R-Cell: A Module for a Self-Reconfigurable Robotic System

Dimitris M. Chatzigeorgiou, Savvas G. Loizou and Kostas J. Kyriakopoulos

Abstract— **In this paper, we introduce the concept of a novel robotic module, the "*R-Cell*". *R-Cell* can be utilized in constructing distributed, homogeneous robotic systems. Each R-Cell is a rectangle endowed with: (a) motion capabilities provided by four revolutionary joints, each equipped with a clamping mechanism, and (b) deformation capabilities realized by four prismatic joints. The proposed module provides the resulting modular robotic structures with dexterous deformation and force creating capabilities. The resulting robotic structures are not defacto rigid, but can change shape & form even without any cell-reconfiguration. Our concept could be useful for a great variety of applications encompassing modular robotics like self-assembly, self-repair and reactive shape optimization just to mention a few, that conventional robots cannot accommodate.**

Index Terms - **modular robotic system, self-reconfiguration, deformation, force-creating capabilities**

I. INTRODUCTION

Usually the design of robotic systems is dictated by the nature of their assigned tasks. In many cases, however, robotic systems capable of performing a diversified array of tasks are sought. One of the issues that arise when considering such multi-purpose robotic systems is the stringent constraints that have to be accommodated when addressing the relevant motion planning and control problems.

One way of alleviating those constraints is to consider modular robotic systems with the capability of self-reconfiguration. *Self-Reconfigurable* (SR) robots can change their shape and functionality, and are capable of "*self-assembly*" and "*self-repair*" [1], [2]. The capability of self-assembly implies that a set of modules can form a specific global configuration without external intervention. The self-repair capability implies that the system has the ability to restore its functionality even if some of its modules fail. Self-assembly and self-repair are indispensable in many areas where human operators have limited or difficult access (e.g. in space [3], nuclear power plants, deep sea or even nano-scale applications).

The roots of modular SR robotics can be traced back in the late 1980s, when *CEBOT* (*CEllular roBOT*) was introduced [4], [5]. Later, two types of modular SR robots were developed, the *Lattice-type* and the *Chain-type*.

Modules of lattice-type SR robots are like biological cells or crystal atoms in matter. The modules are periodically positioned in predefined grid points in order to build the desired structure and have in most cases a lot of actuators and connectors. Based on that concept several SR robots were developed. The *Metamorphic Robot* [6] and the *Fracta* [7] were two of the most successful steps in that direction with many others that followed later. One of the most important was *ATRON*, a lattice-type module created at the University of Southern Denmark with one degree of freedom (DOF) and a high power/weight ratio [9], [10]. Moreover, the *Molecule*, another 3D module for a SR robotic system, was introduced in 1998 [8]. In addition, the *Crystalline Robots* by D.Rus [17] and the *Telecubes* [18] by M.Yim belong to the same category.

The Chain-type modules are connected in a string or tree-topology. That topology may fold up to become space-filling. It is easy to consider that modules from that category build less symmetrical structures and require in most cases less actuators and connectors. The first robot in this category was the *Polybot* [11] and other typical robots are *CONRO* [12], [13] and the *Molecube* [16]. The latter was developed by *V. Zykov* in Cornell University.

Lately a new category arose, apart from lattice and chain-type SR robots, namely the *Hybrid SR robots*. Those robots try to combine all the advantages of the previous categories. The most successful one was the *Modular TRANsformer* (*M-TRAN*) in 1999 [14], [15] . The most recent version of the module, namely the *M-TRAN III*, was built in 2005.

In this paper the concept of an innovative module for a SR robotic system, the *R-Cell*, is introduced. Our module belongs to the Lattice-type family. Due to their special design, R-Cells manage to combine both self-reconfiguring abilities with dexterous motion capabilities. There are four revolutionary joints surrounding the R-Cell that allow the module to rotate around common joints and reposition itself. There are also four prismatic joints in the inner body of the cell that allow R-Cells to "*deform*" and to apply forces within its workspace. Thus, the SR robotic structure consisting of R-Cells is also able to *deform its shape* to some extent. The resulting structure can not only reconfigure itself by rearranging its modules, but also change its functionality by deforming its shape and interact with the environment, i.e. apply *forces* and *torques*, emanating from that "*deformation*" ability. In this paper we investigate the main idea behind the R-Cell concept, which is the "*deformation*" ability and the *force-creating mechanism* of a robotic structure consisting of R-Cell modules.

This novel "deformation" ability could be useful for a great variety of applications. Robots with deformation abilities would be able to "adjust" themselves to external forces or even change their shape in order to apply forces and torques in a more efficient way than simple SR robots or even rigid monolithic robots. Moreover robots equipped with

The authors are with the Control Systems Laboratory, Mechanical Engineering Department, National Technical University of Athens, 9 Heroon Poltechniou Str., Zografou - 15700 Athens, Greece. {dchatzis@csl.mech.ntua.gr, sloizou@mail.ntua.gr, kkyria@central.ntua.gr}

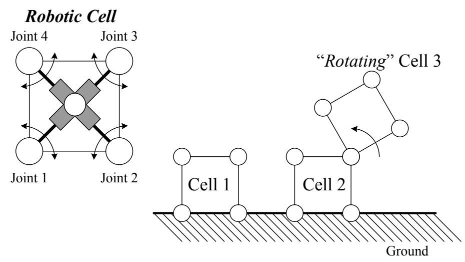

Fig. 1. [Left] The robotic cell and the nodes with their revolutionary joints. [Right] Two cells lying on the ground and a third one rotating in order to be placed on top of the second cell.

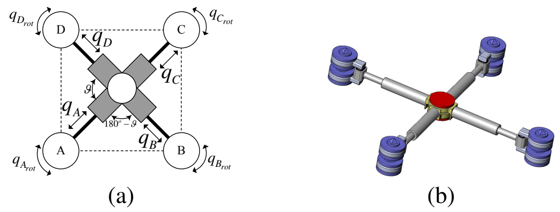

Fig. 2. (a) A 2-D sketch of an R-Cell. All joints (rotational and prismatic) are depicted. (b) A conceptual 3D design of an R-Cell.

this "deformation" ability will be more flexible in strict space conditions. In general, with this ability each structure will be endowed with a distributed force creation mechanism which can accommodate both geometric constraints and allow for optimization.

The rest of the paper is organized as follows: In section II we introduce the basic idea and the conceptual design of our module. In section III we propose a representation technique to facilitate our analysis and we proceed by studying the the Direct- (DK), Inverse- (IK) and Differential-Kinematics (DF). In section IV using concepts from graph theory we propose a general technique for analyzing robotic structures consisting of R-Cells. Section V presents two case studies where the deformation and force delivery capabilities of R-Cells are demonstrated and the paper concludes with section VI.

II. THE ROBOTIC CELL DESIGN

We introduce a new concept of a modular SR robotic system (Fig. 1) composed of Robotic Cells (R-Cells), able to:

1) *rotate* around common joints,
2) *clamp* to each other, and
3) *deform* their shape.

Relative rotation is necessary to allow formulation of *arbitrarily shaped structures*, while clamping is needed to enable structure rigidity when it is needed. Deformability is needed to allow creation of internal forces needed to implement an arbitrary force at a specific reach point (node). Thus cells are not expected to deform while rotating but may do so only when they have acquired their desired position in the structure.

In this work we are addressing only *planar* structures. Furthermore we consider the case where all R-cells are the same, which means that the whole robotic structure is *homogeneous*. The R-Cell has a *cross-like shape* and consists of *four prismatic and five rotational joints* (Fig. 2.a). During R-Cell's "deformation" the four prismatic joints q_A, q_B, q_C, q_D are actuated accordingly, while the angle θ and the rest rotational joints $q_{A_{rot}}, q_{B_{rot}}, q_{C_{rot}}, q_{D_{rot}}$ remain completely passive. Whilst during the "self-assembly" stage the prismatic joints and the previously passive joint θ remain fixed in the default/neutral position and the rotational joints $q_{A_{rot}}, q_{B_{rot}}, q_{C_{rot}}, q_{D_{rot}}$ need to be actuated in order to achieve relative rotation between R-Cells. Consequently, 8 actuators need to be used per module. 4 of them are "responsible" for the deformation capabilities of the R-Cell, and the rest need to be activated only during the relative rotation (self-assembly) phase. A 3D conceptual design of the R-Cell is shown in Fig. 2.b. The clamping mechanism of the R-Cell is not yet fully defined and designed but the main concept is that each clamping mechanism has the ability to "open" and "close" in order to embrace or to be embraced by another module's clamping mechanism, i.e. to act like a "gripper".

III. ANALYSIS OF THE R-CELL

A. Nodes and DoFs distribution

As mentioned above each cell has four *prismatic* and five *rotational* joints in order to accommodate all the set requirements. The rotational joint in the middle of the R-Cell is non-actuated. If a node is not controlled by the module, we may assume that the global position of this specific node is *fixed in space by another module or the environment around it. When a cell is assembled it must have at least two fixed adjacent nodes*. The latter means that each cell should "stand" on two fixed nodes in order to become steady and be able to move the rest of its nodes independently. There are also cells that "stand" on three fixed nodes, which means that there is only one node left for them to move independently. So, there are *two types* of actuated cells in an assembled robotic structure, i.e. cells with two "free" nodes and cells with only one "free" node. Due to the fact that our robotic structure is planar, cells with two "free" nodes have 4 DoFs, while the rest only 2 DoFs. An example of the DoFs distribution among the R-Cells within a structure is given in Fig. 3.

TABLE I

FIXED AND FREE NODES FOR EACH R-CELL OF FIG. 3

Cell	Fixed nodes	Free nodes	Number of DoFs
(1,1)	1,2	3,4	**4**
(1,2)	2,5,3	6	**2**
(1,3)	6,5	7,8	**4**
(2,1)	4,3	9,10	**4**

B. Structure Matrix

The next arising issue is how a desired shape structure (Fig. 4) is materialized. A structure may consist of any

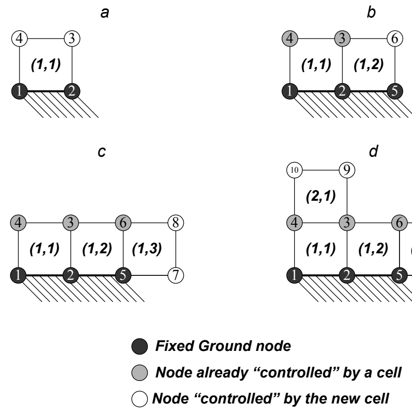

Fig. 3. Structural DoFs distribution among the R-Cells

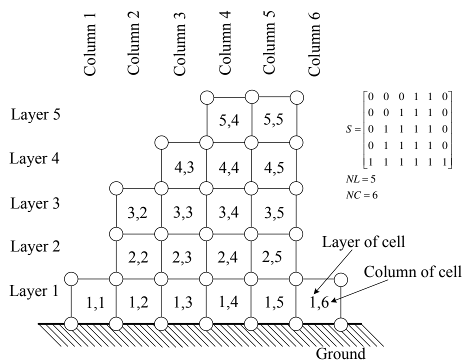

Fig. 4. A robotic structure consisting of $n = 19$ R-Cells. The layers and columns of the structure are spelled out. The matrix S is shown on the right side.

number of R-Cells (n). In order to simplify the representation of the proposed modular robotic structure the binary matrix S is introduced. If NL and NC are the numbers of "*Layers*" and "*Columns*" of the desired structure, respectively, then S is a $(NL \times NC)$ matrix defined by:

$$S_{i,j} = \{ {}^{1\ if\ a\ cell\ is\ placed\ in\ (i,j)}_{0\ if\ a\ cell\ is\ not\ placed\ in\ (i,j)} \tag{1}$$

C. Kinematic Analysis of the R-Cell module

To identify the kinematic capabilities of the R-Cell, we will start by examining the kinematics of R-Cells with two free nodes (4 DoFs) and then R-Cells with only one free node (2 DoFs) (Fig. 5).

1) R-Cell with 4 DoFs: At first the Direct Kinematics (DK) equations are derived. Given the position vectors $\mathbf{p}_A, \mathbf{p}_B$ of the fixed nodes A and B, we can find $\mathbf{p}_E$ by solving the following equation system:

$$\|\mathbf{p}_E - \mathbf{p}_A\| = q_A \tag{2}$$

$$\|\mathbf{p}_E - \mathbf{p}_B\| = q_B \tag{3}$$

By solving the previous system one is able to derive two solutions for $\mathbf{p}_E$. The correct one will be the one in the upper or lower half plane defined by the line passing through AB depending on how we have arranged the R-Cell (e.g for the arrangement in figure 5 the correct lives in the upper half plane). The solution of the direct kinematics problem for nodes C and D is straightforward. The most important conclusions about the DK equations of an R-Cell with 4 DoFs are grouped in Table II.

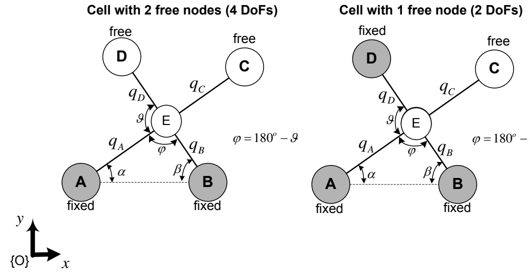

Fig. 5. [Left] R-Cell with two fixed nodes (A and B) and two free nodes (C and D). [Right] R-Cell with three fixed nodes (A, B and D) and one free node (C).

TABLE II

MAIN CONCLUSIONS REGARDING THE DK ANALYSIS OF AN R-CELL

R-Cell with 4 DOFs

Node	**Dependent on the following parameters**
Node C ($\mathbf{p}_C$)	$f(q_A, q_B, q_C, \mathbf{p}_A, \mathbf{p}_B)$
Node D ($\mathbf{p}_D$)	$f(q_A, q_B, q_D, \mathbf{p}_A, \mathbf{p}_B)$

R-Cell with 2 DOFs

Node	**Dependent on the following parameters**
Node C ($\mathbf{p}_C$)	$f(q_B, q_C, \mathbf{p}_A, \mathbf{p}_B, \mathbf{p}_D)$

The Inverse Kinematics (IK) are trivially solvable (see e.g. [20]). For deriving the Jacobian matrix, according to the conclusions from Table II we can see that the Differential Kinematics (DF) equations of the system will be as follows:

$$\dot{\mathbf{p}}_C = \frac{\partial f}{\partial q_A}\dot{q}_A + \frac{\partial f}{\partial q_B}\dot{q}_B + \frac{\partial f}{\partial q_C}\dot{q}_C + \frac{\partial f}{\partial \mathbf{p}_A}\dot{\mathbf{p}}_A + \frac{\partial f}{\partial \mathbf{p}_B}\dot{\mathbf{p}}_B \tag{4}$$

$$\dot{\mathbf{p}}_D = \frac{\partial f}{\partial q_A}\dot{q}_A + \frac{\partial f}{\partial q_B}\dot{q}_B + \frac{\partial f}{\partial q_D}\dot{q}_D + \frac{\partial f}{\partial \mathbf{p}_A}\dot{\mathbf{p}}_A + \frac{\partial f}{\partial \mathbf{p}_B}\dot{\mathbf{p}}_B \tag{5}$$

2) R-Cell with 2 DoFs: For this type of cell the DK equations are derived. Given the position vectors $\mathbf{p}_A, \mathbf{p}_B$ and $\mathbf{p}_D$ of the three fixed nodes A, B and D, we can find $\mathbf{p}_E$ by solving the following equation system:

$$\|\mathbf{p}_E - \mathbf{p}_B\| = q_B \tag{6}$$

$$\|\mathbf{p}_E - \mathbf{p}_D\| = q_D \tag{7}$$

while satisfying that:

$$q_D = \|\mathbf{p}_D - \mathbf{p}_B\| - q_B \tag{8}$$

Then the analytical solution DK problem for the free node C is straightforward in this case too. The most important conclusions about the DK equations of this case are grouped in Table II.

The IK is a trivial problem here too. We are going to introduce the DF equations of this case. According to the conclusions from Table II we can assume that:

$$\dot{\mathbf{p}}_C = \frac{\partial f}{\partial q_B}\dot{q}_B + \frac{\partial f}{\partial q_C}\dot{q}_C + \frac{\partial f}{\partial \mathbf{p}_A}\dot{\mathbf{p}}_A + \frac{\partial f}{\partial \mathbf{p}_B}\dot{\mathbf{p}}_B + \frac{\partial f}{\partial \mathbf{p}_D}\dot{\mathbf{p}}_D \tag{9}$$

By using equations (4),(5) for an R-Cell with 4 DoFs and equation (9) for an R-Cell with only 2 DoFs, one can easily build the Jacobian matrix (J) of a single assembled module. In order to build the extended Jacobian of a whole robotic structure consisting of n R-Cells one has to repeat this procedure n times for all cells within the structure.

IV. Analysis of a Robotic Structure

In this section the typical steps for a complete analysis of a robotic structure consisting of R-Cell modules are introduced. In order to analyze the DoFs distribution among all R-Cells within the robotic structure it is more than obvious that the structure matrix alone doesn't provide all required information. For this purpose *graph theory* [19] was utilized. This means that the whole system consisting of n R-Cells will be fully described by a graph. More specifically the steps that have to be followed are:

- Define the Structure Matrix S (see Eq. 1).
- Define the *"Ground-Nodes"*. These are the nodes which "support" the robotic structure.
- Build the Graph (X) of the whole system. In this case the graph is *undirected*.
- Build the "*Adjacency Matrix*" $A(X)$ of the previous graph.
- Build the "*Adjacency Matrix*" $A(Y)$ of the equivalent *directed* graph (Y). This matrix includes all required information for the final robotic structure.

Firstly, the structure matrix has to be defined. The choice of the number of R-Cells (n) and their configuration is a difficult problem with great importance for the functionality of the final robot. One knows the physics and functionality of each individual component, the repertoire of modules available and the way they are able to clamp to each other. But one doesn't know the combination of the modules that will result in the desired/optimum functionality for the final structure. Moreover, the ground nodes have to be defined by the user. These are the nodes that will support the whole robotic system. It is obvious that *at least two nodes* have to be defined but the more ground nodes the user defines the more stable and strong the final structure will be. This problem is similar to almost any problem of structure support an engineer faces very frequently.

In order to build the graph (X) of the whole system we have to introduce the *center-nodes*. These center-nodes are nothing more than the nodes standing in the middle of each R-Cell's graph and represent their center points. So, in the graph each cell will be represented by a simple graph consisting of four external nodes and one center-node. After this step one is able to build the adjacency matrix of the previous graph without any difficulties, according to graph theory.

The most important and non-trivial step is the transition from $A(X)$ to $A(Y)$ of the equivalent directed graph, which is our main goal. The whole process is described via a block diagram in Fig. 6. The matrix $A(Y)$ includes all important information about the final robotic structure including:

- the ground nodes

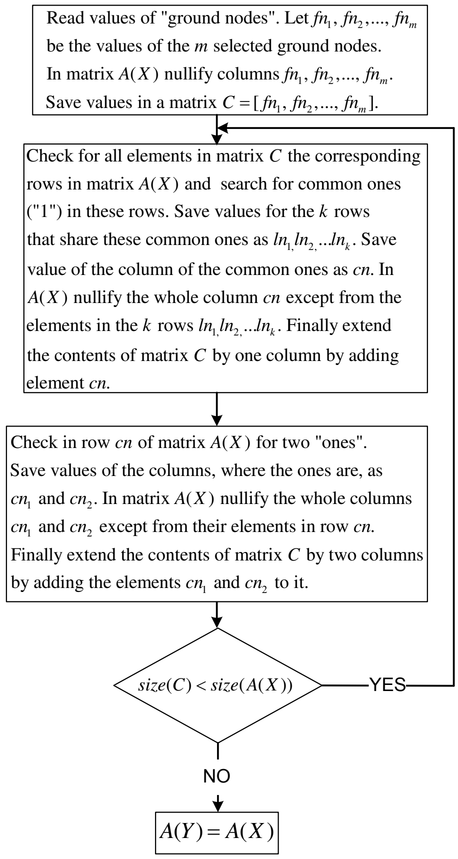

Fig. 6. The block diagram of the whole algorithm, which produces the final matrix $A(Y)$ of the directed graph of the system, given the matrix $A(X)$ and the selected ground-nodes.

- the shape of the structure
- the type of each assembled R-Cell (with 4 or 2 DoFs)
- the relationship between actuators and DoFs within the structure,

which is all the significant information about the final robotic structure in a simple form. An adjacency matrix is also able to provide us with much more information about the inner features of the system, such as the number of *paths* connecting a specific node to another node etc.

Using the structure matrix, graph theory, the DK, IK and DF analysis of our module we are going to develop two examples of assembled robotic structures and exploit their capabilities.

V. Case Studies

In this section, two typical modular SR robotic structures based on the R-Cells modules are introduced to depict the benefits of our concept. For our simulation purposes the R-Cell is a rectangle with a unitary side length. In the first case study a *linear robotic structure* will be used, while in the second one a structure which resembles a *human arm* will be introduced.

A. A linear robotic structure

Consider a case of a linear robotic structure like the one in Fig. 7. This is a structure built with 7 R-Cells. A vertical

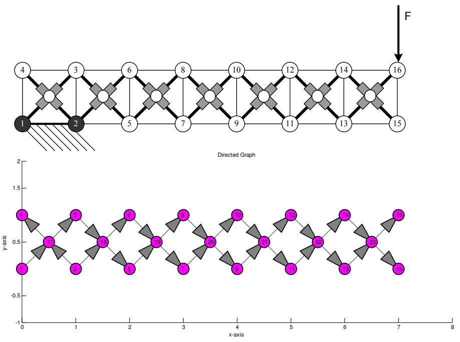

Fig. 7. The robotic structure of case 1. In this case the final robot consists of $n = 7$ R-Cells. The directed graph of this system is also depicted.

force (F) is applied at the far right end. Let $F = 1$ for the simulation purposes. At first the ground nodes are defined, which are in this case the nodes $\{1,2\}$. Fig. 7 also shows the directed graph of this system as it was calculated using the methodology introduced in Fig. 6.

Firstly the DK, IK and DF of the system are calculated. The time needed was 2.6 secs. A system based on the Intel Pentium 4 (3GHz) CPU with 3GB of RAM was used. In the end the Jacobian matrix (J) is at our disposal. When a force F is applied on a specific node i, we are able to calculate the *distribution of the inner forces* τ among the prismatic joints by using the next equation:

$$\tau = J^T(q)\gamma \tag{10}$$

where: τ denotes the $(m \times 1)$ vector of joint forces and γ the $(m \times 1)$ vector of the end-effector (node) forces, while m are the DoFs of the final structure.

In this case study the structure has 16 nodes. 2 of them are fixed because they are ground nodes. Thus, there are 14 free nodes left, which are translated in $m = 28$ DoFs. The distribution of the inner forces by solving eq. 10 is given in Fig. 8 (top left).

To demonstrate the deformation and force handling capabilities of structure composed of R-Cells, we will apply a reactive shape optimization technique, so that the structure will "adaptively" deform, as a reaction to the applied force, in order to better distribute the forces within it's structure. To this extend let us introduce the following potential function:

$$V = \gamma^T J J^T \gamma + \sum_{i=1}^{n} C \left(\frac{1}{(\theta_i - \theta_{\min})} + \frac{1}{(\theta_{\max} - \theta_i)} \right) \tag{11}$$

where $C > 0$ is a constant $\theta_{\min}$ and $\theta_{\max}$ are the minimum and maximum angles allowed for the R-Cell (see figure 5). The first term in the above equation is used to penalize the forces on individual actuators while the second term penalized angles close to the mechanical limits of the system.

If we model the state of our robotic structure using the $m \times 1$ vector $\mathbf{q}$ of prismatic joint's variables we have the following model:

$$\dot{\mathbf{q}} = \mathbf{u}$$

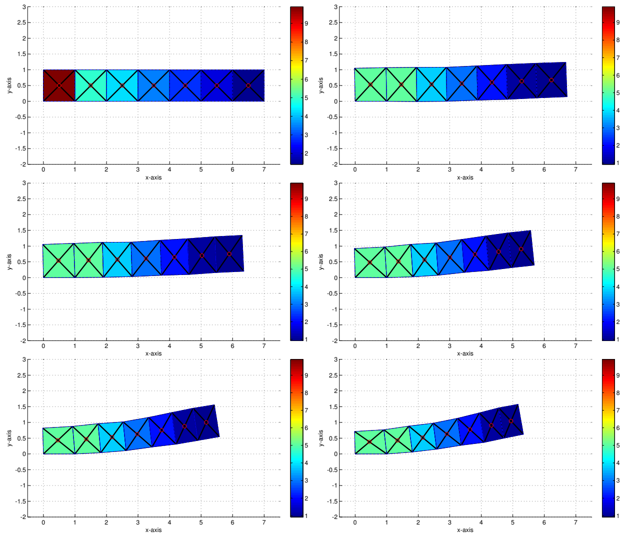

Fig. 8. Snapshots (top left to bottom right - read from left to right) depict the evolution of the structure's shape and distribution of inner forces of the system by using the adaptive shape optimization controller. The maximum value of the forces of each R-Cell is depicted using color coding.

where u is the $m \times 1$ vector of control inputs we wish to apply to our actuators. Using a control law of the form[1]:

$$u_i = \begin{cases} -K\nabla_i V & \begin{cases} q_{\min} < q_i < q_{\max} \\ q_i \leq q_{\min} \wedge (-\nabla_i V > 0) \\ q_i \geq q_{\max} \wedge (\nabla_i V > 0) \end{cases} \\ 0 & \quad elsewhere \end{cases} \tag{12}$$

for $i = 1 \ldots n$ and index i refer's to the i'th element of the corresponding vectors, will enable our structure to adaptively deform in order to relax the load on individual actuators, while making sure that the mechanical limits on the angles and actuators are not violated. As can be seen in Fig. 8 as the mechanical structure is deforming the mechanical loads on the actuators are indeed decreasing. Notice the rapid decrease of the maximum force in the fixed cell. The robotic structure has reacted to the external force and adaptively adjusted it's shape to better withstand the received force.

B. *A robotic arm*

In this case an assembled robotic structure resembling a human arm will be discussed. Consider the situation depicted in Fig. 9. We want to create a structure and apply a force to some point i.e. behind an obstacle. Consequently, we assemble a robotic structure with 18 R-Cells. Ground nodes are in this case the nodes $\{1,2,5,7\}$ of the structure.

The DK, IK and DF of the system are calculated and the time needed was 4.6 secs. Within this time the graphs (directed and undirected) were also built. A system based on the Intel Pentium 4 (3GHz) CPU with 3GB of RAM was used. In this case there are 36 nodes and 64 DoFs in the structure. We would like the "end effector" to apply a force of $F = 1N$ as shown in figure (9,Up), that is split to forces

[1]The stability analysis of the implemented controller is outside the scope of the current paper

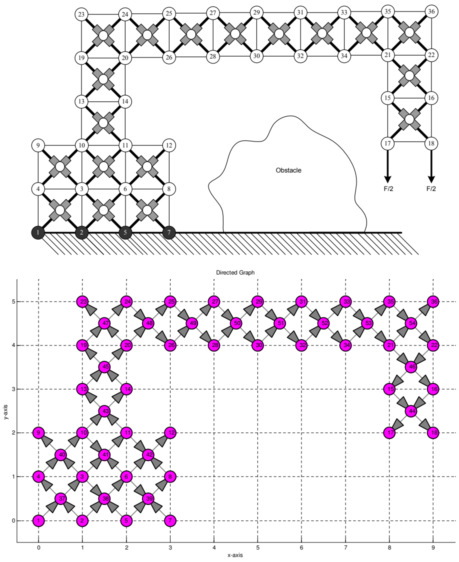

Fig. 9. [Up] The robotic structure of case 2 and the requested force. In this case the final robot consists of $n = 18$ R-Cells. [Down] The corresponding directed graph.

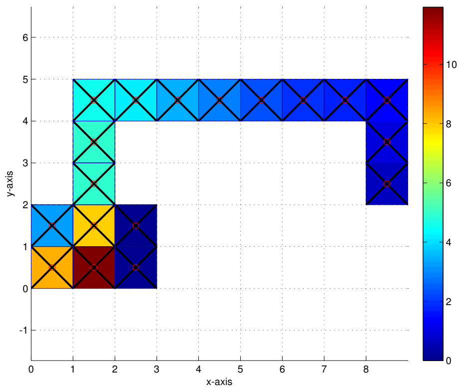

Fig. 10. Inner forces in the structure of the second case study. The mean value of the forces of each R-Cell is calculated and depicted.

of magnitude $F/2$ on each of the nodes 17 and 18. By using eq. (10) we are able to calculate the *distribution of the inner forces* τ among the prismatic joints. The results are shown in Fig. 10.

VI. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

This paper proposes a concept of a module for a homogeneous modular and SR robotic system. The conceptual design and the kinematics analysis were introduced. Structures consisting of R-Cells were demonstrated and their force-handling and deformation capabilities were exploited. One of the most significant capability of our module in comparison with any other state-of-the-art modules is its "deformation" ability. This ability enables the resulting robotic structure to change its shape to some extent in order to apply forces or torques and interact with its environment. Thus, the most novel ability of the proposed system is the ability to change its shape without any cell reconfiguration.

B. Future Works

We currently working on a more detailed mechanical design of the first R-Cell module. A co-operation protocol for a better inner-force distribution among the R-Cell modules is currently being developed. Finally, our research endeavors include investigating the major problem of "*Synthesis*", namely the problem to find the "optimal" shape and cell-configuration in a robotic structure in order to have the desired functionality in the final robot.

REFERENCES

[1] S. Murata, E. Yoshida, H. Kurokawa, K. Tomita and S. Kokaji, "Self-Repairing Mechanical Systems," *Autonomous Robots*, vol. 10, no. 1, pp. 7-21, 2001.

[2] K. Tomita, S. Murata, H. Kurokawa, E. Yoshida, and S. Kokaji, "Self-Assembly and Self-Repair Method for a Distributed Mechanical System," *IEEE Trans. Robot. Autom.*, vol. 15, pp. 1035-1045, Dec. 1999.

[3] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw and S. Homans, "Modular Reconfigurable Robots in Space Applications," *Autonomous Robots*, vol. 14, no. 2-3, pp. 225-237, 2004.

[4] T. Fukuda and S. Nakagawa, "Dynamically Reconfigurable Robotic System," *Proc. IEEE Int. Conf. Robotics and Automation*, vol. 3, pp. 1581-1586, 1988.

[5] T. Fukuda and S. Nakagawa, "Approach to the Dynamically Reconfigurable Robotic System," *J. Intell. Robot. Syst.*, pp.55-72, 1988.

[6] A. Pamecha, C.J. Chiang, D.Stein, and G. Chirikjian, "Design and Implementation of Metamorphic Robots," *Proc. ASME Design Eng. Tech. Conf. Computers Eng. Conf.*, Irvine, CA, 1996.

[7] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita and S. Kokaji, "A 3-D Self-Reconfigurable Structure," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 432-439, 1998.

[8] K. Kotay, D. Rus, M. Vona and C. McGray, "The Self-reconfiguring Robotic Molecule," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 424-431, 1998.

[9] M.W. Jorgensen, E.H. Ostengaard, and H.H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 2068-2073, 2004.

[10] D.J. Christensen, E.H. Ostengaard, and H.H. Lund, "Metamodule Control for the ATRON Self-Reconfigurable Robotic System," *Proc. Intelligent Autonomous Systems (IAS-8)*, pp. 685-692, 2004.

[11] M. Yim, D. Duff, and K. Roufas, "Polybot: A Modular Reconfigurable Robot," *IEEE Int. Conf. Robotics and Automation*, pp. 514-520, 2000.

[12] A. Castano, W-M Shen, and P. Will, "CONRO: Towards Deployable Robots with Inter-Robots Metamorphic Capabilities," *Autonomous Robots*, vol. 8, no. 3, pp. 309-324, 2004.

[13] W.M. Shen, B. Salemi, and P. Will, "Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots," *IEEE Trans. Robot. And Automat.*, vol. 18, no. 5, pp. 700-712, Oct. 2002.

[14] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-Reconfigurable Modular Robotic System," *IEEE/ASME Trans. Mechatron.*, vol. 7, pp. 431-441, Dec. 2002.

[15] E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa, and S. Kokaji, "A Self-Reconfigurable Modular Robot: Reconfiguration Planning and Experiments," *Int. J. Robot. Res.*, vol. 21, no.10-11, pp. 903-915, 2002.

[16] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson, "Evolved and Designed Self-Reproducing Modular Robotics," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 308-319, 2007.

[17] D. Rus and M. Vona, "Crystalline Robots: Self-Reconfiguration with Compressible Unit Modules," *Autonomous Robots*, vol. 10, no. 1, pp. 107-124, 2001.

[18] J. W. Suh, S. B. Homans and M. Yim, "Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robotics," *IEEE Int. Conf. Robotics and Automation*, pp. 4095-4101, 2002.

[19] C. Godsil and G. Royle, "Algebraic Graph Theory," Springer, 2001,

[20] L Sciavicco and B Siciliano, "MODELING AND CONTROL OF ROBOT MANIPULATORS" McGraw Hill, New York, 1996.