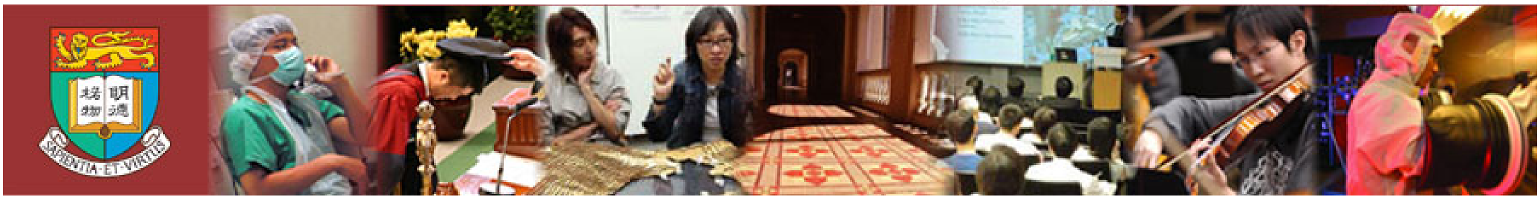


Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Topological Representation and Analysis Method for Multi-port and Multi-orientation Docking Modular Robots

Albert Ko* T. L. Lau H. Y. K. Lau

The University of Hong Kong
Pokfulam road, Hong Kong SAR
Intelligent Systems Laboratory
E-mails*: aux1496@hkusua.hku.hk
Website*: www.air-lab.net

Abstract

For MSR Robots to successfully configure from one configuration into another, the control system must be able to visualize the current structure of the robot, which cannot be done without appropriate information about each module's docking status. Although the type of information required to visualize the structure of a MSR robot differs with the physical design of the modules, there are essential information that are commonly required, such as docking orientation and identity of neighboring modules. This paper presents a novel multi-port and multi-orientation modular robot, and a representation method that can uniquely represent the geometric structure of a group of connected modules and to analyze the number of "reconfigurable DOF" within the structure. The proposed method uses labeled planar graphs and incidence matrices to describe the docking status of the modules within the structure, which helps to effectively encode the data in computer understandable expressions. In addition to the work in configuration analysis, an innovative mechanism for detecting the orientation of each docking port is also presented.

Keywords

Self-Reconfigurable Robots, Modular Robots, Graph Theory, Topological Representative, Mechatronic Design.

1. Introduction

Modular Self-Reconfigurable (MSR) robots [1][2][12][13][16] are robots made up of many identical but independent mechatronic modules that can be connected and disconnected autonomously and to rearrange into different structures that can facilitate the robot to complete its tasks more effectively. Most MSR robots are designed as a self-contained unit equipped with its own processor to control the module's movement and to facilitate communication with neighboring modules.

MSR robots' ability in self-reconfiguration makes them particularly useful for applications in unstructured, remote and hazardous environment such as deep sea and space exploration, urban rescue, and military intelligence. Since all modules are identical, if a module in a system is damaged; the robot can simply discard the damaged module and quickly replace it with another one connected nearby. This functionality gives MSR robots a distinctive advantage over conventional robots in repairing itself while far from home on a mission.

In spite of all the advantages MSR robots has to offer, there are many challenges to overcome before these robots can have practical applications outside of research. One of the most obvious challenges comes directly from the greatest benefit of MSR robots – Self-Reconfigurations. In quest of a truly self-reconfigurable system, disregarding the use of distributed or centralized control system, a precise understanding of the robot's topological structure is essential. Since MSR robots are groups of connected modules; to visualize the structure of the robot, one must begin with the collection of individual module's docking status and represent them systematically for analysis purposes. This paper proposes a method that uses simple planar graphs and incidence matrices to represent the docking status for any modules with *multi-port* and *multi-orientation* docking capacities. The benefit of using graphs and matrices for representation is that the graphs can be constructed intuitively according to the physical connections of the modules, and by employing basic graph theory, the planar graph can transform into a corresponding matrix for different types of analysis, such as enumerating possible configurations and for analyzing the re-configurability of certain structures.

Section 2 begins with presenting the mechatronic design of ModuKnight, a newly developed homogeneous MSR robot, and go on to describe a simple but novel mechanism, the Hardware Orientation Detecting System (HODS), for detecting the docking orientation of a mutual joint between two modules.

Section 3 gives details of the graphical and mathematical representation methods based on Graph Theory and presents an analysis method that helps to determine whether a given structure contains sufficient Degree-Of-Freedoms for self-reconfigurations by extracting information from the incidence matrix.

2. The Hardware Orientation Detecting System (HODS)

This section focuses in describing the design, implementation and motive of the HODS. In order to make the picture complete, this section will first present the general design of ModuKnight, the MSR robot that facilitated this work. Next, an overview of difficulties raised by Multi-port and Multi-orientation systems will be presented before addressing the problem using HODS.

2.1 Mechatronic Design

ModuKnight is a *Chain-type Homogeneous Asymmetric* MSR robot; it is designed with simplicity and light weight in mind. The purpose of the robot is to provide a test-bed for research in reconfiguration and distributed motion control, therefore the casing of the module is designed to allow easy assembly and disassembly, also the head and tail components are all interchangeable. In addition to modularity, the module has all electronic parts placed near to the docking ports for easy maintenance and the more delicate microcontroller is hidden behind the three-way docking port.

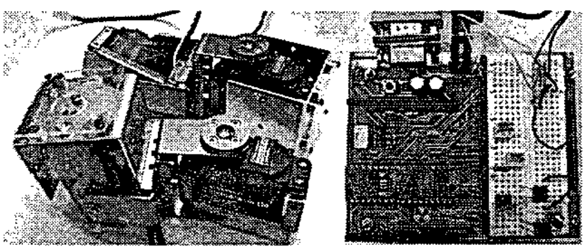


Figure 1: The Second prototype of Moduknight and the BasicATOM development board.

The basic module of ModuKnight contains three sections, the head, the tail and the main body that connects them together. Two high torque (13.0 kg.cm) servo motors connect the head and the tail to the main body and give the module two DOF in the pitch and yaw directions. The module has four neutral docking ports, one in the head section and three in the tail. Each docking port contains an infrared-pair for communications, and tactile sensors for detecting docking orientations and connection status.

While the robot was designed for distribution control, each module is equipped with its own brain for data processing and logic control. ModuKnight uses a self-contained microcontroller, BasicATOM40, which has 32 I/O pins, 14k programming space and can process 33,000 instructions per second. The robot is currently being configured in the form of a hyper-redundant manipulator to test a control theory inspired by the adaptive immune system [6].

2.2 Multi-port Docking Systems

MSR robots can be divided into Heterogeneous [9] and Homogeneous [10] types. As the name suggests, Heterogeneous systems are form of more than one type of modules, whereas a Homogeneous system has only one type of module. If a homogeneous system is to have more than two modules, the modules must have more than one docking port to produce a connected system. A system that contains modules with more than one docking port is a *Multi-port System*. Since each MSR robot has its own design of module, some modules that have docking ports located *Symmetrically* in respect to the joint(s) (DOF), hence, certain connection may produce identical kinematic structure and does not affect the motion freedom of the module [11]. However, some Multi-port modules were designed with docking ports located *Asymmetrically* in respect to the joint(s) [2], (including the ModuKnight presented in this paper, therefore which port to dock determines not only the topological structure of the robot, but the motion freedom of the final structure as well.

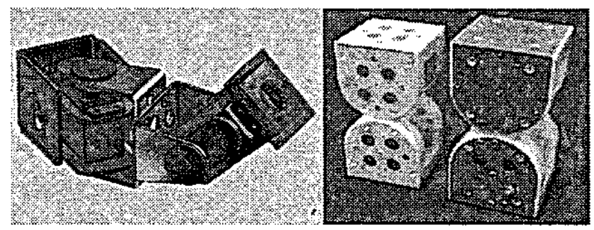


Figure 2: On the left is an asymmetrical module, the ModuKnight, and on the right is a symmetrical module, MTRAN [9].

2.3 Multi-Orientation Docking Systems

Another issue that affects both "Symmetrical" and "Asymmetrical" systems is the docking orientation between two modules. If a docking port can be connected with another in more than one orientation, then it is a Multi-orientation System. This contributes to the fact that if two modules both having only one degree-of-freedom, the docking orientation between them determines if they will produce a system with 2 DOF in the same plane or 2 DOF in two different planes perpendicular to one another.

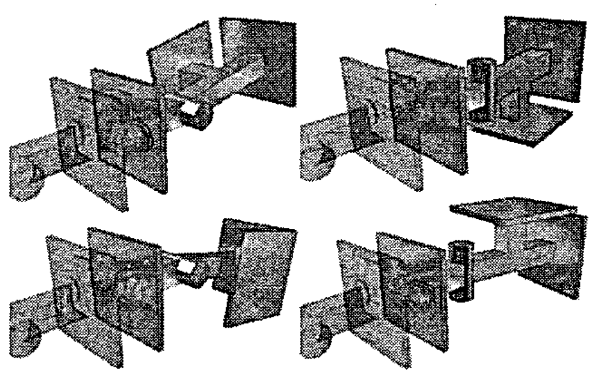


Figure 3: Multi-orientation Docking

In order to determine the docking orientations between two docking ports, a simple mechanism involving only low-cost tactile switches on each docking face was developed. In this paper ModuKnight is being used to illustrate the *Hardware Orientation Detecting Systems* (HODS), but the concept is generally useful for systems with a finite number of possible docking orientations.

ModuKnight has four square docking ports; each can be docked to another in four different orientations. The docking port has two micro tactile switches on the surface, one represented with a circle and the other with a square. When two docking ports of two modules dock together, the switches will only return a contact signal if they are pressing against a switch on the other port, otherwise they will return a null signal. Figure 4 shows the four possible docking orientations for ModuKnight. In mode "A", the circular switches on both ports with return a "contact" signal because they are pressing against each other, but the square switches will return a "null" signal because there is no corresponding switch to press against. Hence, both modules will receive an "On" from the circular switch and a "Null" from the square switch. By collecting feedbacks from the switches, the docking orientation can easily be analyzed using a look up table as shown in Table 1. Note that the circular and square switches are for illustration purpose only, they can be of identical shape as long as they have separate feedback channels for the processor to analyze the orientation.

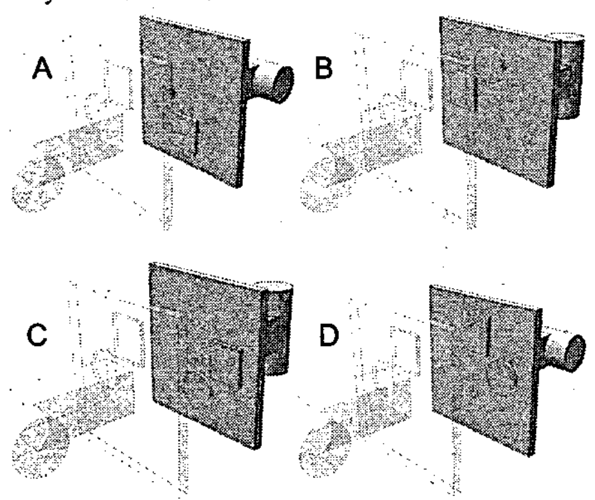


Figure 4: The Left-module is illustrated in transparent color and the Right-module is illustrated in light gray. The gray module rotates counter-clockwise to demonstrate the four possible docking orientations. Note that the locations of the two tactile sensors are identically placed in respect to the surface of the docking port.

The HODS offers two major advantages in mechatronic implementation and distributed control. Firstly, for n equals the number of possible docking orientation, the system requires no more than $n + 1$ switches to determine the docking orientation. Secondly, the HODS returns identical feedbacks to the two docked modules, therefore both modules will have exactly the same idea about their mutual joint. This feature is particularly useful if the robot is to be controlled in a distributed manner.

	Left Module (transparent)		Right Module (light gray)	
Mode	Circle	Square	Circle	Square
A	On	Null	On	Null
B	On	On	On	On
C	Null	Null	Null	Null
D	Null	On	Null	On

Table 1: Summary of signal feedbacks from the tactile sensors of the two docking ports.

3. Structure Representations

This section focuses in docking status representation for homogeneous MSR robots with multi-port and multi-orientation docking capacities, and the analysis of self-reconfigurability by determining the number of available DOF within the same plane.

3.1 Graphical Representations

Traditionally when representing a network of computers, one often uses a vertex to denote the computer and an edge to denote the connection between computers, for MSR robots, unfortunately, this method does not show which ports were involved in the docking and what the orientation of the docking was. Although much effort have been put to understand MSR robots from different directions. Farritor et al. [7] demonstrated hierarchical selection process can reduce the search space for possible structure configurations, Chiang et al. [5] uses similarity metrics to generate motion path for planar reconfiguration, Zhang et al. [17] studied the use of constraint based control framework to control the locomotion of a MSR robot in different configurations. However, only few have focused in the representation and analysis of docking status at modular level, Chen et at. [4] introduced the Assembly Incidence Matrix (AIM) that models the connectivity of dynamic structures but does not work with asymmetric systems, Fei et al. [8] analyzed the type of joints in dynamic structures but lack of support for multi-port docking systems, Castano et al. [3] had adapted the AIM method and carried it further to make it work on multi-port and multi-orientation systems, but the digraph is only applicable to multi-orientation modules with two different docking orientations only.

In this section a new representation method using labeled planar graphs and incidence matrices is presented. Unlike the work mentioned above, this new method works on both symmetrical and asymmetrical systems with multi-port and multi-orientation docking capacities. The basic module of ModuKnight has four docking ports each is capable of docking in four different orientations. Three of the docking ports are located in the base unit and the stand alone docking port is located in the head unit asymmetrically in respect to the two joints; the head and base unit of the module are connected by two DOF that lay in two different planes perpendicular to each other. The module can be represented with a unit graph as shown in Figure 5, the four labeled vertices denotes the four docking ports and the six edges denote the interconnections between the vertices. The number 9317 is an ID uniquely generated for each module.

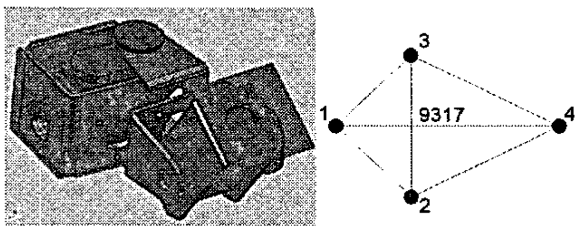


Figure 5: The Unit Graph of ModuKnight. Four vertices denote the docking ports of the module and 9317 is the module ID.

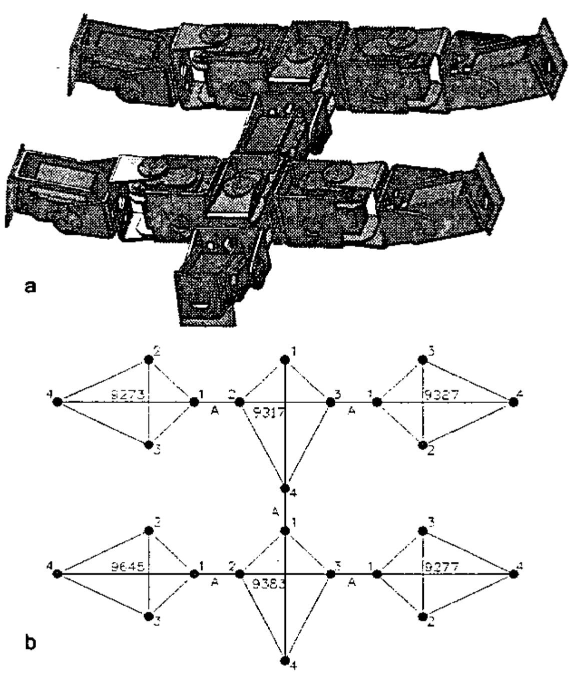


Figure 6: The six-module ModuKnight is represented by a labeled planar graph consisting of six connected unit graphs.

The Unit Graph in Figure 5 can be further developed to represent more complex structures. Shown in Figure 6 is a six-module structure, the edge joining modules 9273 and 9317 is labeled with an alphabet "A" denoting the two modules are docked in mode "A" (refer to Table 1). Note that the graph concerns only the topological but not the kinematics structure of the robot, therefore the planar graph is constructed by rotating the unit graphs above the paper plane only, flipping in any direction is forbidden.

3.2 Matrix Representations

This section describes the representation of the graph in Figure 6 with matrices. Unlike others previous work [3][4], this paper uses Incidence Matrix instead of Adjacency Matrix to represent the planar graph. The advantage of Incidence Matrix is the smaller dimension it produces. Since each docking joint involves two and only two docking ports, therefore the maximum number of docking joints (***i*** row) must be equal to or less than the number of docking ports (***j*** column), hence the resulting incidence matrix is always smaller than a square Adjacency Matrix that has equal number of docking ports in both ***i*** and ***j*** directions.

Conventionally an Incidence Matrix is filled with Ones and Zeros only; however for modules with multi-orientation docking capacity, a more descriptive system is needed to represent the different orientations. The structure in Figure 6 is represented by the Incidence Matrix M(G) in Figure 7. The eight-digit Joint-IDs are labeled outside of the matrix on the left for illustration purpose, this column is the edge set E(G) = $\{e_1, \dots, e_m\}$, where e_n is the Joint ID, which is generated by lining up the two involved module IDs of each joint, in this example there are 5 joints in total. The vertex set V(G) = $\{v_1, \dots, v_n\}$ is illustrated at the top of the matrix, representing the Docking Ports ***l*** in each Module ***k***. The dimension of the matrix is $m \times n$ where $n = kl$ and m is the number of joints within the structure.

Joint ID					9277 1	2	3	4																		
9273-9317	A	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9277-9383	0	0	0	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0
9317-9327	0	0	0	0	0	0	0	0	0	0	A	0	A	0	0	0	0	0	0	0	0	0	0	0	0	0
9317-9383	0	0	0	0	0	0	0	0	0	0	0	A	0	0	0	0	0	0	A	0	0	0	0	0	0	0
9383-9645	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A	0	0	A	0	0	0

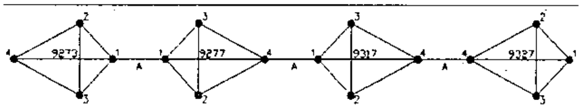
Figure 7: This Incidence Matrix is a representation of the structure in Figure 6. The numbers inside the dashed rectangle indicate that docking port #1 of module 9277 is connected to module 9283 with a type "A" docking orientation.

In general, entry $m_{i,j}$ of A, B, C, or D (denoting the docking orientation) is made if e_j is connected to v_i. For example, Docking Port #1 of module 9277, e_5, is connected with joint 9277-9383, v_2, with type "A" orientation, then an entry at $m_{2,5}$ will be made.

3.3 Analyzing for Unit Dexterity

Chain-type MSR robots are those that can perform locomotion without reconfiguration [15]. This type of robot often requires a minimum number of modules to perform self-reconfiguration; this minimum number is referred to as *Unit Dexterity*. To be precise, self-reconfiguration in this section implies all actions that can create a new docking joint without producing a "detached" module during the process. The unit dexterity of a specific robot type can be easily determined by modeling the modules in CAD software, or by analyzing the type and number of joints in the modules.

This section introduces a mathematical method for analyzing the matrix **M**(G) of small structures to see whether they contain unit dexterity for self-reconfiguration. Figure 8 shows a simple four-module structure of ModuKnight and its corresponding Incidence Matrix. Since ModuKnight has one degree-of-freedom perpendicular to the module's center line on all four sides, therefore whenever docking port 1 or 4 is connected to docking port 1 or 4 of another module, a DOF contribution is made. ModuKnight has unit dexterity of four, meaning at least four modules connected in head to tail format is needed to perform self-reconfiguration.



$$M(G)=\begin{bmatrix} A&0&0&0&A&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&A&A&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&A&0&0&0&A \end{bmatrix}$$

Figure 8: The structure contains three connections that involve docking ports 1 and 4; therefore the structure has four DOF.

This method extract information from the Incidence Matrix **M**(G) by multiplying it to an identity matrix $\mathbf{I}_n$. The resulting matrix $\mathbf{D}_n$ can then be used for analysis purpose.

The matrix M(G) in Figure 8 is being transformed into matrix $\mathbf{D}_n$ using the formula.

$$\mathbf{M(G)_n \times I_n = D_n}$$

M(G) is transformed to $M(G)_n$ by substituting all the alphabets that denote the docking orientations to "1" regardless of their docking types. This is because the final matrix $\mathbf{D}_n$ concerns only with the number of dockings in the correct port rather than the orientation. The matrix $\mathbf{I}_n$ is responsible for extracting information from $M(G)_n$ and it is basically a diagonal matrix that repeats itself vertically every l rows. For l is the number of Docking Ports (DP) in each module, the dimension of the matrix $\mathbf{I}_n$ is $kl \times l$, where k is the number of modules within the structure.

		Port IDs 1	2	3	4
Joint IDs	9273-9277	2	0	0	0
	9277-9317	1	0	0	1
	9317-9327	1	0	0	1

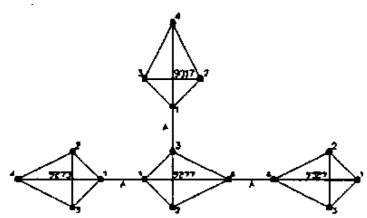
Figure 9: Matrix, $\mathbf{D}_n$, shows how many ports are involved in each joint. For example $\mathbf{m}_{1,1}$ indicates there are two DP #1 (docking port) involved in joint 9273-9277.

$$\begin{bmatrix} 1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1 \end{bmatrix} \times \begin{bmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix} = \begin{bmatrix} 2&0&0&0 \\ 1&0&0&1 \\ 1&0&0&1 \end{bmatrix}$$

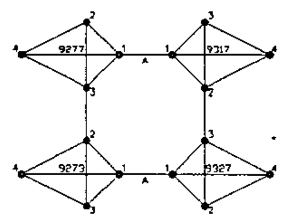
After the operation, the resulting matrix $\mathbf{D}_n$ is produced. Figure 9 illustrates the notation of $\mathbf{D}_n$ matrix. The matrix $\mathbf{D}_n$ can be analyzed for unit dexterity with a set of rules generated by examining the mechanical limitations of the robot:

1. Four DOF in the same line of motion is needed for self-reconfiguration.
2. Each joint contains two and only two docking ports.
3. No DOF contribution if docked with DP #2 and #3.
4. The docking port ID of each module cannot be repeated. (i.e. cannot have two docking port #1 in a module)

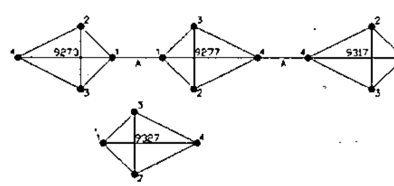
The matrix of Figure 9 complies with all the rules. First, there are three docking joints involving six docking port #1 and #4, meaning all four modules are connected in head to tail format. Second, none of the rows contain more than two docking ports. Third, no DOF was wasted as none of the joints involved DPs #2 and #3. Forth, the sum of each column is equal to or less than the total number of modules in the structure.



$$D_n=\begin{bmatrix} 2&0&0&0 \\ 1&0&1&0 \\ 0&0&0&2 \end{bmatrix}$$



$$D_n=\begin{bmatrix} 0&1&1&0 \\ 2&0&0&0 \\ 0&1&1&0 \\ 2&0&0&0 \end{bmatrix}$$



$$D_n=\begin{bmatrix} 2&0&0&0 \\ 1&0&0&1 \\ 2&0&0&0 \end{bmatrix}$$

Figure 10: Illustrations of different structures with their corresponding incidence matrices.

Figure 10 shows some non-self-reconfigurable structures along with their $\mathbf{D}_n$ matrix. Notice the top structure lost one DOF because the third column indicates one of the docking joint involved DP #3, which cannot utilize the module's motion freedom. The middle structure cannot perform self-reconfiguration neither, because the second and third column of $\mathbf{D}_n$ indicated the two docking joints were formed by DP #2 and DP #3, hence no DOF in that direction. Finally, the bottom structure was generated to demonstrate rule number four, notice the first column sum up to five, which is impossible because four modules cannot have five DP #1. Therefore the matrix results in a disconnected graph. These examples show the extracted information in $\mathbf{M}(G)_n$ can help to determine whether a four-module structure is self-reconfigurable, and provides a way to eliminate disconnected structures when randomly generating a four-module structure from an incidence matrix.

The analysis method described in this section has been tested on many four-module structures and worked well. However when the system increases to a five or six-module structure, it becomes difficult to tell if the "useful" connections were connected along the same motion plane. For example, if two extra modules were connected with DP #1 and 4 to the DP #4 of module 9273 in Figure 10a, then the structure will have enough (five) "useful" DOF for self-reconfiguration. However, if the two modules were connected vertically above DP #4 of module 9317, then the structure still does not have unit dexterity for self-reconfiguration.

At this moment the analysis method is only applicable to a structure with four or more modules, with the aid of a planar graph. Our ambition is, therefore, to search for unit dexterity within a structure using the matrix operations only. A more advanced method under development is to extract other matrices from the incidence matrix, to analyze the branching structure that dictates the self-reconfigurability of the structure.

4. Conclusion

The focus of this paper is the representation of multi-port and multi-orientation homogeneous MSR robots, which was presented separately in two streams, hardware detection and graphical representation.

In hardware detection, we presented the Hardware Orientation Detecting System (HODS) that uses simple tactile sensors to detect the docking orientation of multi-orientation modules. The greatest benefit of the HODS is not only it is simple to implement, but it allows the two connected modules to have the same "feeling" about their mutual joint. This attribute is particularly beneficial when the robot is controlled with a distributed system.

In graphical representation, we presented a complete method that is capable of representing MSR robots with multi-port and multi-orientation docking capacities. Different from others, our method is applicable to modules with any number of possible docking orientations, and the incidence matrices generated are much smaller in size.

Our future goal is to build a complete and general analysis method to determine whether a particular topological structure contains Unit Dexterity for self-reconfiguration using only matrix operations.

References

[1] H. Bojinov, A. Casal, and T. Hogg, "Emergent Structures in Modular Self-reconfigurable Robots," In Proc. of the *Intl. Conf. on Robotics & Automation*, Vol. 2, pp. 1734-1741, IEEE, 2000.

[2] A. Castano, and P. Will, "Mechanical Design of a Module for Reconfigurable Robots," In Proc. of the *Intl. Conf. on Intelligent Robotics & Systems*, Vol. 3, pp. 2203-2209, IEEE, 2000.

[3] A. Castano, and P. Will, "Representing and Discovering the Configuration of Conro Robots," In Proc. of the *Intl. Conf. on Robotics and Automation*, Vol. 4, pp. 3503-3509, IEEE 2001.

[4] I. Chen, and J. Burdick, "Enumerating the Non-Isomorphic Assembly Configurations of Modular Robotic Systems," In Proc. of the *Intl. Conf. on Intelligent Robots and Systems*, Vol. 3, pp. 1985-1992, IEEE 1993.

[5] C. Chiang, and G. Chirikjian, "Modular Robot Motion Planning Using Similarity Metrics," Autonomous Robots, Vol. 10, pp. 91-106, 2001.

[6] D. Dasgupta, "Artificial Immune Systems and Their Applications. Springer," Germany, 1999.

[7] S. Farritor, and S. Dubowsky, "On Modular Design of Field Robotic Systems. Autonomous Robots," Vol. 10, pp. 57-65, 2001.

[8] Y. Fei, X. Zhao, and W. Xu, "Kinematics and Dynamics of Reconfigurable Modular Robots," In Proc. of the *Intl. Conf. on Systems, Man, and Cybernetics*, Vol 4, pp. 3335-3341, IEEE 1998.

[9] T. Fukuda, and G. Iritani, "Evolutional and Self-Organizing Robots-Artificial Life in robotics," In Proc. of the *IEEE Symposium on Emerging Technologies & Factory Automation*, pp. 10-19, IEEE, 1994.

[10] A. Kamimura, S. Murata, E. Yoshida et al., "Self-Reconfigurable Modular Robot – Experiments on Reconfiguration and Locomotion," In Proc. of the *Intl. Conf. on Intelligent Robotics & Systems*, Vol. 1, pp. 606-612, IEEE, 2001.

[11] D. Rus, and M. Vona, "A Physical Implementation of the Self-reconfiguring Crystalline Robot," In Proc. of the *Intl. Conf. on Robotics & Automation*, Vol. 2, pp. 24-28, IEEE, 2000.

[12] J. Suh, S. Homans, and M. Yim, "Telecubes: Mechanical Design of a Module for Self-Reconfigurable Robotics," In Proc. of the *Intl. conf. on Robotics & Automation*, Vol. 4, pp. 4095-4101, IEEE, 2002.

[13] C. Unsal, and P.K. Khosla, "Mechatronic Design of a Modular Self-Reconfiguring Robotic System," In Proc. of the *Intl. Conf. on Robotics & Automation*, Vol. 2, pp. 1742-1747, IEEE, 2000.

[14] C. Unsal, and P.K. Khosla, "A Multi-Layered Planner for Self-Reconfiguration of a Uniform Group of I-Cube Modules," In Proc. of the *Intl. Conf. on intelligent Robots and Systems*, Vol. 1, pp. 598-605, IEEE 2001.

[15] M. Yim, "Locomotion with a Unit-Modular Reconfigurable Robot," Ph.D Thesis. 1994.

[16] M. Yim, D.G. Duff, and K.D. Roufas, "PolyBot: a Modular Reconfigurable Robot," In Proc. of the *Intl. Conf. on Robotics & Automation*, Vol. 1, pp. 514-520, IEEE, 2000.

[17] Y. Zhang, M. Fromherz, L. Crawford, et. al., "A General Constraint-Based Control Framework with Examples in Modular Self-Reconfigurable Robots," In Proc. of the Intl. Conf. on Intelligent Robots and System, pp. 2163-2168, IEEE 2002.