Nano Communication Networks 12 (2017) 53–62

Contents lists available at ScienceDirect

Nano Communication Networks

journal homepage: www.elsevier.com/locate/nanocomnet

Nano-wireless communications for microrobotics: An algorithm to connect networks of microrobots

Ludovico Ferranti [a,b,*], Francesca Cuomo [a]

[a] Sapienza - University of Rome, DIET, Department of Information Engineering, Electronics and Telecommunications, 18, 00184, Rome, Italy

[b] Northeastern University, Department of Electrical and Computer Engineering, 360 Huntington Avenue, Boston, MA 02115, United States

ARTICLE INFO

Article history:
Received 5 March 2016
Received in revised form
9 December 2016
Accepted 30 January 2017
Available online 3 February 2017

Keywords:
MEMS microrobots
Nano-networks
Terahertz band
Distributed algorithm
Topology formation

ABSTRACT

Micro and nanorobotics represents one of the most challenging sectors of modern robotics. Through batch fabrication of Micro Electro-Mechanical Systems (MEMS), advanced small scale sensing and actuating tasks in a wide area of applications can be performed. Most miniaturized electro-mechanical devices are characterized by low-power and low-memory capacity. The huge number of modular robots introduces the need to explore novel self-reconfiguration algorithms to optimize movement and communication performances in terms of efficiency, parallelism and scalability. Nano-transceivers and nano-antennas operating in the Terahertz Band are already a well acquainted communication paradigm, enforcing nano-wireless networking that can be directly integrated in MEMS microrobots. Several logical topology shape-shifting algorithms are already implemented and tested in literature, along with performance evaluation on nano-wireless use. This article aims to provide an algorithm to reconnect groups of microrobots, along with a novel movement model for microrobotics ensembles introduced to enforce more realistic simulations. Special emphasis is given on the need of novel movement algorithms for swarms of microrobots.

1. Introduction

The frontier of robotics technology is being pushed forward by the growing development of Micro Electro Mechanical System (MEMS). MEMS are miniature low power devices, usually batch produced, that entrust sensing and acting capabilities in their environment. The possibility of MEMS mass production due to their small size makes them a very affordable technology and encourages the research and their application for practical purposes in everyday life. Dimensions varies from well below one micron on the lower end of the dimensional spectrum to several millimeters. From the very basic to the most advanced applications sees a massive deployment of nodes which highly affect the ensemble shape. An efficient topology design is therefore needed to ensure an optimal communication among nodes. Due to their small size, MEMS dispose of a very limited power supply, making it difficult to implement a precise and reliable movement. Even though the movement power consumption has been lowered, communication and computation requirements still represent a challenge in terms of energy. The optimization process in terms of number of movements and message exchanges in MEMS is therefore crucial in order to save energy. The self-reconfiguring ability of MEMS modular robots is very complex to control as a distributed coordination of large numbers of identical modules connected in time-varying ways has to be performed. The amount of exchanged information and the number of displacements define the communication and energy complexity of a distributed algorithm. When information exchange involves close neighbors, the complexity is low and the resulting distributed self-reconfiguration scales along with the network size. With a moderate complexity in message exchanging, execution time, number of movements and memory usage, finding an optimal configuration is still challenging. The optimization of the network logical topology has to be performed through rearrangement of the physical topology. Several microrobotics projects are developed under these assumptions. Harvard University Kilobots project main purpose is to achieve a scalable and self-assembly design for microrobotics [1]. Kilobots are tiny modular microrobots, assembled in swarms of thousands, that

* Corresponding author at: Sapienza - University of Rome, DIET, Department of Information Engineering, Electronics and Telecommunications, via Eudossiana, 18, 00184, Rome, Italy.

E-mail addresses: ferranti@ece.neu.edu, ludovico.ferranti@uniroma1.it (L. Ferranti), francesca.cuomo@uniroma1.it (F. Cuomo).

URLs: http://www.ece.neu.edu/wineslab/Ludovico.php (L. Ferranti), http://netlab.uniroma1.it/netgroup/users/francesca-cuomo (F. Cuomo).

http://dx.doi.org/10.1016/j.nancom.2017.01.007
1878-7789/

move and reconfigure themselves through vibrational motors on a lucid plane, which is used to reflect infrared rays as a mean of communication. Another remarkable instance of microrobotics is Claytronics by Carnegie Mellon University and Intel [2]. These modular microrobots projects are all characterized by short range communication. Microrobots need to be close one to another to empower a reliable communication. The use of wireless communications facilitates topology formation and maintenance as well as lower the energy consumption of the system. In this paper we focus on Claytronics project as an instance of microrobotics and we take advantage of nano-wireless communications framework to overtake the limitations caused by short range communication. Nano-wireless technology in Claytronics has been thoroughly investigated in recent years, yet advanced microrobotics algorithm needs to be implemented and optimized. The novel contribution of this paper is an algorithm that distributedly reconnects groups of microrobots, proposing a safety routine that avoid Claytronics to have a single point of failure. Despite the techniques are based on Claytronics, the algorithm can be easily extended to other microrobotics projects to develop a common framework. The remainder of this paper is organized as follows: Section 2 will present two of the most remarkable microrobotics projects and describe the main problems related to microrobots deployment, Section 3 will present the Nano-wireless Communication operating in the Terahertz Band, Section 4 will introduce a new movement model for catoms, Section 5 will present the Rejoin algorithm used to connect networks of microrobots, Section 6 will describe the simulator used to test the algorithm, Section 7 will underline the differences between Rejoin and previously implemented algorithms in the field and Section 8 will draw conclusions and future work.

2. Problem description and motivation

Microrobotics environments are usually characterized by a huge number of modular robotics. Two of the most remarkable microrobotics projects are presented below.

2.1. Kilobots

In [1] swarms of Kilobots are used to investigate collective "artificial" intelligence" (e.g. synchronization, self-assembly, collective transport). A self-organizing swarm of one thousand individual robots can self-assemble into specific shapes based on simple behaviors performed as a whole just like billions of cells assembles into a living organism. To implement low cost locomotion, Kilobots use two sealed coin shaped vibration motors. Activating one motor generates centripetal forces which are converted to a forward force on the Kilobot where the motor is mounted on. Kilobots velocities are approximately 1 cm/s and they rotate approximately at 45 °/s. Each Kilobot is mounted with an infrared LED transmitter and infrared photodiode receiver, which enables an IR communication with neighboring robots, with a communication rate up to 30 kb/s with robots up to 10 cm away.

2.2. Claytronics

Claytronics is a project that combines nanoscale robotics and computer science to create individual nanometer-scale computers called *catoms*, which can interact with each other to form shapes. This idea is more broadly referred to as Programmable Matter. Other approaches to modular robotics used to create a conglomerate of tens or even hundreds of small autonomous robots which move in coordination to achieve a global effect which would not be possible to achieve by a single unit [2]. The main design goal of Claytronics is to scale to millions of micro-scale units. Each Claytronics MEMS micro-robot is equipped with structures called "features". Those features are used as attachments using electromagnetic or electrostatic force and as a direct communication mean. By actuating their features, catoms are able to move around their direct neighbors in the same way stepper motors navigate their surroundings. However, catoms maintain their ability to communicate. Different versions of catoms has been built and dimensions go from 8 m^3 (Giant Helium Catoms) to 4 cm (Planar Catoms) and to few millimeters (Millimeters Scale Catoms). Bigger catoms are used to evaluate forces interaction and physical properties. Technical parameters of catoms such as weight, power consumption and manufacturing methods are shown in [2]. Several catoms form an *ensemble*, which is a dense environment within a certain communication range. Most control algorithms would improve performances from broadcasting capability, speeding information diffusion process, allowing neighbor discovery and providing ability to communicate with non contiguous groups of catoms. Broadcasted messages can also be used as a form of synchronization, which is very useful when coordinating movements of groups of catoms. In Claytronics, energy is provided to each catom from an external power source, and if required can be routed from catom to catom. From the energy point of view, there is a substantial difference from sensor networks because sensors networks have more communications constraints due to the rate they gain energy. In sensor networks, nodes usually rely on battery or have to harvest energy from their environment. In catoms, energy is more often a matter of topology. In [3] a complexity analysis about message exchange complexity is engaged. The distributed algorithms listed below have been implemented using the declarative language Meld [4] and evaluated using the simulator DPRSim by Carnegie Mellon University. Most of those algorithms provide a shape shift from a chain of microrobots (worst case scenario) into a square. Indeed, the chain form represents the worst physical topology for many distributed algorithms in terms of fault tolerance, propagation procedures and convergence. Also, a chain of microrobots represents the worst case for message broadcasting complexity (that is $O(n)$). The redeployment into a square organization (an $\sqrt{n} \times \sqrt{n}$ matrix) allows to obtain the best message broadcasting complexity with $O(n)$. Among several other versions in [5–8], two main algorithms are presented below as an example of different approaches to model complexity in message exchange.

PASC. The main distributed algorithm is Parallel Algorithm with Safe Connectivity (PASC) in [3]. It is used to move catoms from a chain to a square shape. The catom in the middle of the chain is the initiator and other catoms move around it in a parallel fashion.

PAUC. The second distributed algorithm is Parallel Algorithm with Unsafe Connectivity (PAUC) in [9]. With respect to PASC, PAUC ensures Snap-Connectivity only at the end of the procedure, under the assumption that a single catom could freely move. This is not true in general but shows the scenario in which some catoms could be parted from the whole ensemble. Other versions of PAUC is presented in [10]. All the algorithms previously presented have been designed to work on touch communication through features and do not really take advantage of a wireless communication paradigm. In fact, as shown in [11,12] it is demonstrated that using nano-wireless communications dramatically enhances the performance of the ensemble communication. For example, isolated catoms that cannot reach the main ensemble can be reintegrated using the catom walker algorithm in [13], helping to build a strong fault tolerance.

Wireless communication. The adoption of wireless communications for catoms has been broadly investigated in [11,14]. Results show that wireless communications in a network enhances the way catoms are communicate, improving the communication range and allowing easy way of broadcasting information. Using only short range communication, microrobots are structurally unable to reach another unit that is not close to the ensemble and distributed algorithm can hardly achieve convergence when units are moving. Consequently a constant restructuring process on the entire logical topology runs all the time. The main advancement about the integration of a wireless network inside Claytronics project is to overcome the limits of the wired communications. As previously mentioned, the radio channel used for nano-wireless is very specific. A transmitted signal suffers not only from a continuous attenuation with the increasing distance, but also by the channel noise caused by itself and other concurrent transmissions. Results in [11,12] shows that having a high number of concurrent transmissions (from thousands to hundred of thousands), performances can be optimized performing a reduction in the weight of the coding. This can be done increasing the number of "0" symbols (i.e. no sent pulse) against the "1" symbols (a sent pulse). Even though the size of the message with this particular coding scheme increases, the ability to correctly receive the message exceeds its drawbacks. Moreover, in Claytronics and programmable matter in general, due to its application and control layers, there will be large variations over time in communication volume. It can happen that almost no communications are required when the network reaches a stable state or it can happen that only a few catoms may want to transmit. On the contrary, certain events or commands can suddenly trigger a large need for communications. Programmable matter requirements are very different from those of wireless sensors networks, where there is a much more constant network load over time.

3. Nano-wireless communication

To empower nano-wireless communication the antenna have to be reduced down to a few hundreds of nanometers. The main consequence is that the process requires the use of extremely high operating frequencies, which could compromise the feasibility of electromagnetic wireless communications among nano-devices. However, using graphene to fabricate nano-antennas can overcome this limitation. Carbon nano-tubes and graphene nano-ribbons can enable the communications at that frequencies. In [15], a remarkable comparison between nano-patch antennas based on graphene nano ribbons and nano-dipole antennas based on carbon nano-tubes is given, showing how a graphene nano-antenna 1 μm long can efficiently radiate EM waves in the Terahertz Band (0.1–10.0 THz). Nano-wireless communication is therefore a new paradigm being developed in the field of nanotechnology. Building nanonetworks means building nano-sized communicating devices that are able to perform simple tasks at the nanoscale. This technology is feasible for environments with very high node densities, in the order of hundreds of nanosensors per square millimeter. A wide range of applications such as advanced health monitoring systems or high-performance distributed nano-computing architectures are made possible thanks to this new technology. Key features of novel plasmonic nano-transceivers and nanoantennas, which operate in the Terahertz Band, bring along the need to develop tailored communication schemes for nanonetworks [16]. Nano-wireless main challenge is molecular absorption. In [17] the communication bandwidth is selected with respect to molecular absorption level, leading to better performance in terms of communication range and capacity. Results show that for a nominal power of 26.5 nW (100 femtosecond-long pulse with 0.1 aJ energy) the communication range can be up to 15 m providing satisfactory capacity for transactional applications in the range of 4–11 Mbps. This range is reachable only operating in the lower end of the first transparency window 0.1–0.54 THz. Moving up in the spectrum to the next transparency window, the free-space propagation loss dominates the channel characteristics making long communication distances not possible. Using the whole window bandwidth one can get up to 2 Tbps at 0.01 m. In [18] a modulation and channel access scheme for nanonetworks in the Terahertz Band is developed. The technique is based on the transmission of one hundred-femtosecond-long pulses by following an asymmetric On–Off Keying modulation Spread in Time (TS-OOK). TS-OOK technique uses very short pulses in the range of femtoseconds where each pulse transmits a "1" and the absence of pulse transmits a "0". To ensure an efficient medium sharing and reduce interference and noise, each pulse must be transmitted using a period of time much bigger than the duration of the pulse itself. Of course the transmission of "0" has to be preferred to "1" as it does not create any perturbation of the channel and it is also easier to detect. Using a low-weight channel coding can reduce the influence of interference between concurrent transmissions leading to better aggregated bandwidth. The key parameter β represents time between two pulses. The bigger the β, the bigger the multiplexing capability [19]. When pulses are very short (around 100 fs), the propagation delay (3 ns/m) can be greater than the duration of a pulse and collisions can easily occur at receiver side even if symbols were sent at different times. In [20] an energy model for nanomachines is introduced. In this sense, catoms can be seen as a particular instance of nanomachine. Despite the different size and models of catoms, for simulation's sake the nanoantenna position has been intuitively set at the center of each catoms. More details on nanonetworks and their energy efficient deployment can be found in [21]. A visual representation of a catom can be seen in Fig. 1.

Fig. 1. A visual representation of a catom with a nanoantenna in its center.

In [22] two scenarios are considered. In the first one we have a catom walker that needs to be integrated in the main ensemble. Using only short range communication, such task would have not been possible. Thanks to nano-wireless communication a handshake protocol can be implemented so as to allow the catom walker to move toward the main ensemble.

The main contribution of this work are: a new movement model that better abstracts the physical behavior of catoms interaction during the movement; the integration of this movement model into the existing simulation tools; an algorithm that allows groups of catom to join into a single ensemble using nano-wireless communication.

4. Catoms movement model

The simulation tool that has been used is Vouivre Standalone [11]. In this simulator, the movement of catoms is just abstracted and more focus is given to network performance of message exchanging using Nano-wireless communication. Catoms movement is represented as a delay and is not physically mapped in Vouivre. This means that when the movement primitive is called, Vouivre changes the position of a node from the origin to the destination after a delay based on the velocity of the movement.

Topology highly affects movement in Claytronics. It is pretty straightforward that a cluster of twenty catoms is slower than a single catom walker, which is composed by only three catoms. Moreover there are several topologies that are disadvantageous concerning both the movement and the communication. From [3]

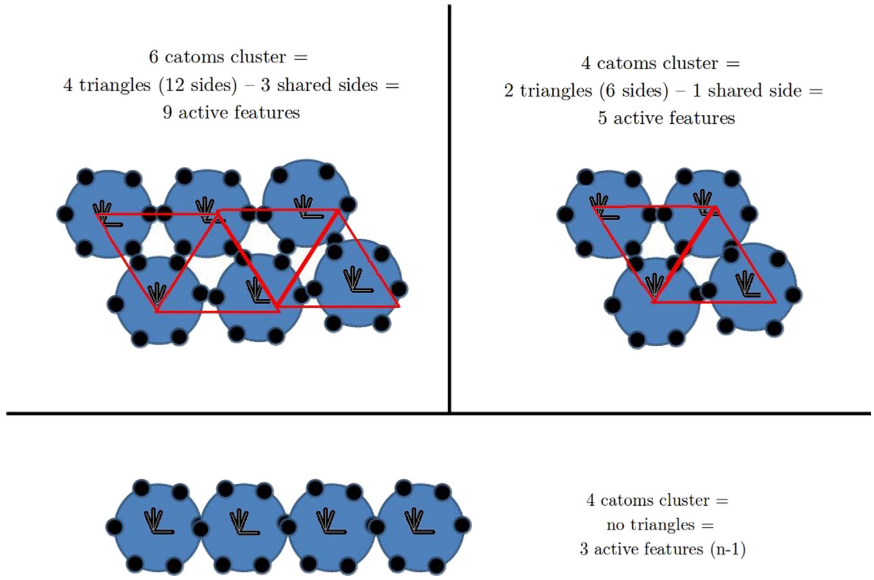


Fig. 2. Geometry approach to catom connectivity.

we know that the worst case scenario is represented by a straight line of catoms. To take into account how the topology of a cluster of catoms can affect the movement, we evaluate a connectivity factor based on the number of features that are active in one cluster. The global number of features active in one cluster represents the number of connections between catoms. The higher the number, the higher the connectivity.

For instance in a cluster with n catoms, the worst case scenario (a line topology) has $n-1$ active features. As the number of features increases (catoms are closely connected) we can assume that the topology gets better and for the whole cluster to make a progress in the motion, less catoms have to be overcame. The best scenario would be where there are no holes in the cluster and several "catom walkers" are interconnected.

To study the incremental law that relates the number of features with the number of catoms, we analyzed the number of catom walkers inside a cluster minus the number of sharing features with other catom walkers. From a geometry point of view, we can represent a catom walker as a triangle of features. Every time a triangle shares a side with another one, it has to be subtracted from the sum of the sides. The global number of features is given by the total number of sides of the triangles with the shared sides subtracted. Fig. 2 summarizes the presented concept.

As we can see, the worst case scenario of connectivity is an $n-1$ number of features active. A number of n features is still good, but a number greater than n represents the best case.

To model the topology factor based on the number of global features in a cluster, we introduce in the model the factor α. The values of α are as follows:

having φ number of active features in a cluster:

$$\alpha = \begin{cases} 1 & \varphi \geq n+1 \\ 0.8 & \varphi = n \\ 0.5 & \varphi = n-1. \end{cases}$$

Relying on the assumption that a bigger cluster is slower with respect to a smaller one, which justifies the algorithmic choice of making a smaller cluster move toward a big one, we can state that the movement delay of a catom cluster is a function of the number of catoms in the cluster and topology.

The number of catoms needed to cover a certain distance D can be expressed as follows:

$$n_d = \left\lceil \frac{D}{2r} \right\rceil$$

with r being the radius of a catom. In case of D not being multiple of n_d, a configuration in which two catoms of different clusters are close enough to attach their features can be found. An additional movement is required to cover a non-multiple configuration so we take the upper integer part with the ceiling function. The number of catoms n_d representing the distance between two clusters can be seen as the number of movements that a cluster needs to carry out to reach the other cluster.

In the worst case scenario, to overcome another catom through feature movement, 3 'ticks' are needed, which means that three features need to be activated and deactivated to progress in the movement.

The velocity of a catom cluster v_{cl} is therefore defined as follows:

$$v_{cl} = \frac{\alpha \cdot 3 \cdot v_f}{n}$$

where v_f is the velocity of feature activation and deactivation that allows a catom to move and n the number of catoms in the cluster. We can express this velocity as

$$v_f = r\omega$$

where r is the radius of a catom and ω it's angular velocity.

At each feature activation and deactivation, a catom covers the distance of 60°, therefore:

$$\omega \cdot t_\varphi = \frac{\pi}{3}$$

where t_φ is the delay introduced from feature activation.

The delay in the movement is given by:

$$t = \frac{D}{v_{cl}}$$

plugging in the equations and expressing them in terms of n_d as:

$$t = \frac{2n \cdot t_\varphi \cdot n_d}{\pi\alpha}.$$

It has be assumed the delay in covering the distance of one catom to be 3 times the delay of feature activation. The parameter α mitigates or strengthen the delay, depending on the topology factor.

Fig. 3 shows the distance between two catom clusters in terms of movements to be performed.

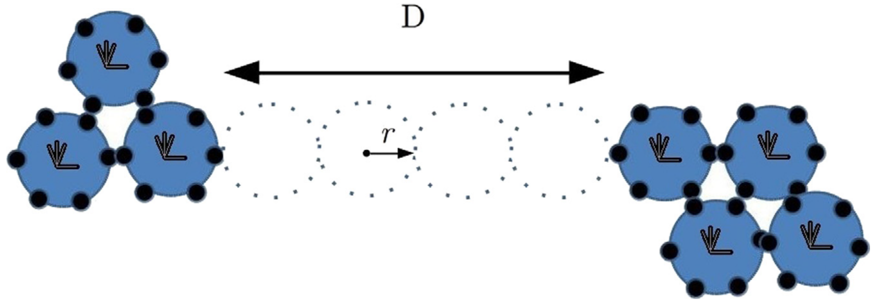


Fig. 3. The model of distance: dotted catoms are the number of catoms to fit in the distance D.

For the simulations that have been carried out, the following model of movement has been used, with several combinations of connectivity factors and clusters in order to stress the concept of intertwined movement and network performance.

In terms of energy consumption we use the model presented in [20]. In particular, the energy used to send or receive a message is

$$E_{packet-tx} = N_{bits} W E_{pul-tx}$$
$$E_{packet-rx} = N_{bits} E_{pul-rx}$$

where E_{pul-tx} and E_{pul-rx} are the energy consumed in the transmission and the energy consumed for the reception of a pulse, W refers to the coding weight, i.e. the rate between transmitting a pulse ("1") or being silent ("0"). By being silent, a transmitter can save energy but not the receiver.

5. Rejoin algorithm

Rejoin algorithm is theoretically introduced in [22]. Given that microrobots are able to estimate their position based on the analysis of wireless signal or sensors, several simulations and demos are executed in controlled environments. In [13] catoms are used for map building purposes, merging the sensors informations of all the catom walkers exploring the environment. In particular, the work in [13] is missing a reconnection strategy once the map is built and catoms keep moving freely in the environment without any particular task. Rejoin algorithm can therefore be executed after catom walkers have accomplished the map building task and reconnect all together in a central ensemble having the information about position and map. The proposed Rejoin algorithm is executed in a distributed fashion and several particular scenarios have to be analyzed. For instance, the scenario in which a cluster sends a PERMISSION message to a smaller cluster to be joined and then receives an HELLO by a bigger cluster. In this particular event, the cluster which asked to join will not find the destination cluster in the position got in PERMISSION message as it has started to join the bigger cluster. This scenario does not really affect performance or convergence as another HELLO messages can be sent upon intervals and the smaller cluster will join the newly big cluster. Fig. 4 shows the presented situation.

An unwanted scenario can be the situation in which a small cluster is stuck between two clusters of the same dimension, moving catoms are prevented to reply to HELLO messages and therefore the situation of a cluster blocked in loop between two cluster avoided. While converging, several clusters that join a big ensemble have to apply collision avoidance policies. Yet collision between two catom groups is not as bad as it could seem in this context. In fact a well-handled impact, can enable two clusters join. But it has to be taken into account that performance can be affected. While a big cluster is joining another, an impact and successively a join with a moving smaller group would make the big cluster even bigger and slow down its converging velocity. In Fig. 5 a collision situation properly shows this scenario.

Proper communication routines during movement can be used for ranging purposes to enable collision avoidance factors. Rejoin algorithm prevents Claytronics to have a single point of failure and to adopt fault tolerance policies. Routing protocols based on nano-wireless communication will result in a wider span of tasks that can be accomplished through catoms cooperations and pave the way to advanced microrobotics applications.

Algorithm 1 Rejoin Algorithm

```
procedure REJOIN(c_i, c_j)
    while true do
        sendMsg(helloMsg(a_i, p_i, n_i, t_i))
        checkMailbox()
        if checkMailbox() = newMsg() then
            if newMsg()=helloMsg(a_j, p_j, n_j, t_j) then
                if n_j ≥ n_i then
                    sendMsg(reqMsg(a_i, p_i, n_i, t_i));
                end if
                if newMsg()=reqMsg(a_j, p_j, n_j, t_j) then
                    sendMsg(perMsg(ai, pi, ni, ti))
                end if
                if newMsg()=perMsg(a_j, p_j, n_j, t_j) then
                    moveToPosition(p_j)
                    connectFeatures()
                end if
                if newMsg()=helpMsg(a_j, p_j, n_j, t_j) then
                    rescuer ←selectClosestCatom(p_j)
                    enableRouting(rescuer)
                    sendCatomWalker(rescuer, p_j)
                end if
            end if
        end if
        n_i ← checkConnectivity()
        if (n_i ≤ 2) then
            sendMessage(helpMessage(a_i, p_i, n_i, t_i));
        end if
    end while
end procedure
```

To approach this problem, the Rejoin algorithm is presented. Having two catom routers c_i and c_j, and for each of them addresses a_i, a_j are known, the position p_i, p_j and the number of catoms in their cluster n_i, n_j. At each iteration, a catom router sends a message with timestamp t_i, t_j to scan the environment for other clusters and then checks whether a new message has been received. Several kind of messages can be exchanged: *helloMsg* (a_j, p_j, n_j, t_j) is the basic message sent by catoms to notify their presence to other catoms in the surroundings. *reqMsg* (a_j, p_j, n_j, t_j) is sent by a router which has received a *helloMsg* by a router that belongs to a bigger cluster. *perMsg* (a_j, p_j, n_j, t_j) is a permission to join sent when a *reqMsg* is received. When a router receives a *perMsg* it starts to move toward the position of the bigger cluster. The functions *moveToPosition* (p_j) and *connectFeature()* are primitives to connect the two clusters through their features. At each iteration a catom router verifies its connectivity and updates its number

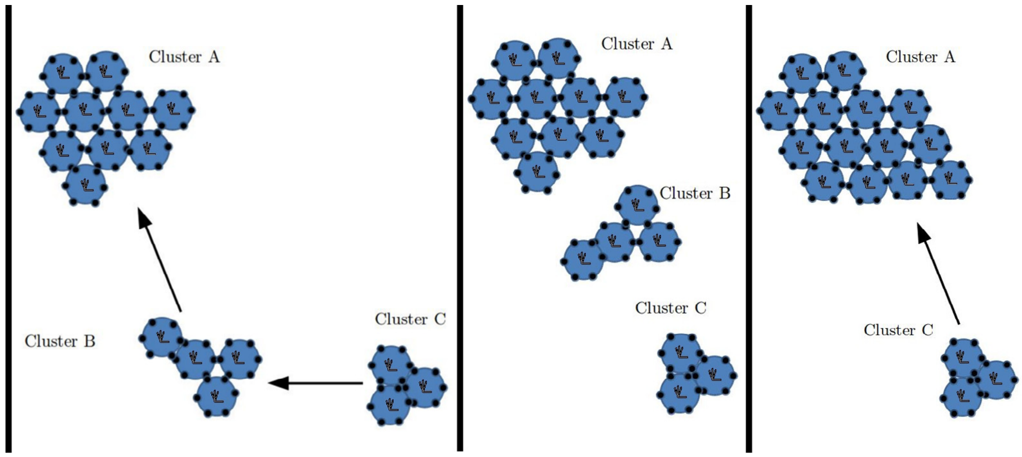


Fig. 4. A catom cluster chasing another. With a further iteration of Rejoin the three clusters would converge to the same cluster altogether.

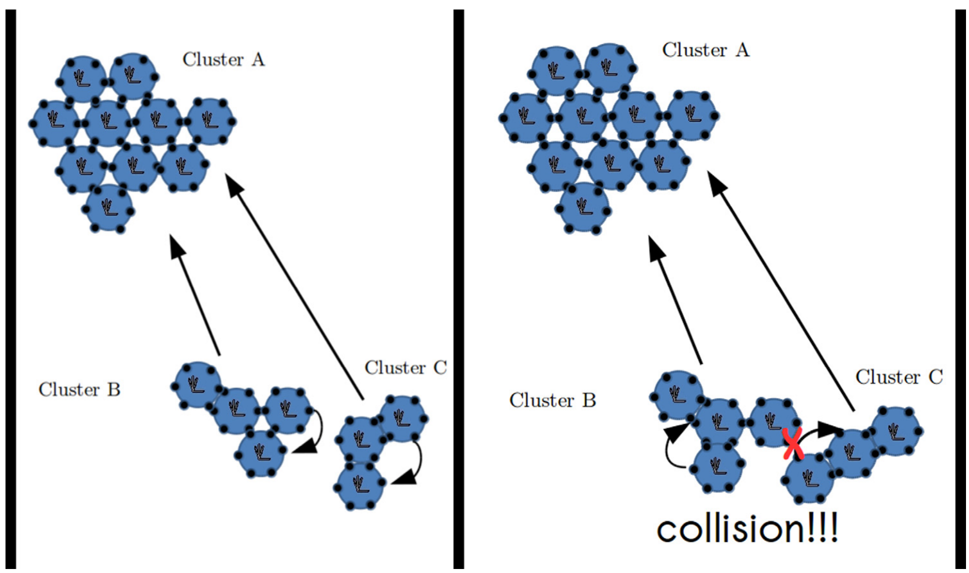


Fig. 5. Collisions are possible in Claytronics. Using sensing capabilities, collision can be handled to join two groups of catoms.

of catoms connected through features. If the number of catoms is smaller than 3, *helpMsg* is sent. When a router receives *helpMsg*, it designates a catom walker and enables the closest catom *rescuer* to use nano-wireless communication. Then the *rescuer* catom will be sent to join the faulty catom cluster.

6. Simulations

Claytronics project had several simulations tools in his history. The main simulation environment for catoms interaction is DPRSim [11]. The design of a new Claytronics application is written in C++ yet a big part of the Claytronics project relies in the Meld language [4]. To better simulate radio channel and concurrent access, a new simulation tool called Vouivre has been developed to work together with DPRSim in [12]. Vouivre is discrete-event based network simulator which includes a module for nanonetworks simulation and is developed in order to cope with the complexity and realism of a network. For this work the Standalone version of Vouivre was used, implementing the movement model presented. Different scenarios were simulated with various catom clusters topologies and results are subsequently reported.

In the first scenario we have 3 catoms cluster initially positioned at equal distance: Cluster A: composed by 3 catoms, which sends an helloMsg() at t1; Cluster B: composed by 5 catoms, which sends an helloMsg() at t2; Cluster C: composed by 8 catoms, which sends an helloMsg() at t3;

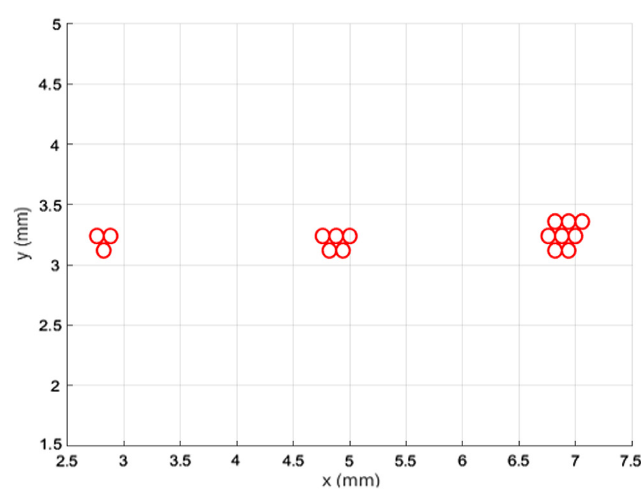


Fig. 6. Scenario 1: Three catom clusters at equal distance.

Fig. 6 represents the simulated scenario. Using Rejoin algorithm routine, the results produced are the following:

At $t1$ Cluster A broadcasts a HELLO, but no cluster replies, as their number of catoms is higher than Cluster A and moving toward it would result disadvantageous. At $t2$ Cluster A receives Cluster B HELLO and, as a smaller number of catoms are attached to it, sends a REQUEST message, requesting to join Cluster B. After an interval a PERMISSION message is received and Cluster A starts moving. For convergence purpose, Cluster A nanoantenna was left switched on. At $t3$ Cluster A receives Cluster C HELLO message and sends a REQUEST to join Cluster C. A PERMISSION is then received and Cluster A, starting from the same position of Cluster B, moves toward Cluster C.

From Cluster B point of view, at $t1$ an HELLO message from Cluster A is received, yet Cluster B is bigger and does not move toward it.

At $t2$ Cluster B executes Rejoin and broadcasts an HELLO message. Since Cluster A wants to join, it sends a PERMISSION message to Cluster A, that later joins. At $t3$ Cluster C sends an HELLO and as Cluster B is smaller than Cluster C, sends a request to Cluster C. As a PERMISSION message is received from Cluster C the movement is triggered.

Finally, at $t1$ Cluster C does not respond to Cluster A and having the biggest number of catoms attached, it does not join Cluster B at $t2$ either.

Please note that HELLO, REQUEST and PERMISSION from other cluster are received,, as messages are generally broadcasted, but not took into account.

At $t3$ broadcasts an HELLO and receives a REQUEST from both Cluster A and B, then a permission is sent to Cluster A and Cluster C.

The position convergence of this scenario is proven at the end of the computation, when the position of each cluster is the same. Fig. 10 shows the information of position x and y of the three catom routers.

Using the previously presented model, a careful and realistic simulation has been carried out.

The following values were used to accurately perform communication taking into account the movement delay:

$$D = 2 \text{ mm}$$

$$v_t = 1 \text{ cm/s} = 0.01 \text{ mm/ms}$$

$$\alpha = 1$$

$$r = 0.06 \text{ mm}.$$

From [20], we assign to each pulse $E_{pul-tx} = 1$ pJ and consider that E_{pul-rx} is approximately 10 times lower than E_{pul-tx}. On average, the weight representing the balance between "1"s and "0"s is $W = 0.5$. Rejoin algorithm messages are based on typical sensor networks packets. The packet length is 116 bits: 2 bytes for frame control, 1 byte for sequence check t, 2 bytes for destination address, 2 bytes for source address, 2 bytes for the number of catoms connected n, 3 bytes for the position p_i, 4 bits to differentiate between HELLO, REQUEST, PERMISSION and HELP messages and finally 2 bytes for frame sequence check.

Therefore, for each message sent we have

$$E_{packet-tx} = 58 \text{ pJ}$$

$$E_{packet-rx} = 11 \text{ pJ}.$$

With an optimal connectivity factor, catoms are not slowed down and therefore their velocity is the highest possible. Figs. 7–9 show the message exchange, taking into account the movement delay. As for the previous scenario, Cluster A broadcasts an HELLO to look for smaller groups. After 100 ns Cluster B broadcasts an HELLO and Cluster A sends a REQUEST to Cluster B. Please note that in the new plots, the simulation scale is way bigger and the HELLOs sent by Cluster A and B looks very close one to another even though they are sent after a very big simulation time. Once Cluster A has received its PERMISSION, the movement is triggered and a delay of 600 ms is introduced. Once Cluster A joins Cluster B, an HELLO is broadcasted by Cluster C. Acting like a new 8 catoms cluster, Cluster A + B antennas send a REQUEST to Cluster C, that replies sending two PERMISSION. Note that this time, to cover the same distance, Cluster A + B need 1600 ms to join Cluster C.

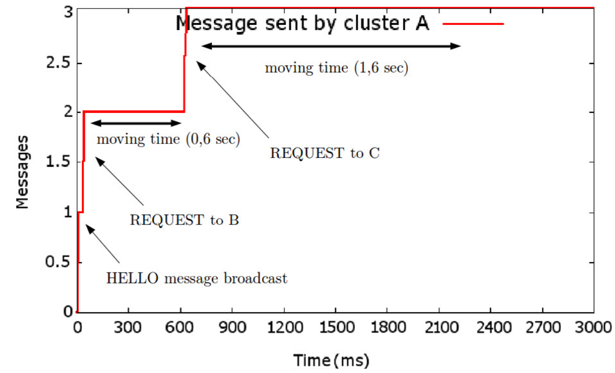


Fig. 7. Cluster A message exchange.

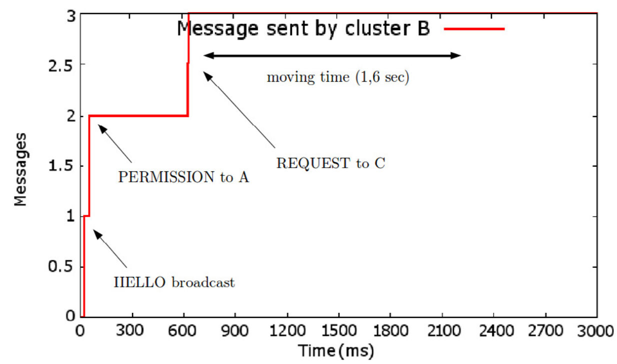


Fig. 8. Cluster B message exchange.

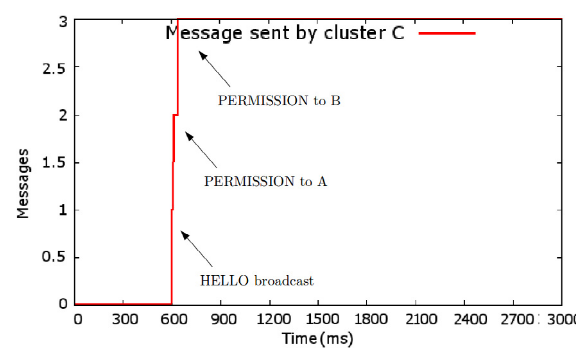


Fig. 9. Cluster C message exchange.

Further simulations with more catoms groups were carried out. A second scenario with a central group of 7 catoms, surrounded by 16 catom walkers has been investigated. Catom walkers were positioned at a distance of 2 mm from the central ensemble and as they maximize their connectivity, they have $\alpha = 1$. Such scenario is presented in Fig. 11. In this case, the main cluster S acts like a sink and broadcasts an HELLO. Fig. 12 shows the HELLO message and the following 16 PERMISSION. Again, moving delay is 600 ms. Final convergence is shown in Fig. 13.

Last simulation were carried out with a grid of different dimension and connectivity catoms. Fig. 14 shows the third implemented scenario. Each cluster is 2 mm distant one to another. First node A, having a bad connectivity and being the slower cluster, sends an HELLO, receiving a REQUEST from D. Note that the

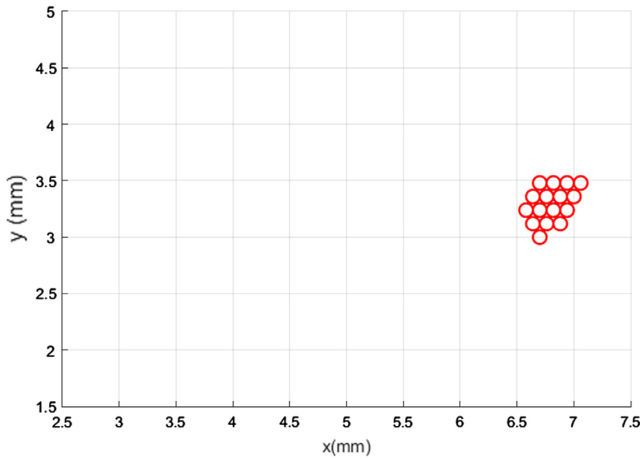


Fig. 10. Final position of the Clusters A + B + C.

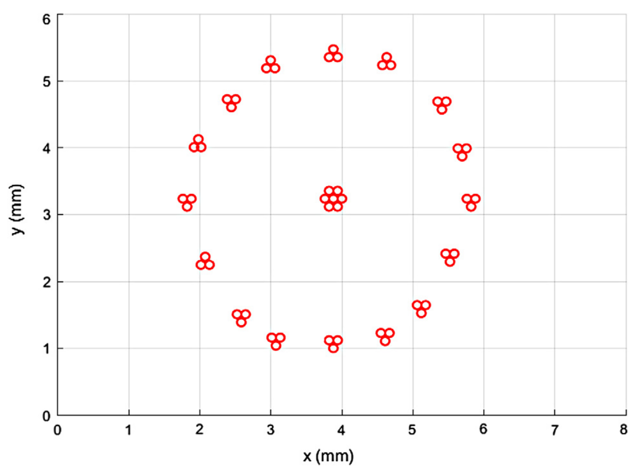


Fig. 11. Second simulated scenario: 16 walkers at 2 mm distance from a big ensemble.

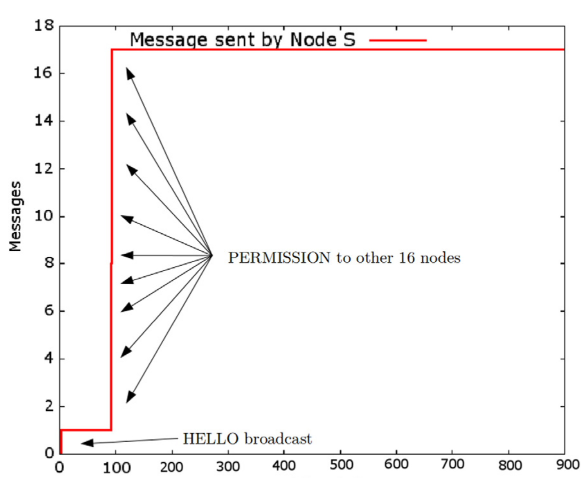


Fig. 12. Message exchange of main ensemble.

connectivity factor of D is better to the A one. After a moving time of approximately 2 s, cluster A + D receives HELLO from cluster B, now being the smallest cluster in the group. No clusters move and cluster C broadcasts an HELLO and all the smaller groups move toward it. To move, all the smaller clusters take up to 5 s. Node C proceeds to broadcast an HELLO that reaches nodes G and H that start their movements, finally reaching the overall cluster. The

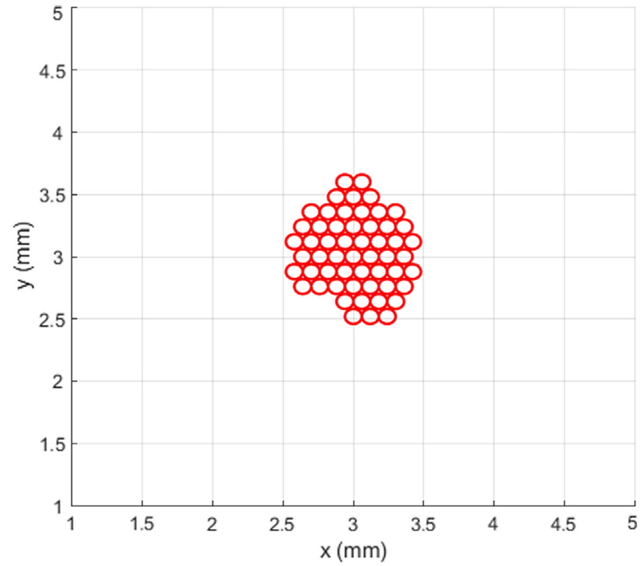


Fig. 13. Second simulated scenario: final convergence.

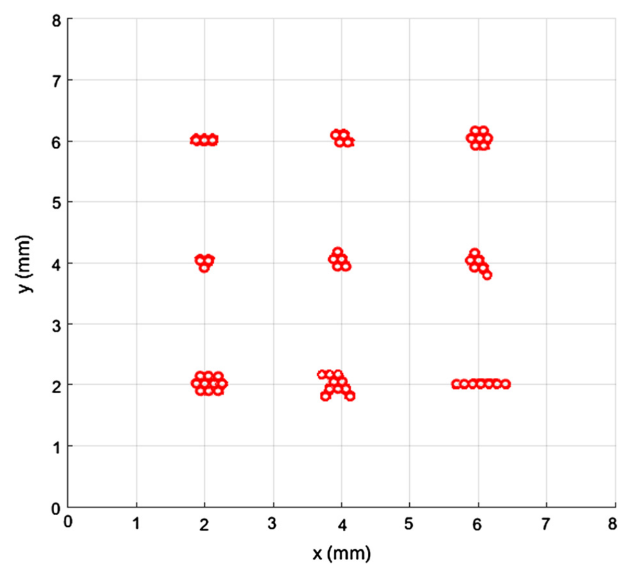


Fig. 14. Third simulated scenario: different clusters with different connectivity factors.

message exchange that best capture this simulation is the one of node D, shown in Fig. 15. Note that when G + H broadcast their REQUEST, node D antenna is enabled and automatically respond with two PERMISSION messages. Final convergence is shown in Fig. 16.

7. Robot connectivity algorithms in literature

The proposed Rejoin algorithm is based on the assumption that the connectivity among all the microrobots is lost and aims to restore it. Even though such algorithms are not fully exploited in literature, it is interesting to see that similar principles and approaches are applied in several fields. The design of robust algorithms for mobile agents assuming a reliable communication is an open challenge in both the fields of robotics and communication due to the distributed nature of computation. In mobile robotics networks this goal is even harder because of the connectivity constraints. Existing distributed algorithms provide coordination but typically assume connectivity is ensured by other means. Another overview of a distributed approach is given in [23] where two

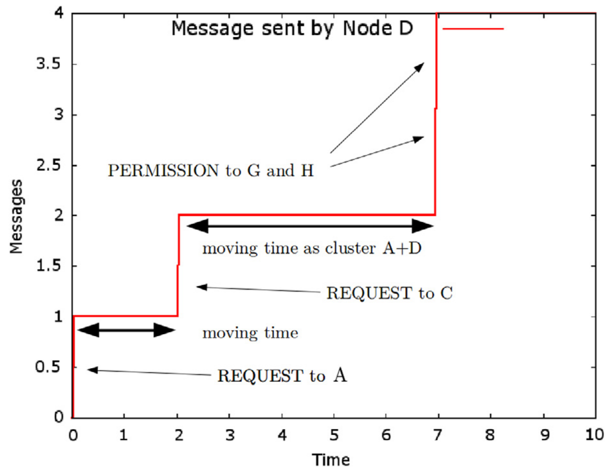


Fig. 15. Message exchange of Node D.

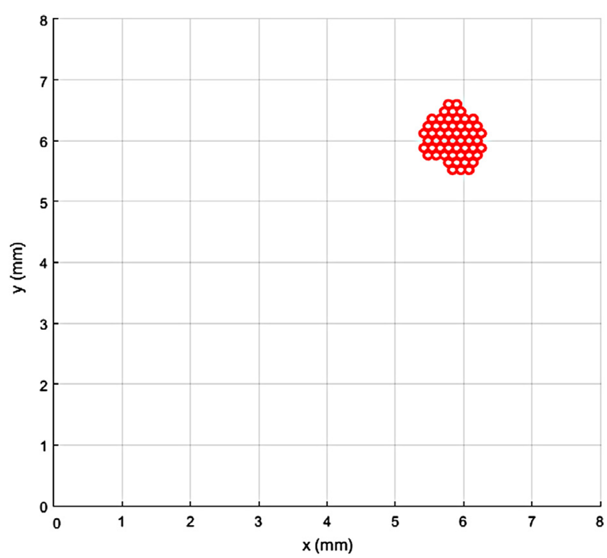


Fig. 16. Third simulated scenario: final convergence.

proximity-limited communication models are took under analysis: the disk-based communication model and the hysteresis-based communication model. With those model, connectivity concepts coming from control theory are optimized under network communication metrics of Bit Error Rate (BER) and Signal to Interference plus Noise Ratio (SINR). Literature already faced the problem of connectivity among mobile agents network, but with a different approach. In particular, a connectivity service which contains a motion planner for agents in a mobile ad hoc network (MANET) and at the same time is able to refine any plan to preserve connectivity is investigated in [24]. Again, the connectivity problem is largely covered in literature in Wireless Sensor Networks. In the particular case of WSNs, failure due to several reasons in sensor nodes is an unwanted scenario. As a result the network may be split into disconnected partitions, deteriorating or nullifying the effectiveness of the network. A repairing partitions protocol to restore the network to its full potential is presented in [25]. The Rejoin algorithm proposed in this work will be discussed in details in the next section, however a comparison to underline the main differences and to analyze it under a robot connectivity framework has to be done. Rejoin algorithm works basically on the opposite of the algorithm developed in [24]. In [24] the input is a connected graph of nodes, the output is a safe path planning that does not break connectivity. In the Rejoin algorithm presented it is the opposite. The

Table 1
Message exchange and energy consumption.

	Scenario A	Scenario B	Scenario C
Total messages sent	9	33	13
Average message sent by each catom router	3	1.94	1.44
Main Ensemble Energy consumption	207 pJ	1173 pJ	276 pJ
Total Energy	621 pJ	2277 pJ	897 pJ

input is a broken connectivity, where connectivity is intended as catoms connected through features, and the output is a movement to restore a single connected ensemble. The approach presented in [23] is highly based on control theory concepts such as Laplacian and Adjacency matrix and also model the communication area of each robots, dividing it into partitions. Moreover it deals with collision avoidance and agent repulsion. The Rejoin algorithm main goal is attraction, which is based in the position informations of the catoms cluster. Therefore a communication partitions with repulsion areas would not really be applicable for rejoining purposes, despite the fact that physical collisions between clusters of catoms executing the algorithm have to be avoided. Concept such as adjacent catoms and logical topology are already presented in [3,5,9] but are only based on feature communications and need further investigation under the wireless communication perspective. From the Wireless Sensor Network point of view, most of the principles used in [25] are similar to the ones that drove the creation of Rejoin algorithm. In fact, in [25] a mobile robotic node navigate until the edge of the safe partition in order to be the closest to the isolated partition. In the Rejoin algorithm, when a faulty catom walker sends an help request, is the closest catom walker that is designated to start the rescue process. Moreover, the final aim of the algorithm in [25] is to restore a single connected network, just like the goal of Rejoin. Indeed there are several differences, just like the differentiation of roles of mobile and fixed node and the presence of a base station, but the goal of reconnecting isolated nodes to the main network is a common principle. Table 1 resumes the energy consumed in all the three scenarios. We can see how the energy consumption of the algorithm is strictly connected to the connectivity factor of the catom clusters. Scenario B, which simulates a high number of catom walkers, has a significant higher energy consumption with respect to Scenario C despite simulating the same number of catoms.

8. Conclusions

This paper presented the use of nano-wireless communication for microrobotics within Claytronics project. To improve scalability and to recover possible faults, nano-wireless communications using femtosecond-long pulses by following an asymmetric TS-OOK modulation were discussed. An algorithm to rejoin different catom clusters is presented. Simulations using a novel movement model to accurately compute communication and movement delay have been carried out and convergence of the algorithm has been shown.

Results show that the algorithm converges with an essential message exchange. Notwithstanding the enormous difference between the communication speed and the movement speed, using nano-wireless communication is a winning strategy, especially in large scale environments.

As previously remarked, the simulator used is a network simulator that abstracts movements. The good accuracy in message exchange simulation is not comparable with the movement accuracy. Essential robotics functions as collision avoidance and position tracking over time are implemented only in the outdated simulator DPRSim. Soon the new physical ODE simulator VisibleSim will embed Vouivre Library and further research will be conducted on the optimization of this algorithm.

For instance, movement of large catoms clusters may result bulky and utterly slow. Splitting a large cluster in smaller catom walker groups, may be a winning strategy to optimize the movement. In this perspective, this algorithm may be the final step of a split-move-rejoin routine for big catom groups.

Future work will be dedicated not only to analyze the behavior and the scalability of such protocols in simulated large microrobots networks, but also to implement faster and richer simulation environments to better analyze the capabilities that microrobotics can offer.

References

[1] M. Rubenstein, C. Ahler, R. Nagpal, Kilobot: A low cost scalable robot system for collective behaviors, in: IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 3293–3298. http://dx.doi.org/10.1109/ICRA.2012.6224638.

[2] S.C. Goldstein, T.C. Mowry, Claytronics: A scalable basis for future robots, RoboSphere 2004.

[3] H. Lakhlef, H. Mabed, J. Bourgeois, Parallel self-reconfiguration for mems microrobots, in: EUROCON 2013, 2013, pp. 283–290. http://dx.doi.org/10.1109/EUROCON.2013.6624998.

[4] M.P.A. Rollman, S.C. Goldstein, P. Lee, T.C. Mowry, P. Pillai, Meld: A declarative approach to programming ensembles, in: IEEE International Conference on Intelligent Robots and Systems (IROS) http://dx.doi.org/10.1109/IROS.2007.4399480.

[5] H. Lakhlef, H. Mabed, J. Bourgeois, Distributed and efficient algorithm for self-reconfiguration of mems microrobots, in: 28th ACM Symposium On Applied Computing, SAC 2013. Coimbra, Portugal, 2013, pp. 1–6. http://dx.doi.org/10.1145/2480362.2480469.

[6] H. Lakhlef, H. Mabed, J. Bourgeois, Optimization of the logical topology for mobile mems networks, Netw. Comput. Appl. 62 (2014) 163–177. http://dx.doi.org/10.1016/j.jnca.2014.02.014.

[7] H. Lakhlef, H. Mabed, J. Bourgeois, An energy and memory-efficient distributed self-reconfiguration for modular sensor/robot networks, J. Supercomput. 69 (2014) 908–929. http://dx.doi.org/10.1007/s11227-014-1196-8.

[8] H. Lakhlef, H. Mabed, J. Bourgeois, S.C. Goldstein, Energy-aware parallel self-reconfiguration for chains microrobot networks, J. Parallel Distrib. Comput. 75 (2014) 67–80. http://dx.doi.org/10.1016/j.jpdc.2014.10.003.

[9] H. Lakhlef, H. Mabed, J. Bourgeois, Efficient parallel self-reconfiguration algorithm for mems microrobots, in: 22nd Euromicro Int. Conf. on Parallel, Distributed, and Network-Based Processing, PDP 2014, Torino, Italy, 2014, pp. 154–161. http://dx.doi.org/10.1109/PDP.2014.35.

[10] H. Lakhlef, H. Mabed, J. Bourgeois, Robust parallel redeployment algorithm for mems microrobots, in: IEEE 28th International Conference on Advanced Information Networking and Applications (AINA), 2014, pp. 1057–1064. http://dx.doi.org/10.1109/AINA.2014.128.

[11] N. Boillot, D. Dhoutaut, J. Bourgeois, Efficient simulation environment of wireless radio communications in mems modular robots, in: IEEE Int. Conf. on Internet of Things, Beijing, China, 2013, pp. 638–645. http://dx.doi.org/10.1109/GreenCom-iThings-CPSCom.2013.118.

[12] N. Boillot, D. Dhoutaut, J. Bourgeois, Using nano-wireless communications in micro-robots applications, in: 1st ACM Int. Conf. on Nanoscale Computing and Communication, NANOCOM 2014, Atlanta, Georgia, USA, 2014, pp. 1–9. http://dx.doi.org/10.1109/TCOMM.2014.033014.130403.

[13] N. Boillot, D. Dhoutaut, J. Bourgeois, Large scale mems robots cooperative map building based on realistic simulation of nano-wireless communications, Nano Commun. Netw. 6 (2) (2015) 51–73. http://dx.doi.org/10.1016/j.nancom.2015.01.004.

[14] J.M. Jornet, J.C. Pujol, J.S. Pareta, Phlame: A physical layer aware mac protocol for electromagnetic nanonetworks in the terahertz band, Nano Commun. Netw. 3 (1) (2012) 74–81. http://dx.doi.org/10.1016/j.nancom.2012.01.006.

[15] J. Jornet, I. Akyildiz, Graphene-based nano-antennas for electromagnetic nanocommunications in the terahertz band, in: 4th European Conference on Antennas and Propagation, EUCAP.

[16] J.M. Jornet, I.F. Akyildiz, Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the terahertz band, IEEE Trans. Wirel. Commun. 10 (2011) 3211–3221. http://dx.doi.org/10.1016/0003-4916(63)90068-X.

[17] P. Boronin, V. Petrov, D. Moltchanov, Y. Koucheryavy, J.M. Jornet, Capacity and throughput analysis of nanoscale machine communication through transparency windows in the terahertz band, Nano Commun. Netw. 5 (3) (2014) 72–82. http://dx.doi.org/10.1016/j.nancom.2014.06.001.

[18] J.M. Jornet, I.F. Akyildiz, Femtosecond-long pulse-based modulation for terahertz band communication in nanonetworks, IEEE Trans. Commun. 62 (2014) 1742–17541. http://dx.doi.org/10.1109/TCOMM.2014.033014.130403.

[19] N. Boillot, D. Dhoutaut, J. Bourgeois, Parameter study and characterization of wireless nanonetworks through simulation, in: 2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), 2014, pp. 43–47. http://dx.doi.org/110.1109/BlackSeaCom.2014.6849001.

[20] J.M. Jornet, A joint energy harvesting and consumption model for self-powered nano-devices in nanonetworks, 2012, pp. 6151–6156. http://dx.doi.org/10.1109/ICC.2012.6364968.

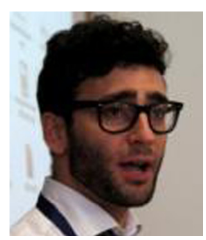
[21] J.M. Jornet, I.F. Akyildiz, Fundamentals of electromagnetic nanonetworks in the terahertz band, Found. Trends Netw. 7 (2–3) (2013) 77–233. http://dx.doi.org/10.1561/1300000045.

[22] L. Ferranti, F. Cuomo, Nanowireless for microrobotics: bridging the gap, in: 2nd ACM International Conference on Nanoscale Computing and Communication (NANOCOM), Boston, Massachusetts, http://dx.doi.org/10.1145/2800795.2800803.

[23] B. Ning, J. Jin, J. Zheng, Connectivity control and performance optimization in wireless robotic networks: Issues, approaches and a new framework, in: International Conference on Modelling, Identification and Control, Melbourne, Australia, 2014, pp. 142–148. http://dx.doi.org/10.1109/ICMIC.2014.7020742.

[24] A. Cornejo, F. Kuhn, R.L. Wild, N. Lynch, Keeping mobile robot swarm connected, in: Distributed Computing, in: Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 2009, pp. 496–511. http://dx.doi.org/10.1007/978-3-642-04355-0_50.

[25] G. Dini, M. Pelagatti, I. Savino, An algorithm for reconnecting wireless sensor network partitions, in: Wireless Sensor Networks, Berlin Heidelberg, in: Lecture Notes in Computer Science, Springer, 2008, pp. 253–267. http://dx.doi.org/10.1007/978-3-540-77690-1_16.



Ludovico Ferranti received his Master of Science in Engineering in Artificial Intelligence and Robotics (110/110) at Sapienza — University of Rome in 2015.

He is a Ph.D. student at Sapienza — University of Rome at DIET (Dipartimento di Ingegneria Elettronica e Telecomunicazioni) and Northeastern University of Boston at Graduate School of Engineering from November 2015. His research interests include Micro and Nanorobotics, Wireless Sensor Networks and Robotics Swarms Algorithms.

Francesca Cuomo eceived her "Laurea" degree in Electrical and Electronic Engineering in 1993, magna cum laude, from the University of Rome Sapienza. She earned the Ph.D. degree in Information and Communications Engineering in 1998 (Univ. Rome Sapienza). From 2005 she is Associate Professor at the University of Rome Sapienza and teaches courses in Telecommunication Networks. Prof. Cuomo has advised numerous master students in computer engineering, and has been the advisor of 8 PhD students in Networking. Her current research interests focus on: Vehicular networks and Sensor networks, Cognitive Radio Networks, Reconfigurable radio systems, Energy saving in the Internet and in the wireless system. She has participated in several National and European projects on wireless network systems such as the RAMON, VICOM, INSYEME, IST WHYLESS, IST EPERSPACE, IST CRUISE. Francesca Cuomo has authored over 110 peer-reviewed papers published in prominent international journals and conferences. She is IEEE Senior Member. She has been in the editorial board of the Elsevier Computer Networks journal and now is member of the editorial board of the Ad-Hoc Networks journal (Elsevier). She has been the TPC co-chair of several editions of the ACM PE-WASUN workshop and Guest Editor of the Ad-Hoc Networks Special issue on "Modeling and Performance Evaluation of Wireless Ad-Hoc Networks", 2015.