école doctorale **sciences pour l'ingénieur et microtechniques**

| **Uncertainty management in a distributed self-reconfiguration algorithm for large modular robots** |
|---|
| **Hosting laboratory: FEMTO-ST Institute** |
| **Specialty of the prepared PhD: Computer Science** |
| **Keywords: distributed algorithmics, modular architectures** |

## Detailed thesis description:

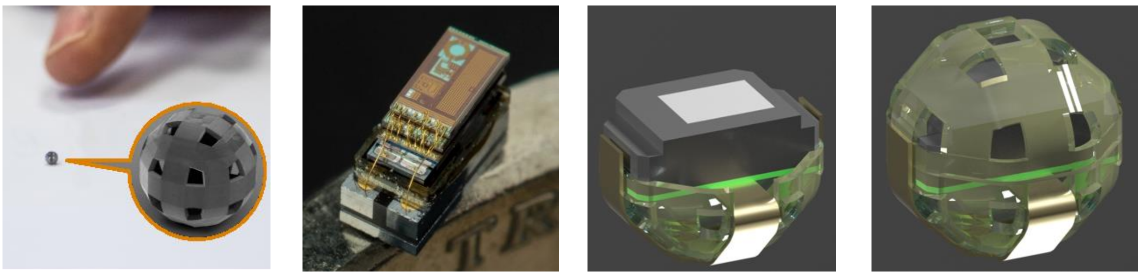
### Context

In our research, we are studying an intelligent system built from a hardware component and a software approach to create programmable matter [1]. It can be used to realize tangible representation systems, such as tangible and interactive 3D displays, as well as physical objects composed of several thousand micro-robots capable of infinite transformation as shown in the example of the mug in Fig. 1.

**Fig. 1: 3D printing of a sculpture made of programmable matter, each sphere would be a micro-robot.**

The hardware component is a millimeter-sized micro-robot [2], called a module and using electrostatic actuators to stick and move. Together, these modules form a modular robot. This device under development is presented in Fig. 2. The software approach aims at providing methods to reconfigure a set of these micro-robots to change the initial shape of the set. This approach does not consider for the moment the mistakes made by the robots, the unexpected interferences from objects in the environment, or the interventions of the user.

The process of modifying the initial shape of an object composed of modular robots is called auto-reconfiguration. It consists in finding a sequence of movements to be executed by the robots in order to transform an initial configuration into a target configuration. This is a complex problem [1]. First, given a modular robot with n modules, the number of possible configurations is huge: $(c.w)^n$, with c the number of possible connections per module, and w the number of ways to connect them together [3]. Second, the branching factor of the configuration space, describing all possible configurations that can be realized from a given configuration, is very high: $O(m^k)$, with $m$ the number of possible moves and $k$ the number of modules free to move [4]. This leads to a combinatorial explosion which makes the search space between any two configurations exponential in the number of modules.



*(a) (b) (c) (d)*

*Figure 2: (a) Shell used to support the robot: (b) a microcomputer, the Michigan Micro Mote (M3), (c) a half-shell with integration of the M3, and flexible electrodes (d) The final robot assembled by Percipio Robotics*

In order to have a fast self-reconfiguration process, we have proposed three concepts [6]: An object is first built with a porous internal structure of the scaffolding type [7], then comes a wrapping envelope of the thickness of a robot [8]. The porous structure allows the robots to pass through the interior of the object, which significantly reduces the construction time. The third device is what we called the smart base (or sandbox). It is a source of modules on which the construction is based, which serves as both a reserve and a foundation, responsible for directing a flow of modules to different points of the object.
**However, this object construction algorithm does not take into account the uncertainties that can occur during a reconfiguration, such as faults or obstacles, nor does it take into account the interventions of the user when he could interact with the set of robots, for example by pushing a part of the modules being reconfigured to indicate a change in the final shape.**

## Thesis subject

The subject of this thesis is to propose a mechanism to respond to uncertainties and user interactions while keeping the reconfiguration time as low as possible. The auto-reconfiguration algorithm will have to detect the arrival of uncertainties and adapt itself by looking for a solution by itself. It should also be able to interpret and react to the user's interactions with the modular robot.
**The first step** is to classify the different sources of uncertainties: hardware, software or external causes (such as obstacles); and the actions of the user on the modular robot. This classification will be based on existing fault classifications [9]: temporary, intermittent, permanent, byzantine, etc., in order to elaborate the most exhaustive list possible. Regarding user interactions, the literature is still sparse (e.g., the work of Kim et al. [12]), so it will be necessary to define a design space, possibly with the help of an elicitation study.
**The second step** will be to set up an algorithm to detect the most likely faults: e.g. bad connection between neighbors, or a mechanically blocked robot. Depending on the difficulty of the algorithms, a first centralized version could be considered, then be distributed using the multi-agent paradigm [10][11]. This distributed detection is already a major obstacle to overcome because some faults are very complicated to detect in a distributed way.
**The third step** will focus on detecting not the faults, but the user actions on the modular robot that will be identified by the design space (twists, translations, etc.). These actions are generally imprecise, can be continuous, spread out in time, and integrate several degrees of freedom to be evaluated. This constitutes a new step in the design of the detection mechanism developed earlier, with specific challenges.
**The fourth step** will deal with the countermeasures to bring in order to circumvent the faults detected in the second step, and to respond to the user actions detected in the third. It will be a matter of modifying the auto-reconfiguration algorithm either by modifying the actions of the robots, for example by avoiding a robot that cannot move anymore, or by bringing a software solution, for example by routing messages between robots differently in the case of a failed communication link.

**Timeline**

- Contextualization: 3 months
- Classification of uncertainties and user interactions: 6 months
- Algorithms for detecting uncertainties and user interactions: 9 months
- Countermeasures: 15 months
- Manuscript writing: 6 months

**Salary**

- Around 2,400 € per month

**References** :

[1] J. Bourgeois et al., « Programmable matter as a cyber-physical conjugation », in IEEE International Conference on Systems, Man and Cybernetics (SMC), oct. 2016.

[2] B. Piranda et J. Bourgeois, « Datom: A Deformable modular robot for building self-reconfigurable programmable matter », Kyoto, Japan, juin 2021.

[3] M. Park, S. Chitta, A. Teichman, et M. Yim, « Automatic configuration recognition methods in modular robots », Int. J. Robot. Res., vol. 27, no 3-4, p. 403-421, 2008.

[4] J. Barraquand et J.-C. Latombe, « Robot motion planning: A distributed representation approach », Int. J. Robot. Res., vol. 10, no 6, p. 628-649, 1991.

[5] M. Rubenstein, A. Cornejo, et R. Nagpal, « Programmable self-assembly in a thousand-robot swarm », Science, vol. 345, no 6198, p. 795-799, 2014.

[6] P. Thalamy, « Distributed algorithms and advanced modeling approaches for fast and efficient object construction using a modular self-reconfigurable robotic system », PhD Thesis, 2020.

[7] P. Thalamy, B. Piranda, et J. Bourgeois, « Distributed Self-Reconfiguration using a Deterministic Autonomous Scaffolding Structure », in Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems, Montreal QC, Canada, mai 2019, p. 140-148.

[8] P. Thalamy, B. Piranda, et J. Bourgeois, « 3D Coating Self-Assembly for Modular Robotic Scaffolds », in IEEE RSJ International Conference on Intelligent Robots and Systems (IROS 2020), Las Vegas, United States, oct. 2020, p. 11688-11695, doi: https://doi.org/10.1109/IROS45743.2020.9341324.

[9] M. Raynal, Fault-Tolerant Message-Passing Distributed Systems. Springer International Publishing, 2018.

[10] M. Guo, D. V. Dimarogonas, et K. H. Johansson, « Distributed real-time fault detection and isolation for cooperative multi-agent systems », in 2012 American Control Conference (ACC), 2012, p. 5270-5275, doi: 10.1109/ACC.2012.6315178.

[11] Chia-Hung Lin, Hui-Jen Chuang, Chao-Shun Chen, Chung-Sheng Li, et Chin-Ying Ho, « Fault detection, isolation and restoration using a multiagent-based Distribution Automation System », in 2009 4th IEEE Conference on Industrial Electronics and Applications, 2009, p. 2528-2533, doi: 10.1109/ICIEA.2009.5138663.

[12] Lawrence H. Kim, Daniel S. Drew, Veronika Domova, and Sean Follmer. 2020. User-defined Swarm Robot Control. Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 1–13. DOI:https://doi.org/10.1145/3313831.3376814

**Applicant profile**: Master degree in computer science

**Financing Institution: ANR Project**

Application deadline : 31/07/2023
Start of contract : 01/09/2023

**Thesis Advisor(s)** ::

**Thesis advisor: Julien BOURGEOIS**
University of Bourgogne-Franche-Comté, FEMTO-ST Institute, CNRS
*Contact for information or for applying: Julien.Bourgeois@univ-fcomte.fr*

**Co-advisor: Justin WERFEL**
Harvard University

**Co-advisor: Quentin ROY**
*University Grenoble-Alpes*