The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Representation and Shape Estimation of Odin, a Parallel Under-actuated Modular Robot

Andreas Lyder, Henrik Gordon Petersen, Kasper Stoy

***Abstract*— To understand the capabilities and behavior of a robot it is important to have knowledge about its physical structure and how its actuators control its shape. In this paper we analyze the kinematics and develop a general representation of a configuration of the heterogeneous modular robot Odin. The basics of estimating the shape of the Odin robot is presented, which leads the way for further research on the Odin robot and similar robots. We present an example of how to represent and estimate the shape of a tetrahedron configuration with various types of modules. We conclude that this representation can be used to find the physical constraints of the Odin robot and estimate the shape of a configuration.**

I. Introduction

Modular robots consists of a number of robotic modules which can be connected in various configurations. By reconfiguring the modules, the robot can exist in a variety of different physical configurations. To understand the capabilities and behavior of a robot it is important to have knowledge about its configuration and how its actuators control its shape. In this paper we develop a general representation of the configuration of the heterogeneous modular robot Odin. As an example, we use this representation to describe the physical structure of the Odin robot, and to find its shape based on the position of its actuators.

One vision of modular robots is to create a robotic set of building blocks which can be brought to different environments and configured to solve a variety of tasks. Some example scenarios are search and rescue [1], space exploration and manipulation [2], or assistance for unexpected maintenance [3]. Common for all these scenarios is that their environments are unknown, and the tasks that needs to be performed cannot be prepared for in advance. Another vision is to create physical displays [4]. Physical displays are able to project a 3-dimensional shape which the user can feel and maybe manipulate. This can be used in a design process and to broadcast and share threedimensional objects.

A robot arm is an example of a conventional robot which is built to perform one task, and perform that task really well. A robot arm is typically a serial chain of 1-degree of freedom rotational joints and the position and orientation of the tip of the arm depends on the angle of each joint. Once this robot arm has been built, its physical structure cannot be changed, and its behavior can only be changed by changing the software controlling the robot. A modular robot may not be able to perform that single task just as well as the robot arm. On the other hand, it can be reconfigured into different physical shapes and be fitted to different environments where it can do a variety of tasks. So, unlike a robot arm, the Odin robot may change its physical structure and behavior each time it is reconfigured.

Andreas Lyder, Henrik Gordon Petersen, and Kasper Stoy are with The Maersk Mc-Kinney Moller Institute, University of Southern Denmark, 5230 Odense M, Denmark – E-mail: {lyder, hgp, kaspers}@mmmi.sdu.dk

Fig. 1. An example of the Odin robot in a tetrahedron configuration with a rigid base of battery modules and a flexible top of telescoping modules. A wireless module is connected to the top node.

In this paper we present a set of representation matrices for the Odin modular robot which describes the configuration of its different types of modules. We conclude that these representation matrices can be used to find the physical constraints of the Odin robot and estimate the shape of a configuration.

II. The Odin Modular Robot

The Odin robot is a heterogeneous modular robot [3], [5], [6]. Fig. 1 shows an example configuration with three battery modules equipped with rigid connectors creating a rigid triangular base at the bottom. Three telescoping links equipped with flexible connectors are placed on the base creating a tetrahedral structure. And finally a wireless controller is placed on top to enable the robot to be remote controlled. Unlike homogeneous modular robots [7], [8], [9], the Odin robot consists of different types of modules with different functionality. This enables us to make modules with very simple functionality, since each module does not have to have power, actuation and sensing, but the functionality can be divided among different modules. We define four different types of modules: actuation, power, sensor, and passive modules. Actuation modules provide actuation such as linear or rotational actuation. Power modules energize the robot through a global power bus, which runs through all modules. A power module can be a battery or maybe a solar panel. A sensor module provides sensor inputs to gain knowledge about the environment. Examples of sensors could be a camera, a GPS unit, a compass, an accelerometer, and many more. Last but not least, some modules does not have any active functionality but provides important structural features to the robot. We call these modules, passive modules.

The Odin robot is based on a link and node structure, where link modules interconnect through node modules. The structure is similar to that of the Tetrobot [10]. The node modules are the most important passive modules since they define the lattice of the Odin robot. The links connected to a node are electrically interconnected with a power bus and a RS-485 communication bus. For this paper we present a CCP node which connects links in a cubic-closed packed (CCP) lattice. Each CCP node has 12 connection points arranged as the faces of a rhombic dodecahedron.

A link consists of a body and typically two connectors, one at each end of the body. Different types of connectors can be used to connect a link to a node. In this paper we present a rigid connector, RC, with no degrees of freedom, and a flexible connector, FC, with yaw-pitch-roll degrees of freedom. The flexible connector has a spring wrapped around a ball-and-socket joint to create a stable but springy structure. We present two different bodies for a link, a battery, BL, a telescoping linear actuator, TL, and a wireless controller, WL. The wireless controller is single-ended and has only a single connector at one end, and can therefore not be placed between two nodes. The mechanical design of the Odin robot is described in more detail in [6].

III. Kinematic Analysis

To create a representation of the Odin robot we need to understand its kinematic structure. The Odin robot is designed to be configured in a lattice, which means that it classifies as a parallel robot. Most parallel robots are characterized by having the same number of actuators as degrees of freedom (we refer to these as fully actuated). The kinematic structure of fully actuated parallel robots as a function of actuator positions is generally locally unique, and when fixing actuator positions, high stiffness and accuracy can be achieved. Various types of fully actuated parallel robots are in use today. The most well-known are the Stewart-Gough Platforms [11] and the Flexpicker robot from ABB. If there are more degrees of freedom than actuators, the parallel robot will be underactuated and the kinematic structure will no longer be locally unique.

Unlike these systems the goal of the Odin modular robot is not to achieve high stiffness. Rather, we typically wish the robot to be underactuated so that it can passively adapt its shape to e.g. obstacles in the environment. The Odin robot is thus made from an arbitrary number of nodes, interconnected using passive or active links with flexible or rigid connectors. The passive degrees of freedom can be determined by how many nodes and how many flexible connectors are part of a configuration. If all the connectors are flexible, each node, except from the base node, has six degrees of freedom , which means that a configuration of n nodes has up to $6(n-1)$ degrees of freedom. The active degrees of freedom is determined by the number of actuated links m. We assume that each actuated link is independent and has one degree of freedom, which for typical connection topologies of nodes and actuators suggests that the total number of degrees of freedom will be significantly higher than the number of actuators. However, it should be mentioned that completely resolving the type of kinematics for arbitrary parallel structures is computational intractable. See [12] for a detailed discussion. It is not the purpose of this paper to make a general kinematic study of different Odin topologies, but rather to illustrate how we can represent Odin configurations and compute equilibrium shapes. For this purpose, we study the tetrahedron example in Fig. 1, where it is easy to show that there is a total of 6 degrees of freedom (position and orientation of the top node) of which 3 are actuated.

IV. Representing Odin

One of the biggest advantages of a modular robot is that it can be put together in several different configurations. However, it also means that each time the configuration of the robot is changed we must change the models for analyzing and controlling the robot. To be able to generate these models we define three configuration matrices which represent the interconnection of modules, the type of connectors, and the type of links that are part of a robot. The configuration matrices are also useful for being able to reproduce robots in experiments and in computer simulations. Common for all matrices is that their size depends on the number of nodes n, in a configuration. This means that the size of the matrices is $n \times n$. In general, the elements of the matrices are tuples, but here we will for simplicity assume that there is at most one link between each pair of nodes and at most one single-ended link to each node. Thus, all connections are unique and we can replace the tuples by single elements.

A. Interconnection Matrix

To represent the interconnection of modules we define an interconnection matrix, IM. Each element in the interconnection matrix represents the connection face on node i which connects node j. This means that a link connected between node i and j is connected to node i on connection face $IM_{i,j}$, and to node j on connection face $IM_{j,i}$. If an element $IM_{i,j}$ reads 0 it means that there is no link connected between node i and j. A single-ended link can only connect to one node, for example the wireless link which has only one connector. These links are represented by the elements in the diagonal $IM_{i,i}$, however, they do not affect the interaction between nodes.

For convenience we have chosen to number the connection faces on the CCP Node from 1 to 12 so that the sum of two opposite faces is always 13. The numbering can be derived from Table I. When configuring the robot the orientation of all the nodes must remain the same to conform with the CCP lattice structure. This means that if a link is connected to node i on connection point 1, it is connected to node j on connection point 12. Due to this, once the elements above the diagonal is known, the elements below the diagonal is also known. This may, however, not be the case for future types of nodes. If we look at the example configuration on Fig. 1, the interconnection matrix will look like:

$$IM_{Tetrahedron} = \begin{pmatrix} 0 & 1 & 2 & 4 \\ 12 & 0 & 3 & 5 \\ 11 & 10 & 0 & 6 \\ 9 & 8 & 7 & 6 \end{pmatrix} \quad (1)$$

B. Connector Matrix

The type of connector used to connect a link to a node may vary. To represent which connectors and where they are used in a configuration we define a connector matrix, CM. Similar to the interconnection matrix, an element in the connection matrix, $CM_{i,j}$, represents the type of connector used on node i to connect a link between node i and j. Also, if a link is connected only to node i the connector is represented by the element $CM_{i,i}$.

What type of connectors that is used to interconnect modules is vital for analyzing the flexibility and deformation of the robot. In this paper we present two different types of connectors, a flexible connector with passive but springy yaw-pitch-roll degrees of freedom, and a rigid connector with no flexibility. In the tetrahedron example on Fig. 1 we have made a triangular base interconnected by links equipped with rigid connectors. The three links connecting the top node are equipped with flexible connectors. In this case the connector matrix look like:

$$CM_{Example} = \begin{pmatrix} 0 & RC & RC & FC \\ RC & 0 & RC & FC \\ RC & RC & 0 & FC \\ FC & FC & FC & FC \end{pmatrix} \quad (2)$$

where FC denotes a flexible connector, RC denotes a rigid connector, and 0 means that there is no connection between node i and j, as described in IM.

C. Link Matrix

The link matrix, LM, represents what type of links are part of a robot and where they are connected. The links represented in the diagonal is only connected to node i, and the links represented above the diagonal connecting node i to j are the same as the links connecting node j to i. Therefore, only the elements in and above the diagonal are relevant.

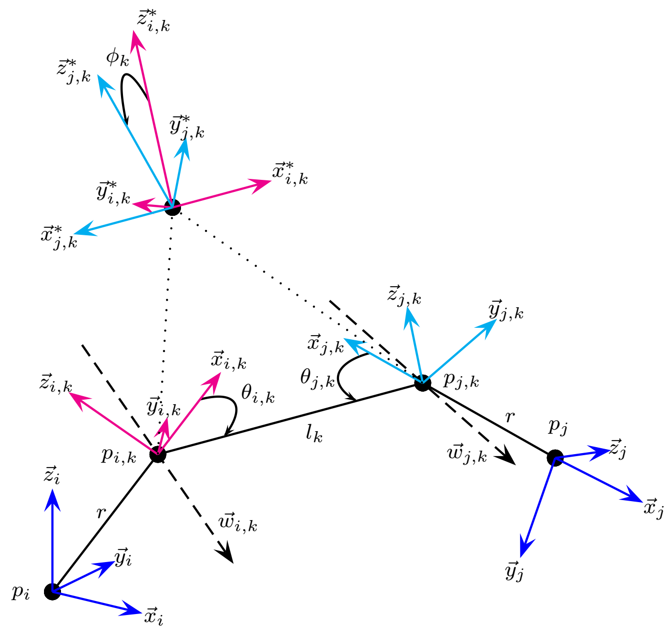


Fig. 2. Two nodes, i and j, interconnected by link k equipped with two flexible connectors. The position of the first node i is denoted p_i and its orientation is illustrated by the axes $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$. The connection point on which link k is connected to node i is denoted $p_{i,k}$. The orientation of the connection point is rotated so that the x-axis, $\vec{x}_{i,k}$, is along the line from the center of the node to the connection point and pointing away from the node. This is the same for node j. When both connection joints are rigid, we choose $\vec{z}_{i,k} = \vec{z}_{j,k}$. If one or both joints are flexible, we choose $\vec{z}_{i,k}$ to be equal to $\vec{z}_{j,k}$ at zero bending and twist angles.

The link matrix describes what functionality is included in a configuration of the Odin robot. If we look at the example on Fig. 1 there are three types of links: Battery links BL, telescoping links TL, and a wireless link WL. The link matrix describing this looks like:

$$L_{Example} = \begin{pmatrix} 0 & BL & BL & TL \\ - & 0 & BL & TL \\ - & - & 0 & TL \\ - & - & - & WL \end{pmatrix} \quad (3)$$

which shows that the battery links are connected in the triangular base, and the telescoping links are able to move the top node. A wireless link is also connected to the top node to enable debugging and remote control.

V. Identifying Constraints

If we neglect rotation of the connection links around their symmetry axes, we can uniquely represent the shape of a configuration by the position and orientation of each node. The position and orientation of a node in three dimensions can be described by six variables. The number of variables of a configuration with n nodes is then $6(n-1)$ since the base node, node 0, is fixed.

$$q = \{q_1, q_2, \ldots, q_{n-1}\} \quad (4)$$

where

$$q_i = \{x_i, y_i, x_i, \alpha_i, \beta_i, \gamma_i\} \quad (5)$$

c	γ_c	β_c	a_c
1	0	0	r
2	$\pi/3$	0	r
3	$2\pi/3$	0	r
4	$\pi/6$	$-\arctan\left(\sqrt{2}\right)$	r
5	$5\pi/6$	$-\arctan\left(\sqrt{2}\right)$	r
6	$3\pi/2$	$-\arctan\left(\sqrt{2}\right)$	r
7	$\pi/2$	$\arctan\left(\sqrt{2}\right)$	r
8	$11\pi/6$	$\arctan\left(\sqrt{2}\right)$	r
9	$7\pi/6$	$\arctan\left(\sqrt{2}\right)$	r
10	$5\pi/3$	0	r
11	$4\pi/3$	0	r
12	π	0	r

TABLE I

ROTATION AND TRANSLATION CONSTANTS FOR THE CONNECTION POINTS ON THE CCP NODE.

Using this representation, the constraints associated with a link can be described by its local variables, which enables us to look at each connection between two nodes individually. Another option would be to represent it by the degrees of freedom of the passive joints, however, the model have shown to be much more complex using this approach.

Fig. 2 illustrates two nodes, i and j, interconnected by a link, k equipped with a flexible connector at both ends. The position of the first node i is denoted p_i and its orientation is illustrated by the axes $(\vec{x}_i, \vec{y}_i, \vec{z}_i)$. The connection point on which link k is connected to node i is denoted $p_{i,k}$. The orientation of the connection point is rotated so that the x-axis, $\vec{x}_{i,k}$, is along the line from the center of the node to the connection point and pointing away from the node. This is the same for node j.

The position and orientation of node i can be described by the transformation T_i. For the CCP node the connection points are arranged according to the CCP lattice. The transformation from node i to the connection point (i,k) can then be found by:

$$^{i}T_{i,k} = Rot[z_i, \gamma_c] \cdot Rot[y_i^*, \beta_c] \cdot Trans[x_{i,k}, a_c] \qquad (6)$$

where γ_c, β_c, a_c are constants which can be looked up in Table I, since the CCP node is rigid. If both connectors on link k connecting node i and j are flexible the length between the connection points must be equal to the length of the link, l_k. However, if the connector on node i is rigid, while the connector on node j is flexible, node i and link k becomes one rigid body. The connection point on node i can now be placed at the end of link k, and the distance between connection point (i,k) and (j,k) must now be zero.

$$0 = \begin{cases} |p_{j,k} - p_{i,k}| - l_k & CM_{i,j} = FC \\ |p_{j,k} - p_{i,k}| & CM_{i,j} = RC \end{cases} \qquad (7)$$

for $i = 0$ to n and $j = i + 1$ to n, and where the position and orientation of connection point (i,k) is described by the transformation:

$$T_{i,k} = \begin{cases} T_i \cdot^{i} T_{i,k} & CM_{i,j} = FC \\ T_i \cdot^{i} T_{i,k} \cdot Trans[\vec{x}_{i,k}, l_k] & CM_{i,j} = RC \end{cases} \qquad (8)$$

This is, though, only the case when $CM_{i,j} \neq 0$. Also elements in the diagonal does not add any constraints.

If a link is equipped with rigid connectors on both ends the links does not add a constraint. The two nodes interconnected can then be represented by one rigid body where the position and orientation of node j can be described by:

$$T_j = T_i \cdot^{i} T_{i,k} \cdot Trans[\vec{x}_{i,k}, l_k] \cdot Rot[\vec{z}_{i,k}, \pi] \cdot [^{j}T_{j,k}]^{-1} \quad (9)$$

VI. ESTIMATING THE SHAPE

If all the connectors are rigid the Odin robot is completely rigid and the configuration has only one solution. Though, a typical configuration will be a combination of rigid and flexible connectors. Since the flexible connectors has a springy joint, the configuration has only one solution, except if the configuration has bi-stable states. Had the flexible joints not been springy, a flexible configuration would have infinite solutions.

The static equilibrium can be found by minimizing the total potential energy stored in all the springs of the system, assuming that there are no external forces on the system. The potential energy stored in the springs depends on the deflection angles and the twists in the flexible connectors' ball-and-socket joints. The deflection angle for each flexible joint can be found by.

$$\theta_{i,k} = \begin{cases} \arccos(\vec{x}_{i,k} \cdot \vec{v}_{i,k}) & CM_{i,j} = FC \\ 0 & CM_{i,j} = RC \end{cases} \qquad (10)$$

where $\vec{v}$ is the vector along the connection link from node i towards node j

$$\vec{v}_{i,k} = \begin{cases} p_{j,k} - p_i & CM_{i,j} = RC \\ p_j - p_{i,k} & CM_{j,i} = RC \\ p_{j,k} - p_{i,k} & \text{otherwise} \end{cases} \qquad (11)$$

Finding the twist angle ϕ_k is a bit more complicated. Unlike the deflection angle, there is only one twist angle for each link since it is equally distributed between the two springs at each connector, if both connectors are flexible. If only one connector is flexible, the twist is only present in that connector's spring. If both connectors are rigid, there is, of course, no twist between the node i and j. To find the twist angle we must first rotate the orientation of the connection point so that the resulting x-axis $\vec{x}_{i,k}^*$ lies along the vector $\vec{v}_{i,k}$. To do this we find the axis vector $\vec{w}_{i,k}$ perpendicular to $\vec{x}_{i,k}$ and $\vec{v}_{i,k}$, and rotate the orientation of the connection point by the deflection angle $\theta_{i,k}$.

$$\vec{w}_{i,k} = \frac{\vec{x}_{i,k}}{|\vec{x}_{i,k}|} \times \frac{\vec{v}_{i,k}}{|\vec{v}_{i,k}|} \qquad (12)$$

$$\vec{w}_{i,k}^* = (T_{i,k})^{-1} \cdot \vec{w}_{i,k} \qquad (13)$$

$$T_{i,k}^* = T_{i,k} \cdot Rot[\vec{w}_{i,k}^*, \theta_{i,k}] \qquad (14)$$

Require: IM, CM, LM {configuration matrices}
Require: l {module length vector}
  $n \leftarrow dim(IM)$ {number of nodes}
  for $i = 1$ to $n-1$ **do**
    for $j = i+1$ to $n-1$ **do**
      if $IM_{i,j} \neq 0$ and $IM_{j,i} \neq 00$ **then**
        - Find $p_{i,k}$, $p_{j,k}$, l_k
        - Identify constraints
        - Find angles $\theta_{i,k}$, $\theta_{j,k}$, ϕ_k
        $k \leftarrow k+1$
      end if
    end for
  end for
  - Find potential energy $U_{total}[q]$
  - Minimize potential energy by solving constrained optimization problem

Fig. 3. Pseudo-code for identifying the constraints and calculating the potential energy of an Odin configuration and estimating its shape.

where $\vec{w}^*_{i,k}$ is the axis vector relative to the position and orientation of the connection point. The twist angle can now be found by finding the angle between the z-axes of $T^*_{i,k}$ and $T^*_{j,k}$

$$\phi_k = \arccos\left(\frac{\vec{z}^*_{i,k} \cdot \vec{z}^*_{j,k}}{|\vec{z}_{i,k}||\vec{z}^*_{j,k}|}\right) \qquad (15)$$

The potential energy of a rotational spring can be written as:

$$U = 0.5 \cdot k_s \cdot \theta^2 \qquad (16)$$

where k_s is the spring constant and θ the angle. In this paper we assume that all the springs are identical, and the springconstant for the deflection and the twist are equal. The total potential energy of the system can then be written as:

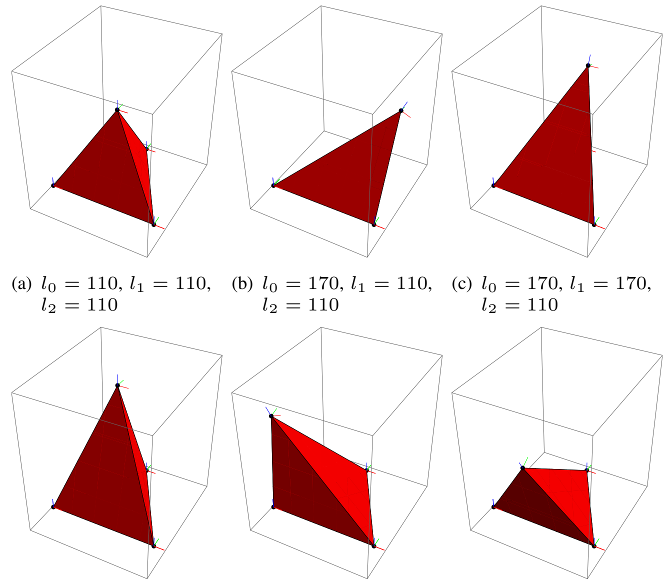
$$U_{total}[q] = 0.5 \cdot k_s \cdot \sum_{k=0}^{m-1} \left(\theta_{i,k}{}^2 + \theta_{j,k}{}^2 + c_k \phi_k{}^2\right) \qquad (17)$$

where m is the number of links in the system and $c_k = 1$ if both ends are flexible connectors and $c_k = 2$ if one end is rigid (placing the whole twist on the other end).

Now, to estimate the shape of the system we must minimize the sum of the potential energy while fulfilling the constraints. This is a constrained optimization problem which for this paper has been solved using the built-in Mathematica function *FindMinimum*. Fig. 3 shows pseudo-code for identifying the constraints and calculating the potential energy of an Odin configuration and estimating its shape.

VII. Tetrahedron Example

Now that we have described the methods for deriving the constraints and the potential energy of a configuration represented by the configuration matrices we present an example of using the methods. If we look at the example on Fig. 1, which was presented previously, we see that it has four nodes. Since the base is connected with links equipped with rigid connectors, it can be described as one rigid body. The top node is connected with three telescoping links equipped with flexible connectors. By changing the lengths of the telescoping links the top node can move. A wireless link is connected on to the top node, but does not affect the kinematic structure.



(a) $l_0 = 110$, $l_1 = 110$, $l_2 = 110$ (b) $l_0 = 170$, $l_1 = 110$, $l_2 = 110$ (c) $l_0 = 170$, $l_1 = 170$, $l_2 = 110$

(d) $l_0 = 170$, $l_1 = 170$, $l_2 = 170$ (e) $l_0 = 110$, $l_1 = 170$, $l_2 = 170$ (f) $l_0 = 110$, $l_1 = 110$, $l_2 = 170$

Fig. 4. Results for estimating the shape of a tetrahedron configuration.The leftmost node is fixed. l_0 is the length of the link between the leftmost node and the top node, l_1 is the length between the rightmost node and the top node, and l_2 is the length between the far node and the top node.

When minimizing the potential energy we need a start guess for the variables which is relatively close to the solution, for the numerical optimization method to converge. Initially the start guess is found by contracting all the telescoping links, so that the lengths of all links are equal. The orientation of the nodes must all be equal to the orientation of the first node which is fixed, otherwise the structure will not conform to the lattice. The potential energy of the initial shape is zero, since all the angles are zero. While continuously finding a solution using the previous solution as the start guess, we can start actuating the telescoping links. The algorithm has not yet been implemented on the modules, but since the actuators run fairly slow compared to the microcontrollers on the modules, the start guess should always be fairly close to the solution. This, of course, depends heavily on the number of modules in the configuration, and must be investigated further in future work.

Fig. 4 shows the results of estimating the shape for the tetrahedron configuration. We have illustrated six extremes where the telescoping links are either contracted or fully extended. From the example we see that the tetrahedron is able to change its shape significantly, and it is clear that the equilibrium found is consistent with the physical robot. Further investigation of the consistency of the results and the performance of the algorithm will be conducted when the algorithm is implemented on the physical robot. Since the constraints and angles associated with a link only are functions of local variables, the method can potentially be distributed to increase the scalability of the method.

VIII. Discussion

Since the configuration of a modular robot changes for different tasks it is important to have a general representation of the robot. For the Odin robot we have achieved this by developing three representation matrices describing the interconnection, the connectors used, and the type of modules that contributes to the functionality of the robot. The representation matrices can be used to remember useful configurations and recreate these configurations of the robot both in simulation and in the physical world, and in between those two. In this paper we have used the representation matrices to generalize a method for estimating the shape of the Odin robot.

With the current modules, the Odin robot relies on its ability to deform to be able to perform locomotion and manipulation, or to resemble a shape for a physical display. It can only be reconfigured by hand for different environments and tasks. How the actuators control its shape in a given configuration is therefore important for controlling the robot. This paper does not present a general study on the kinematics of the Odin robot. However, to understand the physical structure, we have done a short kinematic analysis of the robot to illustrate how to represent different Odin structures and estimate their shape.

In future work, estimating the shape will be used to analyze the flexibility of the Odin robot. The flexibility of the ball-and-socket joints is limited by the mechanical constraints which does not allow the deflection and twist angle to be more than 20° degrees. In the tetrahedron example discussed in the previous section the deflection angles reach a maximum of 42° degrees. This is also due to the fact that the base is rigid. Had the base been more flexible, the potential energy would divide the angles more evenly among the connectors. However, this is an indication of a design constraint that should be reconsidered when designing the next iteration of the Odin robot.

The proposed method for estimating the shape of the Odin robot does not include external forces. In the physical world the robot is constantly influenced by gravity, and as it encounters obstacles other external forces will apply. In the tetrahedron example on Fig. 1 gravity is negligible compared to the spring forces, but in larger configurations the estimated shape will deviate from the physical shape. However, if the robot is able to sense its physical shape while knowing the estimate, the deviation can be used to reason on the external forces. Having knowledge about the external forces can be used to detect different environments and materials for optimizing the control strategies. In a search and rescue operation this knowledge could also be used to differentiate humans from fallen rocks.

If we instead of setting the lengths of the links, when minimizing the potential energy, set the positions and orientations of the nodes as a representation of the desired shape, we may also be able to estimate the length of the links to reach the desired shape. This could be used to create physical displays, however, the desired shapes has to to lie within the configurations' workspace.

IX. Conclusion

This paper presents the basics of estimating the shape, and leads the way for further research on the Odin robot, and similar robots. We have, in short, analyzed the kinematics of the Odin robot in order to get an understanding of its physical structure. We have presented a representation for the Odin robot which describes the physical structure of the robot and the module functionalities in the configuration. We have used this representation to generalize the identification of the physical constraints, and to identify the deflection and twist angles within a configuration. By minimizing the potential energy, induced by the springs in the connectors depending on the deflection and twist angles, we have presented an example of estimating the shape of a tetrahedron configuration. We conclude that the representation is well suited for generalizing shape-estimation of the Odin robot.

References

[1] 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, March 2003.

[2] M. Yim, D. G. Duff, and K. Roufas, "Modular reconfigurable robots, an approach to urban search and rescue," in *Proc. 1st International Workshop on Human-friendly Welfare Robotics Systems*, Taejon, Korea, January 2000, pp. 69–76.

[3] T. Fukuda, S. Nakagawa, Y. Kawauchi, and M. Buss, "Self organizing robots based on cell structures - cebot," in *Proc. IEEE/RSJ International Workshop in Intelligent Robots and Systems (IROS 1988)*, Tokyo, Japan, 1988, pp. 145–150.

[4] P. Pillai, J. Campbell, G. Kedia, S. Moudgal, and K. Sheth, "A 3D fax machine based on claytronics," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 4728–4735.

[5] J. Sastra, S. Chitta, and M. Yim, "Dynamic rolling for a modular loop robot," *International Journal of Robotics Reseach Invited*, 2007.

[6] A. Lyder, R. F. M. Garcia, and K. Stoy, "Mechanical design of odin, an extendable heterogeneous deformable modular robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, September 2008, pp. 883–888.

[7] E. H. Østergaard, K. Kassow, R. Beck, and H. H. Lund, "Design of the ATRON lattice-based self-reconfigurable robot," *Autonomous Robots*, vol. 21, no. 2, pp. 165–183, September 2006.

[8] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-reconfigurable modular robotic system," in *Proc. IEEE/ASME Transactions on Mechatronics*, vol. 7(4), December 2002, pp. 431–441.

[9] J. W. Suh, S. B. Thomas, and M. Yim, "Telecubes: Mechanical design of a module for self-reconfigurable robotics," in *Proc. IEEE International Conference on Robotics and Automation, (ICRA 2002)*, Washington, DC, May 2002, pp. 4095–4101.

[10] G. J. Hamlin and A. C. Sanderson, "Tetrobot modular robotics: Prototype and experiments," in *Proc. IEEE International Conference on Intelligent Robots and Systems, (IROS 1996)*, vol. 2, Osaka, Japan, November 1996, pp. 390–395.

[11] K. Harib and K. Srinivasan, "Kinematic and dynamic analysis of stewart platform-based machine tool structures," in *Robotica*. Cambridge University Press, 2003, vol. 21, pp. 541–554.

[12] R. P. Sunkari, "Structural synthesis and analysis of planar and spatial mechanisms satisfying gruebler's degrees of freedom equation," Ph.D. dissertation, University of Maryland, 2006.