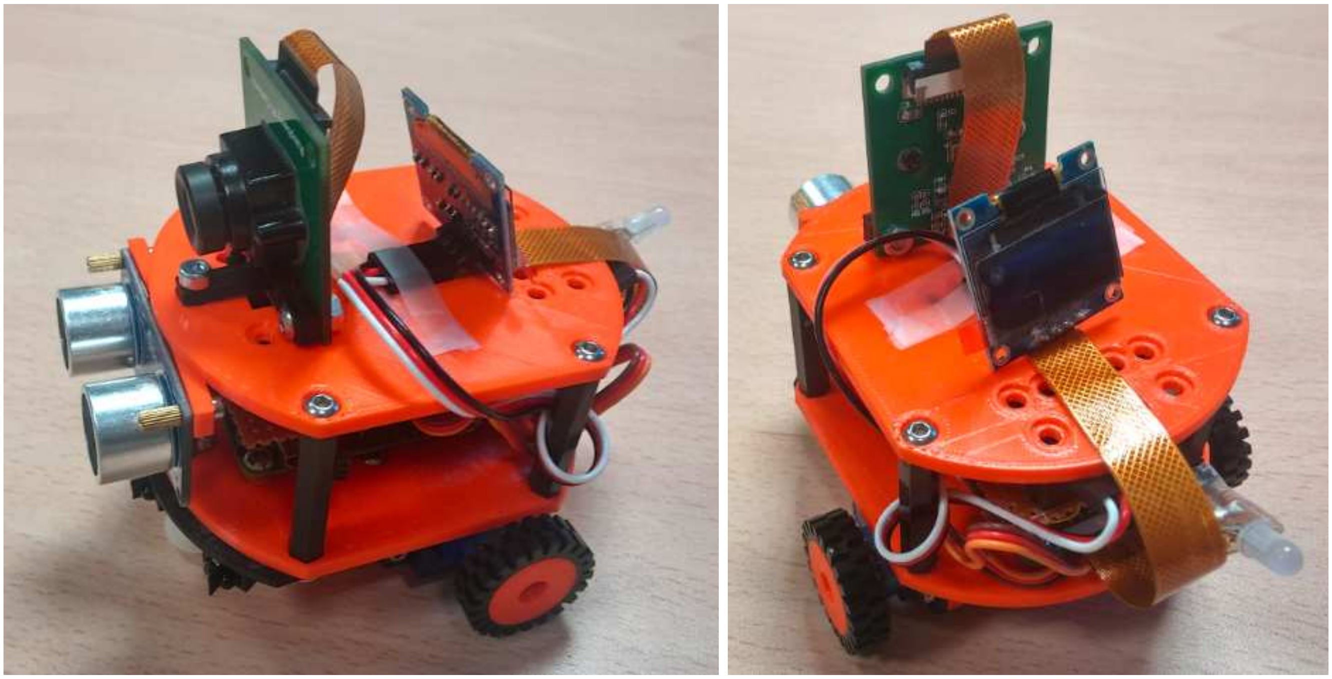

Fig. 3: Two views of the final assembled micro-robot *Proteus-I*.

All these components are connected to different I/O pins in a rather standard way. The detailed description of these connections is out of the scope of this paper and will be omitted here to keep the paper to a manageable size.

B. Robot assembly

Figure 2 and 3 show different steps of the assembly process of the robot and the final assembled robot. As shown in the pictures, the micro-robot consists of two vertical layers: the lower layer, shown in Figure 2, hosts the single-board micro-computer and the infrared sensors attached to the front part (top picture), the battery, the magnetometer, the boost step-up power, the LED RGB, and the stripboard for additional connections (bottom picture). The upper layer supports the camera, the OLED micro-display, and the ultrasound sensors. Figure 3 shows the final assembled micro-robot from two different viewpoints.

C. Programming framework

There are several operating systems and programming frameworks that can be used for the Raspberry Pi Zero micro-computer. Among all possibilities for the operating system, we recommend to use *Raspbian*, a free operating system based on *Debian*, optimised for the *Raspberry Pi* hardware. *Raspbian* comes with over 35,000 packages and is the recommended operating system for normal use on a *Raspberry Pi*. Other options are, for instance, Ubuntu MATE, Snappy Ubuntu, Pidora, Linutop, SARPi. Arch Linux ARM, Gentoo Linux, FreeBSD, Kali Linux, RISC OS Pi and many ohers.

Regarding the programming framework, it is strongly dependent on the programming language of choice. In this implementation, we use Phyton running on Wing IDE. We remark that Wing IDE is not directly supported by the Raspberry Pi, but it is possible o set up Wing IDE on a computer connected to the Raspberry Pi to work on and debug Python code remotely. Other possibilities include Kivy, KivyPie, GTK+, pygtk, PyQt, and Glade.

IV. Support for Swarm Intelligence Algorithms

In this section, some of the most popular swarm intelligence algorithms are briefly revisited. Then, we discuss the main features of the micro-robot *Proteus-I* to support the implementation of such algorithms by either hardware or software.

A. Particle swarm optimization

Particle Swarm Optimization (PSO) is a global stochastic optimization algorithm for dealing with problems where potential solutions (called *particles*) can be represented as vectors in a n-dimensional search space [11], [12]. Particles are distributed over such space and provided with an initial velocity and the capacity to communicate with other neighbor particles, even the entire swarm. Particles 'flow" through the solution space and are evaluated according to some fitness function after each instance. Particles evolution is regulated by two memory factors: memory of their own best position and knowledge of the global or their neighborhood's best. Particles of a swarm communicate good positions to each other and adjust their own position and velocity based on these good positions. As the swarm iterates, the fitness of the global best solution improves so the swarm eventually reaches the best solution.

The dynamics of the particle swarm is considered along successive iterations, like time instances. Each particle modifies its position P_i along the iterations by storing the coordinates

P_i^b associated with the best solution (fitness) it has achieved so far. These values account for the *memory* of the best particle position. In addition, members of a swarm can communicate good positions to each other, so they can adjust their own position and velocity according to this information. To this purpose, we also collect the best global position P_g^b from the initial iteration. The evolution for each particle i is given by:

$$\begin{aligned} V_i(k+1) &= w\, V_i(k) + \gamma_1 R_1 [P_g^b(k) - P_i(k)] + \\ & \quad \gamma_2 R_2 [P_i^b(k) - P_i(k)] \\ P_i(k+1) &= P_i(k) + V_i(k) \end{aligned} \quad (1)$$

where $P_i(k)$ and $V_i(k)$ are the position and the velocity of particle i at time k, respectively, w is called *inertia weight* and decide how much the old velocity will affect the new one and coefficients γ_1 and γ_2 are constant values called *learning factors*, which decide the degree of affection of P_g^b and P_i^b. This procedure is repeated several iterations until a termination condition is reached.

B. Firefly algorithm

The *firefly algorithm* (FFA) is a nature-inspired metaheuristic algorithm introduced in 2009 by to solve optimization problems [38]. The algorithm is based on the social flashing behavior of fireflies in nature. The key ingredients of the method are the variation of light intensity and formulation of attractiveness. In general, the attractiveness of an individual is assumed to be proportional to their brightness, which in turn is associated with the encoded objective function. In the firefly algorithm, there are three particular idealized rules, based on some of the major flashing characteristics of real fireflies:

1) All fireflies are unisex, so that one firefly will be attracted to other fireflies regardless of their sex;
2) The degree of attractiveness of a firefly is proportional to its brightness, which decreases as the distance from the other firefly increases due to the fact that the air absorbs light. For any two flashing fireflies, the less brighter one will move towards the brighter one. If there is not a brighter or more attractive firefly than a particular one, it will then move randomly;
3) The brightness or light intensity of a firefly is determined by the value of the objective function of a given problem. For instance, for maximization problems, the light intensity can simply be proportional to the value of the objective function.

TABLE I: Firefly Algorithm pseudocode

```
begin
    Objective function f(x), x = (x_1, ..., x_D)^T
    Generate initial population of n fireflies x_i (i = 1, 2, ..., n)
    Formulate light intensity I associated with f(x)
    Define absorption coefficient γ
    while (t < MaxGeneration) or (stop criterion)
        for i = 1 to n
            for j = 1 to n
                if I(i) > I(j))
                    then move firefly i towards j
                end if
                Vary attractiveness with distance r via e^{-γr}
                Evaluate new solutions and update light intensity
            end for
        end for
            Rank fireflies and find the current best
    end while
    Post-processing the results and visualization
end
```

C. Bat algorithm

The *bat algorithm* (BA) is a bio-inspired swarm intelligence algorithm originally proposed in 2010 to solve optimization problems [40], [41]. The algorithm is based on the echolocation behavior of microbats, which use a type of sonar called *echolocation*, with varying pulse rates of emission and loudness, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. The idealization of the echolocation of microbats is as follows:

1) Bats use echolocation to sense distance and distinguish between food, prey and background barriers.
2) Each virtual bat flies randomly with a velocity $\mathbf{v}_i$ at position (solution) $\mathbf{x}_i$ with a fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for prey. As it searches and finds its prey, it changes the frequency of their pulses and adjust the rate of pulse emission r, depending on the proximity of the target.
3) It is assumed that the loudness will vary from an (initially large and positive) value A_0 to a minimum constant value A_{min}.

With these idealized rules indicated above, the basic pseudocode of the bat algorithm is shown in Algorithm 1. Basically, the algorithm considers an initial population of $\mathcal{P}$ individuals (bats). Each bat, representing a potential solution of the optimization problem, has a location $\mathbf{x}_i$ and velocity $\mathbf{v}_i$. The algorithm initializes these variables with random values within the search space. Then, the pulse frequency, pulse rate, and loudness are computed for each individual bat. Then, the swarm evolves in a discrete way over generations, like time instances until the maximum number of generations, $\mathcal{G}_{max}$, is reached. For each generation g and each bat, new frequency, location and velocity are computed as:

$$f_i^g = f_{min}^g + \beta(f_{max}^g - f_{min}^g) \quad (2)$$

$$\mathbf{v}_i^g = \mathbf{v}_i^{g-1} + [\mathbf{x}_i^{g-1} - \mathbf{x}^*]\, f_i^g \quad (3)$$

$$\mathbf{x}_i^g = \mathbf{x}_i^{g-1} + \mathbf{v}_i^g \quad (4)$$

where $\beta \in [0, 1]$ follows the random uniform distribution, and $\mathbf{x}^*$ represents the current global best location (solution), which is obtained through evaluation of the objective function at all bats and ranking of their fitness values. The best current solution and a local solution around it are probabilistically selected according to some given criteria. Then, search is intensified by a local random walk. For this local search, once a solution is selected among the current best solutions, it is perturbed locally through a random walk. If the new solution achieved is better than the previous best one, it is probabilistically accepted depending on the value of the

TABLE II: Cuckoo search algorithm pseudocode.

```
begin
    Objective function f(x), x = (x_1, ..., x_D)^T
    Generate initial population of n host nests x_i (i = 1, 2, ..., n)
    while (t < MaxGeneration) or (stop criterion)
        Get a cuckoo (say, i) randomly by Lévy flights
        Evaluate its fitness F_i
        Choose a nest among n (say, j) randomly
        if (F_i > F_j)
            Replace j by the new solution
        end
        A fraction (p_a) of worse nests are abandoned and new ones
                are built via Lévy flights
        Keep the best solutions (or nests with quality solutions)
        Rank the solutions and find the current best
    end while
    Postprocess results and visualization
end
```

TABLE III: Different features of the swarm intelligence methods supported by *Proteus-I*.

PSO	*Computation (real-time)*: supported (ARM1176JZF-S); *Positioning*: magnetometer & built-in digital compass *Sensors*: servomotors (for velocity) *Communication*: Wifi (IEEE 802.11n)
FFA	*Computation (real-time)*:supported (ARM1176JZF-S); *Positioning*: magnetometer & built-in digital compass *Sensors*: front and side infrared sensors *Communication*: Bluetooth 4.1
BA	*Computation (real-time)*: supported (ARM1176JZF-S); *Positioning*: magnetometer & built-in digital compass *Sensors*: ultrasound sensor (HC-SR04) *Communication*: Bluetooth 4.1
CSA	*Computation (real-time)*: supported (ARM1176JZF-S); *Positioning*: magnetometer & built-in digital compass *Sensors*: mini-camera *Communication*: Wifi (IEEE 802.11n)

loudness. In that case, the algorithm increases the pulse rate and decreases the loudness.

D. Cuckoo search algorithm

The *cuckoo search algorithm* (CSA) is another metaheuristic algorithm proposed in 2009 [42] and inspired by the obligate interspecific brood-parasitism of some cuckoo species that lay their eggs in the nests of host birds of other species to escape from the parental investment in raising their offspring and minimize the risk of egg loss to other species. In this algorithm, the eggs in the nest are interpreted as a pool of candidate solutions while the cuckoo egg represents a new coming solution. The method uses these new (and potentially better) solutions associated with the parasitic cuckoo eggs to replace the current solution associated with the eggs in the nest. This replacement, carried out iteratively, will eventually lead to a very good solution. In addition to this representation scheme, CSA is also based on three idealized rules [42]:

1) Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
2) The best nests with high quality of eggs (solutions) will be carried over to the next generations;
3) The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0, 1]$. For simplicity, this assumption can be approximated by a fraction p_a of the n nests being replaced by new nests (with new random solutions at new locations).

The CS algorithm starts with an initial population of n host nests and it is performed iteratively. For each iteration, a cuckoo egg is selected randomly and new solutions are generated by using the Lévy flight. The CS evaluates the fitness of the new solution and compares it with the current one. In case that the new solution brings better fitness, it replaces the current one. On the other hand, a fraction of the worse nests are abandoned and replaced by new solutions to increase the exploration of the search space looking for more promising solutions. The rate of replacement is given by the probability p_a, a parameter of the model that has to be tuned for better performance. Moreover, for each iteration step, all current solutions are ranked according to their fitness and the best solution reached so far is stored as the vector $\mathbf{x}_{best}$.

E. Proteus-I features for swarm intelligence methods

Clearly, the major ingredients of all previous swarm intelligence methods are a population of individuals with the ability to compute their own positions according to some evolution equations and to communicate their positions to other members of the swarm. These two features are incorporated in this implementation. The individuals of the robotic swarm are identical versions of this robot *Proteus-I*. Its *Raspberry Pi Zero W* micro-processor is powerful enough to perform all the required computations in real time, including the global positioning computed in coordination with the magnetometer and its built-in digital compass, so that the robot can determine its current position with acceptable accuracy. In addition, the Wifi and Bluetooth communication protocols supported by the robot as discussed in previous section allow them to communicate their position to other members of the swarm. Finally, the different sensors of the robot can be used to meet the needs associated with the different swarm intelligence algorithms. For instance, the ultrasound sensors are ideal for the bat algorithm while the infrared sensors are particularly well suited for the firefly algorithm. Similarly, the mini-camera is adequate for the cuckoo search algorithm. Finally, particle swarm optimization does not need any particular sensor, but takes advantage of the capacity of the servomotors to compute the distance and the velocity, as required by the swarm intelligence algorithm. All these features are summarized in Table III. From this table, it can be concluded that the robot *Proteus-I* supports all major features of a number of swarm intelligence methods and provides a low-cost general-purpose robotic platform fully suitable for swarm robotics purposes.

ACKNOWLEDGMENT

This research work has received funding from the project PDE-GIR of the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 778035, the Spanish Ministry of Economy and Competitiveness (Computer Science National Program) under grant #TIN2017-89275-R of the Agencia Estatal de Investigación and European Funds FEDER (AEI/FEDER, UE), and the project #JU12, supported by public body SODERCAN and European Funds FEDER (SODERCAN/FEDER UE).

References

[1] Arkin, C.R.: *Behavior-Based Robotics*. MIT Press, Cambridge, Mass, USA (1998).

[2] Arvin, F., Samsudin, K., Ramli, A. R.: Development of a miniature robot for swarm robotic application. *International Journal of Computer and Electrical Engineering*, **1**, 436–442 (2009).

[3] Arvin, F., Murray, J.C., Licheng S., Zhang, C., Yue, S.: Development of an autonomous micro robot for swarm robotics. In: Proc. of IEEE Int. Conf. on Mechatronics and Automation - ICMA'2014, pp. 635–640 (2014).

[4] F. Arvin, Yue, S., Xiong, C.: Colias-Φ: an autonomous micro robot for artificial pheromone communication. *International Journal of Mechanical Engineering and Robotics Research*, **4**(4), 349–353 (2015).

[5] Bonabeau, E., Dorigo, M., Theraulaz. G: *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York (1999).

[6] Caprari, G., Estier, T., Siegwart, R.: Fascination of down scaling - Alice the sugar cube robot. *Journal of Micro-Mechatronics*, **1**(3), 177–189 (2002).

[7] Caprari, G., Siegwart, R.: Mobile micro-robots ready to use: Alice. *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems –IROS 2005*. Edmonton, Canada, 3845–3850 (2005).

[8] Engelbretch, A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley and Sons, Chichester, England (2005).

[9] Faigl, J., Krajnik, T., Chudoba, J., Preucil, L., Saska, M.: Low-cost embedded system for relative localization in robotic swarms. In: Proc. of IEEE Int. Conf. on Robotics and Automation - ICRA'2013, pp. 993–998 (2013).

[10] Kernbach, S., Thenius, R., Kernbach, O., Schmickl, T.: Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system. *Adaptive Behavior*, **17**(3), 237–259 (2009).

[11] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. *IEEE International Conference on Neural Networks*, Perth, Australia 1 (1995) 942-1948

[12] Kennedy, J., Eberhart, R.C., Shi, Y.: *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, CA (2001)

[13] JMcLurkin, J.: Stupid robot tricks: a behavior-based distributed algorithm library for programming swarms of robots. M.S. thesis, *Massachusetts Institute of Technology* (2004).

[14] McLurkin, J., Smith, J.: Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots. In: *Distributed Autonomous Robotic Systems*, **6**, Springer, 399–408 (2007)

[15] McLurkin, J., Lynch, A. Rixner, S. Barr, T. Chou, A.,. Foster, K., Bilstein, S.: A low-cost multi-robot system for research, teaching, and outreach. in: *Distributed Autonomous Robotic Systems*, *83*, Springer, 597–609 (2013).

[16] McLurkin, J., Smith, J. Frankel, J., Sotkowitz, D., Blau, D., Schmidt, B.: Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots. In: *AAAI spring symposium*, 72–75 (2006).

[17] Mondada, F. Franzi, E., Guignard, A.: The development of Khepera. *Proceedings of the 1st International Khepera Workshop*, HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut, **64**, 7–14 (1999).

[18] Mondada, F., Gambardella, L.M., Floreano, D., Nolfi, S., Deneubourg, J.-L., Dorigo, M.: The cooperation of swarm-bots: physical interactions in collective robotics. *IEEE Robotics and Automation Magazine*, **12**(2), 21–28 (2005).

[19] Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Magnenat, S., Zufferey, J.C., Floreano, D., Martinoli, A.: The e-Puck, a robot designed for education in engineering. In: *Proceedings of the 9th conference on autonomous robot systems and competitions*, 1(1), 59–65 (2009).

[20] Pugh, J., Raemy, X., Favre, C., Falconi, R., Martinoli, A.: A fast onboard relative positioning module for multirobot systems. *IEEE/ASME Transactions on Mechatronics*, **14**(2), 151–162 (2009).

[21] Rubenstein, M., Ahler, C., Nagpal, R.: Kilobot: A low cost scalable robot system for collective behaviors. In: *IEEE International Conference on Robotics and Automation – ICRA'2012*, 3293–3298 (2012).

[22] Rubenstein, M., Cornejo, A., Nagpal, N.: Programmable self-assembly in a thousand-robot swarm. *Science*, **345**(6198), 795–799 (2014).

[23] Gauci, M., Ortiz, M.E., Rubenstein, M., Nagpa, N.l: Error Cascades in Collective Behavior: A Case Study of the Gradient Algorithm on 1000 Physical Agents. *International conference on Autonomous Agents and Multi-Agent Systems – AAMAS 2017*, 1404–1412 (2017).

[24] Sahin, E.: Swarm robotics: from sources of inspiration to domains of application. In: Swarm robotics. *Lecture Notes in Computer Science*, **3342**, 10–20 (2005).

[25] Sauter, J. A., Matthews, R., Parunak, H. V. D., Brueckner, S. A.: Effectiveness of digital pheromones controlling swarming vehicles in military scenarios. Journal of Aerospace Computing, Information, and Communication, **4**(5) (2007) 753–769.

[26] Saska, M., Vonasek, V., Krajnik, T., Preucil, L.: Coordination and navigation of heterogeneous MAVUGV formations localized by a hawk-eye-like approach under a model predictive control scheme. International Journal of Robotics Research, **33**(10) (2014) 1393–1412.

[27] Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G,. Kernbach, S., Szymanski, M., Crailsheim, K.: Get in touch: cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, **18**(1), 133–155 (2009).

[28] Seyfried, J., Szymanski, M., Bender, N., Estana, R., Thiel, M., Wörn, H.: The i-swarm project: intelligent small world autonomous robots for micro-manipulation. In: Swarm robotics. *Lecture Notes in Computer Science*, **3342**, 70–83 (2005).

[29] Suárez, P., Iglesias, A.: Bat algorithm for coordinated exploration in swarm robotics. *Advances in Intelligent Systems and Computing*, **514**, 134–144 (2017).

[30] Suárez, P., Gálvez, A., Iglesias, A.: Autonomous coordinated navigation of virtual swarm bots in dynamic indoor environments by bat algorithm. *Lecture Notes in Computer Science*, **10386**, 176–184 (2017).

[31] Suárez, P., Iglesias, A., Gálvez, A.: Make robots be bats: specializing robotic swarms to the bat algorithm. *Swarm and Evolutionary Computation*, 44(1), 113–129 (2019).

[32] Tan, Y., Zheng, Z.Y.: Research advance in swarm robotics. Defence Technology Journal, 9(1) (2013) 18–39.

[33] Turgut A.E., Celikkanat, H., Gokce, F., Sahin E.: Self-organized flocking with a mobile robot swarm. *Int. Conf. on Autonomous Agents and Multiagent Systems – AAMAS 2008*, 39–46 (2008).

[34] Turgut A.E., Celikkanat, H., Gokce, F., Sahin E.: Self-organized flocking in mobile robot swarms. *Swarm Intelligence*, **2**(2) 97–120 (2008).

[35] Valdastri, P. Corradi, P., Menciassi, A., Schmickl,T., Crailsheim, K., Seyfried, J., Dario, P.: Micromanipulation, communication and swarm intelligence issues in a swarm microrobotic platform. *Robotics and Autonomous Systems*, **54**(10) 789–804 (2006).

[36] Vasarhelyi, G., Virgh, C., Tarcai, N., Somorjai, G., Vicsek, T.: Outdoor flocking and formation flight with autonomous aerial robots. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS 2014, pp. 3866–3873 (2014).

[37] Wagner, I., Bruckstein, A.: Special Issue on Ant Robotics. Annals of Mathematics and Artificial Intelligence, **31**(1-4) (2001).

[38] Yang, X.S.: Firefly algorithms for multimodal optimization. *Lectures Notes in Computer Science*, **5792**, 169–178 (2009) .

[39] Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms (2nd. Edition). Luniver Press, Frome, UK (2010).

[40] Yang, X.S.: A new metaheuristic bat-inspired algorithm. *Studies in Computational Intelligence*, **284**, 65–74 (2010).

[41] Yang, X. S.: Bat algorithm for multiobjective optimization. *Int. J. Bio-Inspired Computation*, **3**(5), 267–274 (2011).

[42] Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: Proc. World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–214. IEEE Press, New York (2009).

[43] Zahugi, E.M.H., Shabani, A.M., Prasad, T.V.: Libot: Design of a low cost mobile robot for outdoor swarm robotics. In: Proc. of IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems - CYBER'2012, pp.342–347 (2012)