Datom: A Deformable modular robot for building self-reconfigurable programmable matter

Benoît Piranda and Julien Bourgeois

FEMTO-ST Institute, Univ. Bourgogne Franche-Comté, CNRS,
1 Cours Leprince-Ringuet - 25200 Montbéliard, France
{benoit.piranda, julien.bourgeois}@femto-st.fr

Abstract. Moving a module in a modular robot is a very complex and error-prone process. Unlike in swarm, in the modular robots we are targeting, the moving module must keep the connection to, at least, one other module. In order to miniaturize each module to few millimeters, we have proposed a design which is using electrostatic actuator. However, this movement is composed of several attachment, detachment creating the movement and each small step can fail causing a module to break the connection. The idea developed in this paper consists in creating a new kind of deformable module allowing a movement which keeps the connection between the moving and the fixed modules. We detail the geometry and the practical constraints during the conception of this new module. We then validate the capability of motion for a module in an existing configuration. This implies the cooperation of some of the other modules placed along the path and we show in simulations that it exists a motion process to reach every free positions of the surface for a given configuration.

1 Introduction

The idea of designing hardware robotic modules able to be attached together has given birth to the field of modular robotics and when these modules can move by themselves they are named Modular Self-reconfigurable Robots (MSR) [12][9] also named earlier as metamorphic robotic systems [2] or cellular robotic systems [3]. There are five families of MSR namely: lattice-based when modules are aligned on a 3D lattice, chain-type when the modules are permanently attached through an articulation, forming a chain or more rarely a tree, hybrid which is a mix between lattice-based and chain-type, mobile when each module can move autonomously and more recently continuous docking [11] where latching can be made in any point of the module. Since then, there have been many robots proposed and built by the community using different scales of modules and different latching and moving technologies. However, none of them have succeeded to reach a market.

Our objective is to build programmable matter [1] which is a matter than can change one or several of its physical properties, more likely its shape, according to an internal or an external action. Here, programmable matter will be constructed using a MSR, i.e. a matter composed of mm-scale robots, able to stick together and turn around each other as it has been described in the Claytronics project [4]. The Programmable Matter

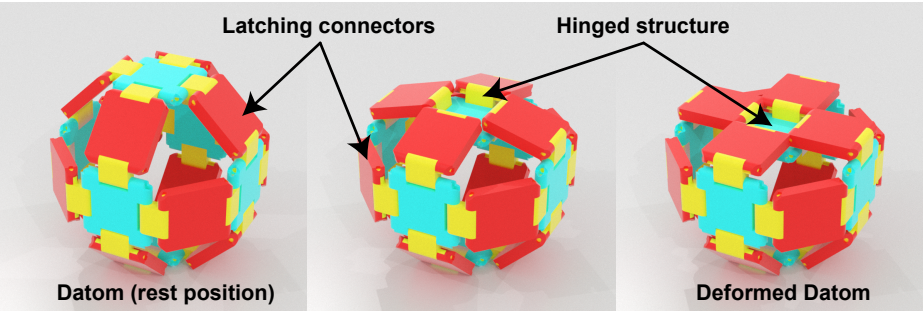


Fig. 1: Overview of the *Datom* in different deformation states.

Project[1] is a sequel of the Claytronics project and reuses most of its ideas and concepts. The requirements for each module are the following: mm-scale, being able to move in 3D, compute and communicate with their neighbors and the idea is to have thousands of them all linked together. Moving in 3D is the most complicated requirement as it needs a complex trade off between several parameters during the design phase. For example, moving requires power and, therefore, power storage, which adds weight to the module, the trade off being between having more power by adding more power storage and having a module as light as possible for easing the movement. We are currently building and testing a quasi-spherical module we designed [7]. This module rolls on another module using electrostatic electrodes. This way of moving creates uncertainty in the success of the movement as it is a complex sequence of repulsing/attaching/detaching actuations and we would like to study a movement where the moving module always stay latched to the pivot module.

The idea that drives this work is to design a motion process which never disconnects the moving and the fixed modules. We propose to define a deformable module named *Datom* (for Deformable Atom as a reference to the Claytronics Atom, Catom). Each module is strongly connected to neighbors in the Face-Centered Cubic (FCC) lattice with large latching connectors (drawn in red in Figure 1). Two connected modules must deform their shapes to align future latched connectors while the previous connection is maintained as shown in Figure 1. When new connectors are aligned they are strongly attached and the previous connection is released. Finally, the two modules return to their original shape.

2 Related works

Crystalline Robot [8], developed by Rus et al. in 2001 is an interesting solution. These robots can move relatively to each other by expanding and contracting. A robot can move a neighbor by doubling its length along $\overrightarrow{x}$ and $\overrightarrow{y}$ axes. These robots are grouped in meta-modules of 4x4 units placed in a 2D square grid. Robot to robot attachment is

[1] http://www.programmable-matter.com/

made by a mechanical system called "lock-and-key" located on the square connected faces.

In [10] Suh et al propose the Telecube, a cubic robot able to move in a cubic lattice. Similarly to previous work, Telecube can shrink using internal motors to move a neighbor. Telecube are grouped in meta-modules made of $2 \times 2 \times 2$ units. The six arms are terminated by sensors to detect neighbors and electro-permanent magnets connect the arm of the neighboring module.

The *3D Catom* model presented in [7] is a robot geometry that can move in a FCC lattice in rolling on the border of its neighbors. It uses electrostatic actuators, both for latching on planar connectors and rolling around cylindrical parts separating connectors.

3 The *Datom* model

We propose in the paper a design of a deformable robot based on the same lattice as the *3D Catom*. But the deformation must maintain some connectors in contact during the motion. Figure 2 shows the decomposition of the movements of a mobile *Datom B* (in yellow) moving around a fixed *Datom A* (in red) to go from the position shown in Figure 2.a to the position shown in Figure 2.e. We consider that connectors A_4 and B_5 are initially attached.

The first part of the motion consists in simultaneous deformations of both modules while keeping the connection between A_4 and B_5. At the middle of the deformation process (see Figure 2.c), four connectors of B are aligned in front of four connectors of A: $(\{(A_2,B_3),(A_3,B_2),(A_4,B_5),(A_5,B_4)\})$.

At this stage, four different attachment can be applied in order to reach four different positions. To move to the final destination presented in the Figure, connectors A_3 and B_2 are then attached and the connection (A_4,B_5) is released. A mirrored deformation from the previous ones moves module B to its final position.

3.1 Theoretical geometry of the deformable module

The shape of the module is deduced from the shape of the *3D Catom* proposed in [7]. From this initial geometry, we retain the position of the 12 square connectors, centered at P_i. These positions are imposed by the placement of modules in the FCC lattice.

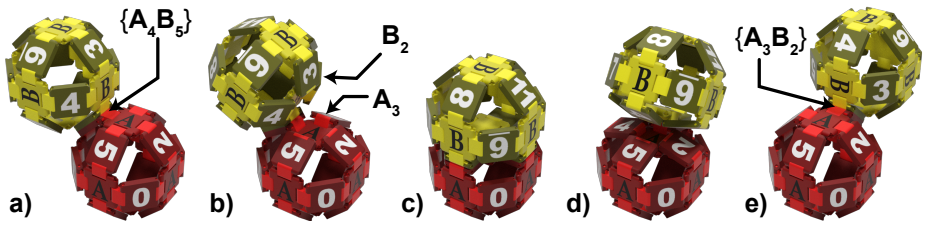


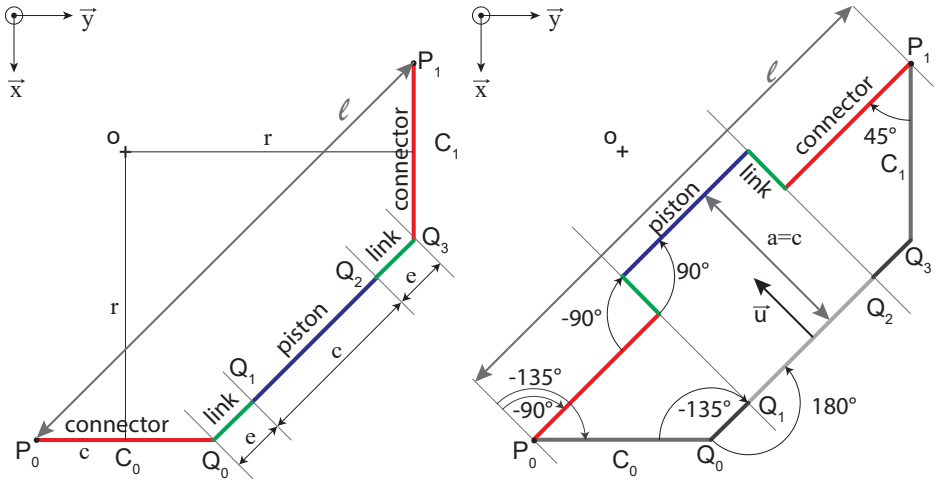
Fig. 2: Five steps of the motion of a *Datom B* around the fixed *Datom A* (the pivot). Darker elements are connectors, numbered from 0 to 11.

$$\begin{array}{c|c|c|c} P_0(r,0,0) & P_1(0,r,0) & P_6(-r,0,0) & P_7(0,-r,0) \\ P_2(\frac{r}{2},\frac{r}{2},\frac{r}{\sqrt{2}}) & P_3(-\frac{r}{2},\frac{r}{2},\frac{r}{\sqrt{2}}) & P_4(-\frac{r}{2},-\frac{r}{2},\frac{r}{\sqrt{2}}) & P_5(\frac{r}{2},-\frac{r}{2},\frac{r}{\sqrt{2}}) \\ P_8(-\frac{r}{2},-\frac{r}{2},-\frac{r}{\sqrt{2}}) & P_9(\frac{r}{2},-\frac{r}{2},-\frac{r}{\sqrt{2}}) & P_{10}(\frac{r}{2},\frac{r}{2},-\frac{r}{\sqrt{2}}) & P_{11}(-\frac{r}{2},\frac{r}{2},-\frac{r}{\sqrt{2}}) \end{array} \quad (1)$$

The size of the *Datom* is given by the distance between its two opposite connectors, this diameter is equal to $2 \times r$ (where r is the radius). The external square face of the red part presented in Figure 1 are used to latch two *Datoms* together. Electrostatic latching actuators presented in [7] with *3D Catom*, need a maximum surface. Therefore, to ensure the best connection strength between two *Datoms*, we need to maximize the size c of the square connectors.

In order to express the several dimensions of the proposed solution, we consider the plane of four co-planar connectors of a deformed *Datom*. In Figure 3, connectors of length c are drawn in red and the piston actuator of length c also is drawn in blue. Mechanical links (drawn in green) of length e are placed between piston and connectors. Figure 3a shows 2 connectors C_0 and C_1 viewed from the top. In order to align them, we propose to turn them around the $\overrightarrow{z}$ axis at points P_0 and P_1 with an angle of $+45°$ for C_0 and $-45°$ for C_1 as shown in Figure 3b. We can observe that the maximum width of a connector is limited by the 'diagonal' length ℓ which is equal to $3 \times c$, we can express also $\ell = \sqrt{2}(r + \frac{c}{2})$. We deduce the expression of c depending of the radius r:

$$c = \frac{2 \times r}{3\sqrt{2} - 1} \approx 0.61678 \times r \quad (2)$$



(a) Rest state. (b) Compress state.

Fig. 3: Size and position of the parts around one of the piston of the deformable structure. The translation of the piston leads the rotation of the links and then the connectors.

Considering Figure 3a, we can write a relation between r and c and e parameters:

$$r = \frac{c}{2} + \left(\frac{c}{2} + e\right)\sqrt{2} \tag{3}$$

That allows to express e in relation to the radius r:

$$e = r\left(\frac{2-\sqrt{2}}{3\sqrt{2}-1}\right) \approx 0.18065 \times r \tag{4}$$

3.2 Deformation process of a single *Datom*

Considering Figure 3b, we can now express the amplitude (a) of of the piston during the deformation process.

$$a = \frac{\sqrt{2}}{2}c + e = \frac{\sqrt{2}}{2}c + \left(\frac{2-\sqrt{2}}{3\sqrt{2}-1} \times \frac{3\sqrt{2}-1}{2}c\right) = c \tag{5}$$

We obtain that the amplitude of motion of the piston is equal to the size of a connector.

The deformation to compress one side of the module is obtained by translating the corresponding piston along its $\overrightarrow{u}$ axis. It implies that the angle of joint between links and connectors (Q_0) goes from $-135°$ to $-90°$ and angle of joint between links and piston (Q_1) goes from $180°$ to $90°$. Finally the angle of joint between fixed links and connectors (P_0) goes from $-135°$ to $-90°$ as shown in Figure 3b.

During this deformation, only one of the 6 pistons must move at a time in order to use the other elements as fixed supports at P_0 and P_1 joints.

3.3 Design of a thick *Datom*

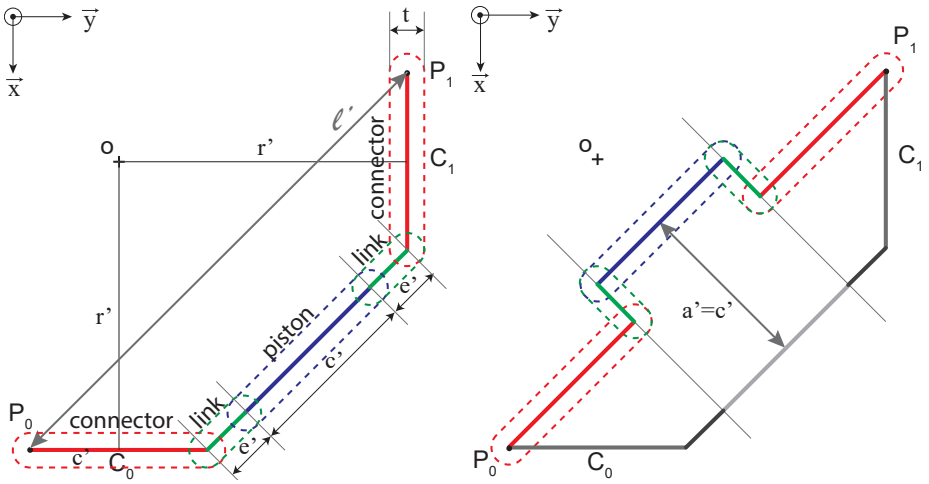
The theoretical shape of the *Datom* is not usable as is. To create a real functional module, we must consider that connectors have a not null thickness.
Let t be the thickness of the several mobiles parts of the module (connectors, links and pistons). In order to place the *Datom* in the FCC lattice, the main constraint is to keep the distance between two opposite connectors equal to $2 \times r$. We define $r' = r - \frac{1}{2}t$ as the new radius taking into account the connector thickness. Using this new radius, we can express c' and e' respectively the new size of connectors and links:

$$c' = \frac{2 \times r'}{3\sqrt{2}-1} \tag{6}$$

$$e' = \frac{2-\sqrt{2}}{3\sqrt{2}-1}r' \tag{7}$$

The geometry of the link part implies that the thickness t must be less than $e' = 0.19859 \times r$ (See Figure 4).

We use rotation limits of each joint (between connector and link and also between link and piston) to shape blocking plots. These blocking plots help for the stability of the whole system. For example, Figure 5 shows blocking plots for the joint between the connector and the link parts.



(a) Rest state. (b) Compress state.

Fig. 4: Size and position of components taking into account of the thickness.

4 Motion capabilities in an ensemble

We now consider a configuration of several *Datoms* placed in a FCC lattice. The motion of a *Datom B* is possible around another connected *Datom A* (called the pivot) only. Considering two *Datoms* linked by a couple of connectors, it exists 2 available directions of motions, each one associated to a piston of the pivot. And we will show below that for each of these directions there is 3 possible motions (*"Turn left"*, *"Go ahead"* and *"Turn right"*). Finally, each mobile *Datom* having 12 connectors, it implies that each *Datom* is able to make 72 different motions, but only a few of them are possible (or valid) at a time. Then we need a tool to detect the real list of valid motions depending on the neighborhood of the couple of *Datoms*.

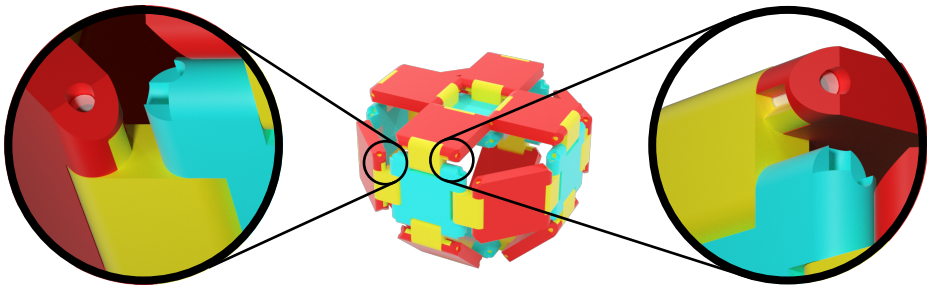


Fig. 5: Zooms on mechanical joints to show the angular blocking plots between the connector and the link parts.

The motion rules method proposed by Piranda et al. in [6] consists in defining a list of motions that are available for a considered module, taking into account several constraints in the neighboring cells of the lattice.

To simplify the explanation, we consider a couple of connectors of the *Datom B* and the pivot *A* that produces a motion in the plane $(A, \overrightarrow{x}, \overrightarrow{y})$, with the translation of the piston placed at the top of *A*. To simplify notations in the following, we use the same letter to name a free cell of the lattice and the *Datom* placed in the cell if it exists.

Definition 1 *A motion rule is a list of tuples* (P,S) *where P is a position in the grid relative to the pivot A and S is a status of the cell placed at position P. Status S can have one of the following values, or a combination of* $\emptyset$ *and one of these values:*

- $\emptyset$*, if the cell must be empty (no module at this position),*
- *a module name, if the cell must be filled by this Datom,*
- $def(\overrightarrow{X})$ *if the cell must be filled by a deformed module, the deforming piston being oriented in the direction* $\overrightarrow{X}$*,*
- $def(\overrightarrow{X}, \overrightarrow{Y})$*, if the cell must be filled by a Datom initially deformed along* $\overrightarrow{X}$ *axis and deformed along* $\overrightarrow{Y}$ *axis at the end of the motion.*

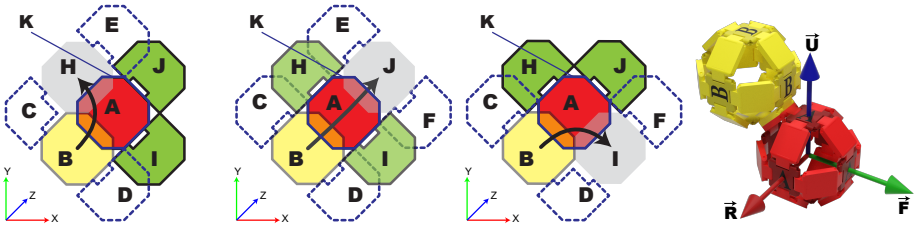
Theorem 1 *A motion rule is valid if all tuples of its list are validated by the current configuration. The Table 2 gives the list of tuples for each motion rules.*

Table 2 gives the list of tuples for the three possible motions of *B* around the pivot *A* using the deformation consisting in translating the piston of *A* defined by vector $\overrightarrow{U}$. The direction to the right of $\overrightarrow{BA}$ is defined by $\overrightarrow{R} = \overrightarrow{BA} \times \overrightarrow{U}$ and the front direction $\overrightarrow{F} = \overrightarrow{U} \times \overrightarrow{R}$ are expressed relatively to the positions of *A* and *B* (cf. Figure 6d).
Every motion rule is defined relatively to a pivot *A* placed at the origin of the system and *B* at the top rear position of *A*. Using these relative directions we can express a motion rule for all the 72 different possible motions. We add the two contextual following rules $\{((0,0,0),A),(\overrightarrow{U} - \overrightarrow{F},B)\}$ that express that the position $(0,0,0)$ must be filled by *A* and the position of *B* is obtained adding the vector $(\overrightarrow{U} - \overrightarrow{F})$ to the position of *A*.

Figure 6 shows for each basic motion, an initial configuration that make it valid and every cells used by at least one motion rule tuple.
First, we consider the horizontal plane of cells containing the moving module *B* in yellow (level 0), which also contains *H*, *I* and *J* cells. Two of these cells are drawn in green and the third one corresponding to the goal cell is in grey. The pivot *A* (drawn in red) is placed in the underneath plane of cells (level -1). Also, we must take into

Table 1: Position of *Datoms* concerned by the motion of *B* (relatively to *B*).

Cell	Position	Cell	Position	Cell	Position
A	$(1,1,-1)$	*E*	$(1,2,1)$	*J*	$(1,1,0)$
B	$(0,0,0)$	*F*	$(2,1,1)$	*H*	$(0,1,0)$
C	$(0,1,1)$	*H*	$(0,1,0)$	*K*	$(0,0,1)$
D	$(1,0,1)$	*I*	$(1,0,0)$		



(a) Turn left. (b) Go ahead. (c) Turn right. (d) Motion rule axes.

Fig. 6: The three possible motions of a module B linked to a pivot A to reach position G. For each motion we show the cells used by "motion rules" in the neighborhood of A. Last picture shows relative axes for a couple of *Datom* and a piston (here the top of A).

account the cells placed on the plane above B: C, D, E, F and K placed on level 1. Table 1 precises the coordinates of each *Datom* relatively to B, and Figure 7a shows a 3D view of the *"Turn left"* initial configuration using the same color system.

Cells with dotted border of Figure 6 may contain a module or be free but the goal cells must initially be free of module. At the top plane, the cell K placed over A must be free, while cells C, D, E and F may contain a module that must be deformed to free the path for B.

The three available motions are presented separately, Figure 6a: *"Turn left"* with a goal placed at H, Figure 6b: *"Go ahead"* with J as goal and Figure 6c: *"Turn right"* allowing B to reach cell I. In the case of *"Turn left"* (resp. *"Turn right"*) motion, if the cell C (resp. D) is filled, these corner *Datoms* must be deformed twice during the motion. The first deformation allows B to go on the top of A, then the deformation changes to allow B to reach its final position.

For example, in the first case (Figure 6c), before moving module B to the goal cell I using A as a pivot, we must verify that the cell K on the plane on top of A is empty and then ask modules eventually placed at the C, D, F, H and J cells to deform themselves in order to free the path. Figure 7 shows some key steps of the *"Turn left"* motion of the module B in 3D and a video[2]: available on *YouTube* shows every details of the three different motions:

a) Initial configuration, *Datom* B plans to turn to left, it sends messages to the cells I, J, C, D and E to ask *Datoms* eventually filling these cells to free the path by deforming themselves (in the Figure, only I, J, D and E are represented).
b) *Datoms* I, J, D are now deformed, B can start the motion.
c) B is actuating synchronously with the pivot A to create the motion.
d) A and B are in the middle stage of the motion, they both change the connectors attachment. If there is a *Datom* in cell F, it changes its deformation to allow the final motion of B.
e) B reaches its final position, and asks C, D, E, F and H to release their deformation.
f) Final configuration.

[2] Motions of simulated and printed Datoms on video: https://youtu.be/EHuBtkRV4Jo

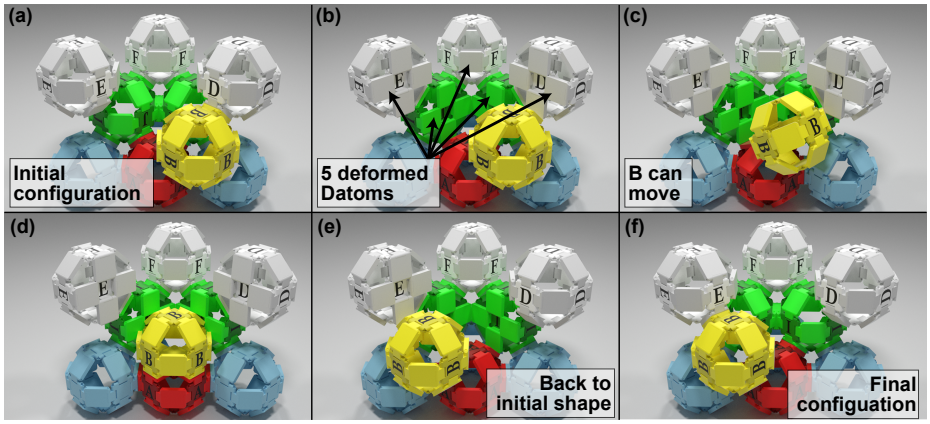


Fig. 7: Six steps of *turn left* displacements of the module B in a constrained configuration. a) Initial configuration, b) First the blocking modules must be deformed. c-e) Motion steps. f) Final configuration.

Table 2: Motion rules applied from the pivot

Rule	Tuples list by motion	Cell
All	$\{((0,0,0),A),$	A
	$(\overrightarrow{U}-\overrightarrow{F},B),$	B
	$(2\overrightarrow{U},\emptyset)\}$	*K*
Turn left	$\{(\overrightarrow{U}-\overrightarrow{R},\emptyset),$	Goal
	$(\overrightarrow{U}+\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{F})),$	*J*
	$(\overrightarrow{U}+\overrightarrow{R},\emptyset\vee\mathsf{def}(-\overrightarrow{R})),$	*I*
	$(2\overrightarrow{U}+\overrightarrow{R}-\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{R})),$	*D*
	$(2\overrightarrow{U}-\overrightarrow{R}-\overrightarrow{F},\emptyset\vee\mathsf{def}(\overrightarrow{R},\overrightarrow{F})),$	*C*
	$(2\overrightarrow{U}-\overrightarrow{R}+\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{F}))\}$	*E*
Turn right	$\{(\overrightarrow{U}+\overrightarrow{R},\emptyset)$	Goal
	$(\overrightarrow{U}-\overrightarrow{R},\emptyset\vee\mathsf{def}(\overrightarrow{R})),$	*H*
	$(\overrightarrow{U}+\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{F})),$	*J*
	$(2\overrightarrow{U}-\overrightarrow{R}-\overrightarrow{F},\emptyset\vee\mathsf{def}(\overrightarrow{R})),$	*C*
	$(2\overrightarrow{U}+\overrightarrow{R}-\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{R},\overrightarrow{F})),$	*D*
	$(2\overrightarrow{U}+\overrightarrow{R}+\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{F}))\}$	*F*
Go ahead	$\{(\overrightarrow{U}+\overrightarrow{F}),\emptyset$	Goal
	$(\overrightarrow{U}-\overrightarrow{R},\emptyset\vee\mathsf{def}(\overrightarrow{R})),$	*H*
	$(\overrightarrow{U}+\overrightarrow{R},\emptyset\vee\mathsf{def}(-\overrightarrow{R})),$	*I*
	$(2\overrightarrow{U}-\overrightarrow{R}-\overrightarrow{F},\emptyset\vee\mathsf{def}(\overrightarrow{R})),$	*C*
	$(2\overrightarrow{U}+\overrightarrow{R}-\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{R})),$	*D*
	$(2\overrightarrow{U}+\overrightarrow{R}+\overrightarrow{F},\emptyset\vee\mathsf{def}(-\overrightarrow{R})),$	*F*
	$(2\overrightarrow{U}-\overrightarrow{R}+\overrightarrow{F},\emptyset\vee\mathsf{def}(\overrightarrow{R}))\}$	*E*

A particular case must be considered when C, D, E or H modules are only attached by one of the 4 connectors linked to the compressed piston. In this case, they must be removed first in order to make the motion of B possible.

5 Simulation

Simulations have been executed in *VisibleSim* [5], a modular robot simulator[3]. The goal of these experiments is to show that a *Datom* can reach every free positions at the surface of a configuration in successively applying the unitary motions presented above.

5.1 Algorithms

We implement a first algorithm that places a *Datom* at a goal cell G placed on the surface of a configuration. Then from this position, it computes every valid motions from this point testing successively the rules presented Table 2, the reached positions are memorized in every neighbor modules. According to Theorem 2, it exists a sequence of motions to go from each of these cells to G. More precisely, we implement a gradient algorithm that affects the distance 0 to the G cell, then the distance 1 is set to each cell which can reach the G cell after exactly one basic motion, and so on. It allows to define a gradient of distances to go from every reachable cells to G.

Theorem 2 (Bidirectional motions) *Each displacement is bidirectional, if a motion rule is valid to go from a cell X to a cell Y, it exists a valid motion rule to go from Y to X.*

Proof. If it exists a valid *"Go ahead"* motion rule to go from X to Y, as the motion constraints are symmetrical relatively to the up direction $\overrightarrow{U}$ of pivot A, the *"Go ahead"* motion rule will be valid for a motion from Y to X, using the same pivot A.

"Turn left" and *"Turn right"* motion rules are symmetrical relatively to the plane $(A, \overrightarrow{F}, \overrightarrow{U})$. If it exists a valid *"Turn left"* motion rule to go from X to Y, it exists a valid *"Turn right"* motion rule to go from Y to X, and reciprocally.

A second algorithm has been implemented to move a module B (with $ID = 1$) from one cell (a free cell of the border) to the goal position. The *Datom B* computes the list of reachable free cells from its current position. It then selects one of the cell which the minimum distance value. It sends a message to all modules that must be deformed to allow its motion and, after an acknowledgement applies the motion. And so on, until it reaches the goal cell which distance is 0.

5.2 Results

For this experiment, we construct a simulated configuration made of 141 *Datoms*. A $7 \times 7 \times 2$ box is covered by an obstacle making an arch whose hole is two *Datoms* high

[3] Video presentation of *VisibleSim*: https://youtu.be/N09KElCbUNk

(cf. Figure 8.a). And in a second time, we add a red *Datom* that reduces the size of the hole to one *Datom* high only (cf. Figure 8.b).

For these two configurations, we calculate the distance from the position $G(6,5,2)$ in the lattice (the position of the green module) to all reachable cells. The distance of these cells is represented in the second screenshot (center) where the configuration is viewed from the top. Colored spheres are placed at the center of the cells reachable from the goal position (marked by the red sphere), the spheres' color represents the distance from the cell to the goal. We can observe that distances of cells at the left of the configuration are higher in the second case because the red *Datom* suppresses the shortcut passing through the arch.

The third screenshot presents the results of the second algorithm for the two cases. The red line shows the steps of the motion of the green module from the position $(0,0,2)$ marked by the blue *Datom* to the goal position (green *Datom*). In the top image, the *Datom* can pass under the arch while in the second image the path goes above the obstacle.

A video that shows the deformation of a 3D printed version of the *Datom* and some results obtained on the simulator is available on *YouTube* [4].

6 Conclusion

This work proposes a new model of deformable robot for programmable matter called a *Datom* which allows to realize safe motions in a FCC lattice. The size of the components relatively to the diameter of the robot and the angular limits between these pieces are precisely detailed for the realization of a real robot.

We study precisely how to implement the motion of a module in an ensemble to allow a module to step by step reach every free cell at the surface of a configuration.

[4] *YouTube* video: https://youtu.be/EHuBtkRV4Jo

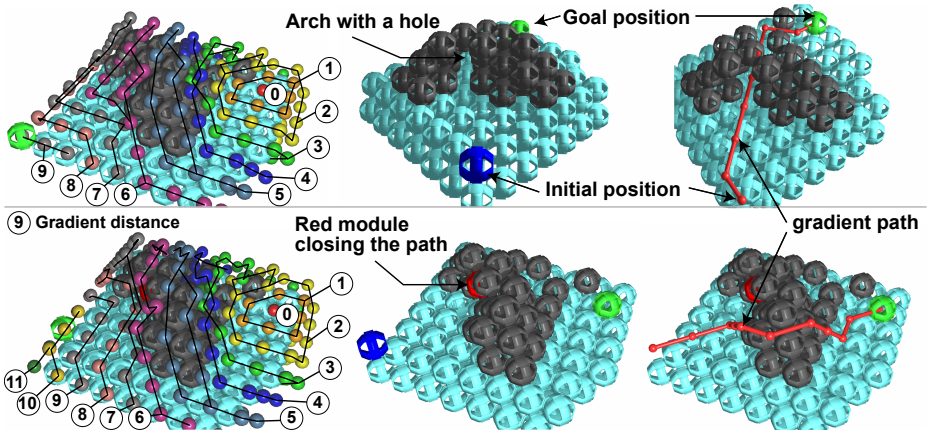


Fig. 8: Simulation results of the two algorithms (gradient and motion) on two similar configurations.

These motions are possible if many other modules collaborate and must synchronize their own deformation, in order to free the path for another one.

In order to build a mm-scale *Datom*, we envision the following steps. First, we will use our microtechnology center to produce the 3D printed structure of the *Datom*. This structure can fully be printed without further assembly as the hinges can be directly printed. The latching can use the same electrostatic electrodes developed for the *3D Catom* [7] and the deformation could be obtained using a soft bimorph actuator.

ACKNOWLEDGMENT

This work was partially supported by the ANR (ANR-16-CE33-0022-02), the French Investissements d'Avenir program, the ISITE-BFC project (ANR-15-IDEX-03), and the EIPHI Graduate School (contract ANR-17-EURE-0002).

References

1. Bourgeois, J., Piranda, B., Naz, A., Boillot, N., Mabed, H., Dhoutaut, D., Tucci, T., Lakhlef, H.: Programmable matter as a cyber-physical conjugation. In: IEEE Intl Conf. on Systems, Man, and Cybernetics (SMC), pp. 002,942–002,947. Budapest, Hungary (2016)
2. Chirikjian, G.S.: Kinematics of a metamorphic robotic system. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 449–455. IEEE (1994)
3. Fukuda, T., Kawauchi, Y., Buss, M.: Communication method of cellular robotics cebot as a selforganizing robotic system. In: IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS), pp. 291–296. IEEE (1989)
4. Goldstein, S.C., Mowry, T.C.: Claytronics: An instance of programmable matter. In: Wild and Crazy Ideas Session of ASPLOS. Boston, MA (2004)
5. Piranda, B.: VisibleSim: Your simulator for Programmable Matter. In: Algorithmic Foundations of Programmable Matter (Dagstuhl Seminar 16271). Dagstuhl (2016)
6. Piranda, B., Bourgeois, J.: A distributed algorithm for reconfiguration of lattice-based modular self-reconfigurable robots. In: PDP 2016, 24th Euromicro Int. Conf. on Parallel, Distributed, and Network-Based Processing, pp. 1–9. Heraklion Crete, Greece (2016)
7. Piranda, B., Bourgeois, J.: Designing a quasi-spherical module for a huge modular robot to create programmable matter. Autonomous Robot Journal **42**(8), 1619–1633 (2018)
8. Rus, D., Vona, M.: Crystalline Robots: Self-Reconfiguration with Compressible Unit Modules. Autonomous Robots **10**(1), 107–124 (2001)
9. Støy, K., Brandt, D., Christensen, D.J., Brandt, D.: Self-reconfigurable robots: an introduction. Mit Press Cambridge (2010)
10. Suh, J.W., Homans, S.B., Yim, M.: Telecubes: Mechanical design of a module for self-reconfigurable robotics. In: ICRA, vol. 4, pp. 4095–4101 (2002)
11. Swissler, P., Rubenstein, M.: Fireant: A modular robot with full-body continuous docks. In: Proceedings of the 2018 IEEE International Conference on Robotics and Automation (2018)
12. Yim, M., Shen, W.M., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S.: Modular self-reconfigurable robot systems [grand challenges of robotics]. IEEE Robotics & Automation Magazine **14**(1), 43–52 (2007)