Docking in Self-Reconfigurable Robots

Wei-Min Shen and Peter Will
Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292, USA
{shen,will}@isi.edu

Abstract

Docking is a crucial action for self-reconfigurable robots because it supports almost all practical advantages of such robots. In addition to the classic docking challenges found in other applications, such as reliable dock/latch mechanics, effective guiding systems, and intelligent control protocols, docking in self-reconfigurable robots is also subject to some unique constraints. These constraints include the kinematics constraints imposed on the docking modules by other modules in the configuration, communication limitations between the docking and relevant modules, and the demand for distributed control software because of the dynamics of configuration. To solve these challenging problems, this paper reports a set of solutions developed in the CONRO reconfigurable robot project. The paper presents a three-stage docking process, six different alignment protocols, distributed inverse kinematics, and other techniques such as dynamic lubrication that are essential for successful docking in CONRO-like robots. These solutions enable CONRO robots to perform autonomous and distributed reconfigurations in a laboratory environment, and they also suggest important considerations for docking in self-reconfiguration in general.

1. Introduction

Self-reconfigurable robots are robots that can change their individual and collective shape and size in order to meet operational demands in uncertain and dynamic environments. Such robots typically consist of a collection of autonomous modules that can connect and disconnect among themselves. Since reconfigurations in these robots are achieved by changing the interconnections among modules, the actions of docking and dedocking play a crucial role for the success of self-reconfigurable robots.

A successful docking action between two connectable modules consists of at least three integrated complex stages. First, the robot must maneuver all relevant modules in the system so that the two docking modules are physically positioned close to each other. For example, for a snake-shaped robot to become a loop (connecting the head with the tail), all modules in the snake must bend their joints in order to bring the head of the snake to see the tail. Second, the two docking modules must be aligned to each other to satisfy the constraints for docking. This alignment is a complex process and must coordinate the actions of many modules in compliant with the perceived docking guidance signals. Third, once the two modules are aligned for docking, they must be pushed to establish the final mechanical/electronic connection. This final establishment typically requires movements in a precise trajectory with critical forces.

The nature of self-reconfigurable robots also poses additional complexities for the above three stages. In a self-reconfigurable robot, docking is not an action local to the two docking modules but an action global to many modules in the configuration. For example, many modules must be moved in order to bring the two docking modules together, and many modules must apply their torques in order to push the docking modules with critical forces. Docking in a self-reconfigurable robot is also a distributed action. Due to the dynamics of robot configuration, we cannot pre-determine a designated module to perform the centralized control for all possible docking situations because any damage to this module will paralyze the entire system. Finally, docking in self-reconfigurable robots must also be fast and efficient (for docking is performed very frequently in a self-reconfigurable robot), reliable (a docked connection must be as secure as a fixed connection), and energy saving (an established connection should not consume any power).

Due to these difficulties, autonomous docking remains an active research topic for self-reconfigurable robots. For example, Nilsson [1] has designed a 2D self-aligning docking device but it is still an open problem how to trade the device's generality for the tolerance of errors. Roufas et. al. [2] have experimented 6D docking sensing using IR LEDs. Fukuda and Nakagawa [3] studied docking with CEBOT. Murata et. al. [4] constructed a complex mechanism for connecting arms. Bereton and Khosla [5] have used visual images as guidance for docking between mobile robots. Their docking connector has a forklift and a receptacle and allows approximately 30-degree alignment errors. Their robots are skid-steered, i.e., when the forklift pins are partially in the receptacle and robot pushes straight ahead, the wheels of the robots would slip on the ground and allow the robot to center the pins in the receptacle. This feature, however, is not generally available for modular self-reconfigurable robots. Many researchers also assumed simplified solutions for docking.

For example, the current prototypes of Polybots [6] use tele-operations to assist docking. Proteo robots [7] assume that a module can dock with another by "rolling over" onto that module and such actions are local and trivial. Robot Molecules [8] sacrifice the low-power consumption requirement to use simple magnetic or electro-magnetic connections.

This paper is a report of a set of integrated solutions to the docking problem in CONRO-like self-reconfigurable robots. Although the current experiments are for 2D docking, the challenges listed above (such as global cooperation, distributed control, speed and efficiency) are all present and the integration of all component solutions into a working system is a non-trivial matter. The paper discusses solutions for alignment, trajectory following, distributed inverse kinematics, and other issues in docking. Section 2 describes the hardware of CONRO docking mechanism and guidance system. Section 3 presents the software architecture for distributed control of docking. Section 4 gives a brief discussion on the dedocking process. Section 5 presents the method for distributed inverse kinematics. Section 6 reports docking experimental results in CONRO and compare six different alignment strategies. Finally, Section 7 concludes the paper with a set of future research directions.

2. CONRO Docking/Guidance Hardware

The CONRO self-reconfigurable robots are made of a set of connectable, autonomous, and self-sufficient modules. Illustrated at the top part of Figure 1, each module has two batteries, one micro-processor, two motors, four docking connectors for connecting with other modules, and four pairs of infrared emitter/receiver for communicating and sensing other modules. Some modules are also equipped with other miscellany sensors such as tilt sensors and miniature cameras. More information and movies of CONRO can be found at http://www.isi.edu/conro.

Each CONRO module has two degrees of freedom: DOF1 for pitch (up and down) and DOF2 for yaw (left and right). Each DOF has a home position (when the joint is straight), and has two joint limits (when the joint reaches the maximal or the minimal angle). With these two DOFs, a single module can wiggle its body, and two or more connected modules can perform locomotion with coordinated actions. For example, a body of six legs can perform hexapod gaits, while a chain of modules can mimic a snake or a caterpillar motion (see the lower part of Figure 1). The control of the coordinated actions are distributed among modules and conducted by a novel and powerful distributed control mechanism called *Hormone-Based Control* [9-11].

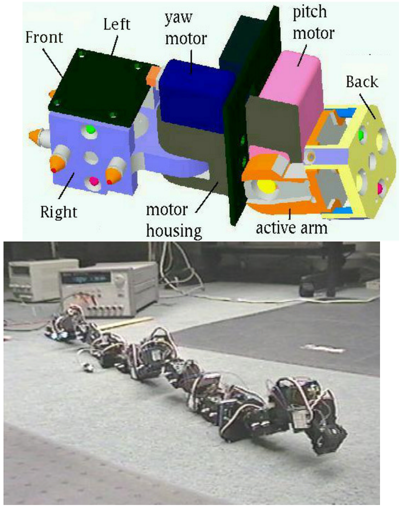

Figure 1: The modules of CONRO robots

CONRO modules can connect to each other by their docking connectors located at either ends of a module. At one end, three male connectors are located at the three sides of the module (shown as *Front*, *Left*, *Right* in Figure 1), each of which consists of two docking pins. At the other end, a female connector is located at the tip end of a module (shown as *Back* in Figure 1), which consists of two holes for accepting other module's docking pins. This female connector has a locking/releasing mechanism behind the holes, and can have two states. In the default or non-active state, it can accept and lock the incoming pins by a spring motion. In the activated state, it can release the lock by triggering a SMA actuator. The connector/leasing mechanism is power efficient and it consumes no electric energy when in the default state. For detailed design and implementation of these docking connectors, please see a separate paper [12].

CONRO modules communicate with each other using infrared transmitters and receivers. At each connector, there is an infrared emitter/receiver pair located between the pins/holes (shown as the upper and lower holes on the vertical center line on each connector at the top part of Figure 1). When a module is connected to another module through a connector, the two pairs of infrared emitter/receiver at the docked connectors will be aligned to form a bi-directional infrared communication link. Since there are four connectors for each module, there can be up to four communication links for each module.

The infrared emitter/receivers are also used as sensors for guiding two modules to align each other during a docking action. When two modules are in the range of the infrared signals, they can measure the strength of the received the signal and use the measurements to estimate the quality of the alignment (i.e., orientation and distance) between two modules. The correlation between the signal measurement and the quality of the alignment is straightforward. With fixed orientations of the two modules, the measurement corresponds linearly with the distance between the modules. Similarly, with a fixed distance between the two modules, the measurement corresponds to the orientations of the modules.

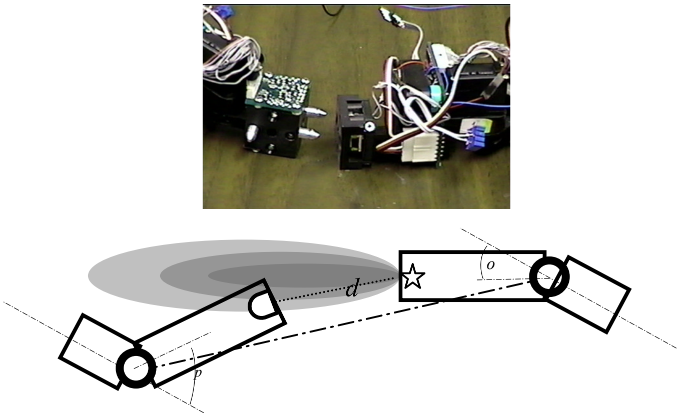

Figure 2: The measurement of docking signals

Figure 2 shows two CONRO modules in a close distance range and illustrates the docking signal in a similar situation, where *o* and *p* are the orientation angles of the two modules, *d* the distance between them, and the dashed line indicates the desired alignment in this particular situation. One uncertain element in this guidance system is the shape of infrared lobe from the emitter. There is no guarantee that a given emitter will produce an ideal lobe (shown in Figure 2) that points to the correct direction and that has a smooth gradient field with a single maximum peak. To make a successful docking, the lobe of the signal must be tuned and the control protocol must take this uncertainty into consideration. Furthermore, according to the features of infrared transmitting and receiving, adjusting one docking module alone can only result in a local maximum in the alignment. In order to achieve the global optimal alignment, both docking modules must adjust their orientations collaboratively.

3. Docking Control for CONRO Robots

As we mentioned earlier, there are three stages in a docking process. First, two to-be-docked modules must move close enough so that they can sense each other's docking guidance signals. Second, the two docking modules must use the docking guidance signals to maneuver their locations and orientations so that their connectors are aligned and close to each other. Finally, the modules must push the connectors into each other so the mechanical mechanism can be securely latched and locked. Notice that a unique challenge in docking is that the two docking modules are not free-floating elements, but constrained as two parts of the same body. Thus, the movement of these modules may be related and can influence each other. As a result, all three stages must involve many modules in addition to the two to be docked in the configuration.

In the following discussion for the three stages, all communication between modules is subject to the constraint that modules can only talk to their immediate neighbors. Thus, a message from the head of a snake robot to the tail module must relay through all modules in the body. This communication constraint forces us to prefer a local and distributed control method to a centralized control method so that we do not need to assume any pre-designated module as a brain-like control center.

3.1 Maneuvering for alignment preparation

To bring two modules close enough for docking, we assume that the two docking modules are connected through a "docking chain" of modules in the current configuration, and this chain must be long enough so that it can bend to allow the two docking modules sense and touch each other. If we view a robot configuration as a graph with modules as nodes and connections as edges, then the docking chain is the shortest path in the graph between the two docking module nodes.

Let L be the number of modules in the docking chain and assume all modules in the docking chain are connected "straightly" through front-back connectors, then each module must bend approximately $2\pi/L$ to allow the two docking module to see and touch each other. In general, however, not all connections are straight, and a connection can be perpendicular either "positively" (if it branches out on the same side of the bending direction), or "negatively" (if it branches out on the opposite side of the bending direction). Since a positive connection contributes $+\pi/2$ to the overall bending, while a negative one contributes $-\pi/2$, each module in a general docking chain should bend

$$(4+m-n)\pi/2L \quad \ldots\ldots \quad (1)$$

where m and n is the number of "positive" and "negative" connections in the chain, respectively. Notice that when m=n=0, then the above equation gives $2\pi/L$, which is the expected value for a straight chain.

Let $B_{\min}$ and $B_{\max}$ be the minimal and maximal bending angle for each module, then the necessary condition for bending the docking chain to allow the two docking modules to sense and touch each other is

$$B_{\min} < (4+m-n)\pi/2L < B_{\max} \quad \ldots\ldots \quad (2)$$

This necessary condition for docking can be used to check if two modules in a given configuration can dock with each other or not. Given any pair of modules in a configuration graph, one can find the docking chain in the graph using the minimal spanning tree algorithm. The values of L, m, and n can then be identified based on the length of the chain and the types of connections involved in the chain. Again, all these computations can be achieved using the powerful, distributed hormone-based control framework [9-11].

3.2 The leader-follower alignment protocol

Once the modules in the docking chain are bent using the method described above, the two docking modules are able to sense each other’s docking guidance signals. To accomplish a proper alignment, however, the two docking modules must both work actively and collaboratively. This is different from many docking tasks in classic applications, such as a spacecraft docking to a space station or a truck docking to a loading deck, where one side of the docking can be assumed static while the other can move freely and independently. In self-reconfigurable robots, however, the movements of two docking modules are limited and constrained by the current robot configuration (all other modules are assumed to be held rigid), and an alignment cannot be achieved unless both sides of the docking adjust their orientations and positions. In a 2D environment shown in Figure 2, for example, unless both docking modules adjust their orientations (o and p), the desired alignment cannot be achieved.

To solve this problem, we have investigated a number of different search strategies that adjust the orientations of the two docking modules in some joint fashion. One approach that works well is to engage the two modules in a leader-follower relationship and adjust their orientations jointly in a hill-climbing search for the maximum measurement of guidance signals. The basic idea is that whenever the leader module (which could be chosen arbitrarily) rotates to an orientation o, it will ask the follower module to perform a scan and find/report an orientation p for the follower that gives the best guidance signal measurement m_{op}. The value p and m_{op} indicate the goodness of the leader’s orientation o in the context of aligning with the follower. In the following description, we call the above procedure “find the best alignment for the follower” (FBAF) and assume the procedure FBAF(o) returns a pair of values (p, m_{op}). We also use the notation (o, (p, m_{op})), where (p, m_{op})=FBAF(o), to represent a joint alignment (o, p) and its goodness m_{op}. Using these measurements, the leader can decide to which direction it should move its orientation in order to perform an effective search for the best alignment. It can also be used to decide when the alignment process can be determined.

To decide the direction for orientation search, the leader first evaluates two consecutive joint alignments (o, (p, m_{op})) and (o', (p', m'_{op})), where $o'=o+\Delta$. The value of Δ represents the size of increment in orientation change, and the sign of Δ represents the direction of orientation change. If $m'_{op}>m_{op}$, then the leader selects Δ as its search direction. If $m'_{op}<m_{op}$, the leader selects $\Delta = -\Delta$ as its search direction. If $m'_{op}=m_{op}$, the leader repeats the above process until $m'_{op}\neq m_{op}$

Once the direction of orientation search is determined, the above procedure can be used for the leader to find the best alignment by hill-climbing. Let (o,(p,m_{op})) be the current alignment. The leader increments its rotation $o'=o+\Delta$ and obtains (p',m'_{op}) for o' from the follower. If $m'_{op}\geq m_{op}$, which indicates that o' gives a better or equal alignment than o, then the leader sets $o=o'$, $p=p'$, $m_{op}=m'_{op}$, and continues the research. If $m'_{op}<m_{op}$, which indicates a decrease in the quality of alignments, then the leader determines the search and declares (o,p,m_{op}) as the best alignment for the current distance.

Once the best alignment has been found, the two docking modules must move towards each other in the trajectory specified by the alignment. This will reduce the distance between the two modules and increase the value of m_{op}. Since the movements in self-reconfigurable modules are inevitably noisy and uncertain, a new alignment must be performed after the distance is reduced. This alignment-then-move-closer action will be repeated until the value of m_{op} is above a threshold. This indicates that the two connectors are aligned and close enough, and the final pushing stage can begin.

3.3 Establishing the final connection

The final pushing stage in the docking is to have the two modules to push each other in the trajectory specified by the alignment. Such movements, which are also used to reduce the distance between docking modules during alignment, require multiple modules to coordinate their actions using inverse kinematics. For example, imagine that the two docking modules in Figure 2 are aligned along the dashed line. In order to move the left (right) docking module along the alignment line, the movements of the modules behind the left (right) docking module must be coordinated using inverse kinematics. It is such coordinated movements that provide the necessary pushing force for the finally docking stage. Since the movement must not deviate from the trajectory, the step of the movement must be sufficiently small in order to ensure the smoothness of the movement. It is interesting that the computation of inverse kinematics can be distributed among modules, as we will see in the next section.

The final pushing must also overcome the friction between the docking pins and their corresponding sockets and latches. This friction is significant in self-reconfigurable robot because the strengths of modules’ motors are typically limited. To overcome this problem, we have adopted the idea of dynamic lubrication. During the latching and locking stage, the two docking modules are also performing high-frequency small “shaking” movements while pushing forward. Such movements significantly reduce the friction during docking and ensure that the docking pins to be securely locked by the spring latch behind the holes.

4. Distributed Inverse Kinematics

One action that is frequently used in docking is to move a docking module on a given trajectory. For example, after two docking modules are aligned, they must move along their tip directions respectively to reduce the distance between them. Similarly, such movement is also used to push the docking pins into the holes and lock them by the spring latch.

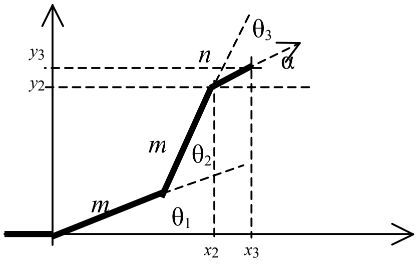

Figure 3: Inverse kinematics for three modules

To generate such actions, a chain of modules that is immediately connected to the docking module must coordinate their movement using inverse kinematics. This is similar to the classic robotics problem of controlling a robot arm to put a peg into a hole. The difference is that each module in a reconfigurable robot is an independent and autonomous system, while in classic robotics applications all motors and sensors are controlled by a single computer.

To distribute the computation for inverse kinematics among reconfigurable modules, let us consider a three-module chain illustrated in Figure 3. Given the current angles of each module θ_1, θ_2, and θ_3, and the lengths of the relevant segments *m* and *n,* we can compute the tip position (x_3, y_3) using the forward kinematics as follows:

$$x_3=m\mathrm{Cos}(\theta_1)+m\mathrm{Cos}(\theta_1+\theta_2)+n\mathrm{Cos}(\theta_1+\theta_2+\theta_3) \quad \ldots(3)$$

$$y_3=m\mathrm{Sin}(\theta_1)+m\mathrm{Sin}(\theta_1+\theta_2)+n\mathrm{Sin}(\theta_1+\theta_2+\theta_3) \quad \ldots(4)$$

$$\alpha=\theta_1+\theta_2+\theta_3 \quad \ldots(5)$$

Assume the tip is to be moved along the α direction by a distance δ, then the new tip position (x'_3, y'_3), the new middle position (x'_2, y'_2), and the new module angles θ'_1, θ'_2, and θ'_3, can be computed as follows:

$$x'_3= x_3+\delta\mathrm{Cos}(\alpha) \quad \ldots(6)$$

$$y'_3= y_3+\delta\mathrm{Sin}(\alpha) \quad \ldots(7)$$

$$x'_2= x'_3-n\mathrm{Cos}(\alpha) \quad \ldots(8)$$

$$y'_2= y'_3-n\mathrm{Sin}(\alpha) \quad \ldots(9)$$

$$W=\mathrm{atan}(y'_2 / x'_2) \quad \ldots(10)$$

$$\theta'_1=W+\mathrm{acos}[\, x'_2 / 2m\mathrm{Cos}(W)] \quad \ldots(11)$$

$$\theta'_2=2W - 2\theta'_1 \quad \ldots(12)$$

$$\theta'_3= \alpha - (\theta'_1+\theta'_2) \quad \ldots(13)$$

Let the three modules from the origin to the tip of the chain be named as M1, M2, and M3, then under the constraints that modules can only communicate with their immediate neighbors, the above computation can be distributed among the three modules as follows:

M1 computes $A=m\mathrm{Cos}(\theta_1)$ and $B=m\mathrm{Sin}(\theta_1)$ and sends $[A,B,\theta_1]$ to M2;
M2 computes $\phi= \theta_1+\theta_2$, $C=A+m\mathrm{Cos}(\phi)$ and $D=B+m\mathrm{Sin}(\phi)$ and sends $[\phi,C,D]$ to M3;
M3 computes (3)-(9) and sends $[x'_2, y'_2]$ to M2;
M2 computes (10) and sends $[W, x'_2]$ to M1;
M1 computes θ'_1 by (11) and sends $[\theta'_1]$ to M2;
M2 computes θ'_2 by (12) and sends $[\theta'_1+\theta'_2]$ to M3;
M3 computes θ'_3 by (13).

This example illustrates that by sending intermediate results to neighbor modules, a chain of modules can compute the inverse kinematics in a distributed fashion. This style of computation is both desirable and sometimes necessary for self-reconfigurable robots because no single module can be assumed always available as the control center and every module is independent, autonomous, and has very limited on-board computation resource.

5. The Process of Dedocking

Compared to docking, dedocking is a relatively simple process because no alignment is needed. When a robot decides to disconnect an existing connection, it releases the latching mechanism by activating the SMA of the female connector for a short period of time. During this time, the robot "pulls" the two dedocking modules apart using the inverse kinematics method described above with a negative value for the distance parameter δ. The robot can check if a dedocking is successful or not by measuring the guidance signals between the two dedocking modules. If the measured value is below a threshold in comparison with the value when the connection is in place, then the dedocking succeeds and the robot can treat the two modules as disconnected.

6. Experimental Results

The above techniques and methods are implemented on CONRO reconfigurable modules, and we have conducted experiments on a 7-module snake robot to dock its head with its tail on a regular office table surface. We have made 10 runs each starting from a random initial state of being a straight snake and each run includes all three stages of docking. The average speed is approximately 3minutes/docking, and the successful ratio is 80%. The experiments are autonomous (i.e., all programs are on the modules) and modules are powered by an external power supply and triggered by a single message issued through an external command line.

We have also conducted a set of experiments to compare the difference between six alternative alignment search protocols (see Table 1). Each experiment is a single increment in the search process although it may involve many movements of modules. All experiments are from the same initial condition and repeated for 10 runs. In Table 1, each entry reports (1) the quality of the resulting alignment (i.e., the signal value measured at the end of the experiment), and (2) the number of module movements in each experiment, which is directly in proportional to the time needed to complete the alignment. As we can see, the protocol of leader-follower with rotating around the joints of modules finds the best alignment and requires the least number of movements and time.

Table 1: Comparison of alignment protocols

Leading schema	Rotating around the point of		
	Joints	Internal signal	External signal
Leader and Follower	(56±5) (7±3)	(46±10) (21±8)	(40±10) (34±9)
Alternate Leadership	(54±5) (11±3)	(43±5) (29±9)	(41±7) (38±10)

7. Conclusions

In this paper, we have developed a set of solutions for docking in self-reconfigurable robots and demonstrated the results through experiments with the CONRO reconfigurable robots. Our future research includes docking in 3D environment as well as automatic selection of configuration based on perceptions. Both tasks are necessary for the deployment of self-reconfigurable robots to real-world applications.

8. Acknowledgements

This research is sponsored mainly by DARPA/MTO under contract number DAAN02-98-C-4032, and in part by AFOSR under award number F49620-97-1-0501. We thank B. Khoshnevis, A. Castano, and R. Kovac for their contributions in designing and constructing the CONRO modules and connectors.

References

1. Nilsson, M. *Symmetric Docking in 2D: A Bound on Self-alignable Offsets.* in *Proc. IASTED'99: Robotics and Automation.* 1999. Santa Barbara, California.
2. Roufas, K., Y. Zhang, D. Duff, M. Yim. *Six degree of freedom sensing for docking using IR LED emitters and receivers*. in *The 7th Intl. Symp. on Experimental Robotics*. 2000. Hawaii.
3. Fukuda, T., and Y. Kawauchi, *Method of Autonomous Approach, Docking and Datching Between Cells for Dynamically Reconfigurable Robotic System CEBOT*. Japan Soc. Mech. Eng. (JSME) Int'l Journal, 1990. **33**(2): p. 263-268.
4. Murata, S., E. Yoshida, H. Kurokawa, K. Tomita, S. Kokaji, *Self-Repairing Mechanical Systems.* Autonomous Robots, 2001. **10**: p. 7-21.
5. Bererton, C., P. Khosla. *Toward a Team of Robots with Repair Capabilities: A Visual Docking System.* in *Seventh International Symposium on Experimental Robotics*. 2000.
6. Yim, M., D.G. Duff, K.D. Roufas. *PolyBot: A Modular Reconfigurable Robot*. in *Proceedings of the IEEE International Conference on Robotics and Automation*. 2000.
7. Bojinov, H., A. Casal, and T. Hogg. *Multiagent Control of Self-Reconfigurable Robots*. in *International Conference on Multi-Agent Systems*. 2000.
8. Kotay, K., D. Rus, M. Vona, and C. McGray. *The self-reconfiguring robotic molecule*. in *Proceedings of the IEEE International Conference on Robotics and Automation*. 1998.
9. Shen, W.M., Y. Lu, P. Will. *Hormone-based Control for Self-Reconfigurable Robots*. in *Proceedings of International Conference on Autonomous Agents*. 2000. Barcelona, Spain.
10. Shen, W.M., B. Salemi, and P. Will. *Hormones for Self-Reconfigurable Robots*. in *Proceedings of the 6th International Conference on Intelligent Autonomous Systems*. 2000. Venice, Italy.
11. Salemi, B., W.-M. Shen, P. Will. *Hormone-Controlled Metamorphic Robots*. in *International Conference on Robotics and Automation*. 2001. Seoul, Korea.
12. Khoshnevis, B., A. Castano, R. Kovac, W.-M. Shen, P. Will. *Reconnectable Joints for Self-Reconfigurable Robots*. in *International Conference on Intelligent Robots and Systems*. 2001.