Mathematical Foundation for Hormone-Inspired Control for Self-Reconfigurable Robotic Systems

Feili Hou, Wei-Min Shen

Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292, USA
Email: {flhou, shen}@isi.edu

Abstract – **In this paper, we present a general mathematical foundation of hormone-inspired control for the self-reconfigurable robotic system. Problem considered here is the lack of a mathematical description to analyze and explain the dynamic behavior of self-reconfigurable robots. In the global level, the idea of virtual disconnection is developed to abstract the low level module control away from the high level synchronization for both cyclic and acyclic robot configuration. In the module layer, the linear space model is developed to describe each module's internal state, input-output hormone transformation, and its action selection. As a combination of hormone and modern control theory, the approach in this paper gives more features, such as predictability and stability analysis etc, to hormone-inspired control, and makes it applicable to self-reconfigurable systems in general. Simulation and experimental results show the capacity of our method.**

Index Terms – reconfigurable robotics system, state space model, hormone-inspired distributed control, virtual disconnection

I. INTRODUCTION

One of the challenges in self-reconfigurable robots is the collaboration between the modules to accomplish global behaviors. How to control the local motion of individual modules? How to communicate when the module connections are changed unexpectedly? How to coordinate the motion of each module to achieve given mission? To solve these problems, a significant amount of work has been devoted in the control methods for the self-reconfigurable robot. Yim[1] used a pre-computed gait control table to control the modules. Since it is based on centralized control, the system is not robust if the central controller fails. As for the distributed control method, Stoy[4-5] designed a role-based control algorithm for periodic locomotion of acyclic configurations. Shen and Salemi [6-9] proposed to use hormones to synchronize the modules and achieve consistent global locomotion. For the lattice-based robots, Butler[10] also presented a set of generic locomotion algorithms for modular robots based on cellular automata.

Although these methods were successful to achieve locomotion for self-reconfigurable robots, some limitations still remain. First, there is a lack of communication protocol to prevent messages from circulating in cyclic configurations. An earlier attempt[6] only yielded low performance for it mixed the low-level control with the high-level communication. Second, all the control rules are expressed in tables or programs without any mathematical models. As such, most existing approaches require extensive knowledge to program and operate a configurable robot and the control rules are difficult to understand and cannot be incrementally augmented for new configurations. The mathematical model developed here is to provide a foundation to alleviate these problems.

Integrated from the hormone-inspired control and the modern control theory[15], we propose to use matrix notation and linear algebra to model a reconfigurable robot whose shape may be an arbitrary graph. To break the loops in a cyclic graph, we introduce the concept of virtual disconnections for certain edges in the graph so that the cyclic graph can be viewed as a virtual tree. Messages generated from the root of this virtual tree can propagate down to each module through the graph without re-circulating. If damage has occurred, the modules will automatically detect the changes in their local topology and a new virtual tree will be automatically arranged. In this sense, all modules are regarded as homogeneous for the purpose of control, but a generalized discrete state space model is developed to describe how a module selects its action and modifies the received hormones based on the hormone input and its local state.

The state space model with the communication protocol has inherited all the features of hormone-inspired control and it is robust to module failures, scalable to shape changing, and able to coordinate asynchronously without any global timer. Moreover, the integration with the modern control theory has brought out the following new advantages:

1. Analysis and generality: The state space model gives a mathematical description of the module's dynamic behavior. Thus, we can use classic control theory tools, such as stability estimation, frequency domain analysing etc, to analyze the reconfigurable robot. This generalizes the hormone-inspired control for other types of self-reconfigurable systems in general, such as distributed sensor network, swarm robotic systems, etc.
2. Scalable to multi-hormones: The model is not only adaptable to dynamic configuration changes such as module addition, deletion or rearrangement, but also scalable to multi hormones. Because the state-space model can represent a MIMO system, the structure of the state variables remains the same whether there are 2, 20 or 100 hormones.
3. Predictable and controllable: With the state space model, the hormone inputs can be controlled to create states and outputs as desired. Moreover, we can predict each module's action in the next few steps, and thus predict the global behavior of the robot.

4. Flexible: With the abstracted subsystem of each module, we can combine some connected modules to compose a larger subsystem and easily get its new state space model.
5. Efficient: As a expansion of the adaptive Communication protocol[6] , the proposed idea of virtual disconnection abstracts the low-level module control into the high-level module coordination for both cyclic and acyclic configuration, and simplifies the adaptation for the dynamic topological changes.

The remainder of this paper is organized as follows: Section II deals with the synchronization and coordination among the modules. Section III develops the state space model. Section IV describes our experiments in simulation as well as on a new self-reconfigurable robot called SuperBot[13]. Section V concludes the paper with some future work.

II. Synchronization and Communication

As stated above, the self-reconfigurable robot is constituted by various modules, where individual module is deemed as the subsystem that runs the same program. The global behavior of such a robot emerges from the interconnection and communication between individual modules. To achieve the desired global behavior, two main tasks need to be solved: A communication protocol for module coordination to achieve the global behaviour, and a model to describe the local behavior of individual modules. This section deals with the first task, and section III discusses the second.

A. Virtual Disconnection

In our communication protocol, a module is not addressed by any global ID, but by its local topology and function. Thus, a module can automatically switch behavior if its location is changed, and the robot is robust to configuration changes. Before proposing the protocol, we will give the definition of local topology vector[14] to represent a module's location in the robot, and propose the concept of virtual disconnection to prevent message from re-circulating in loops.

Each module has a local topology to represent how its connectors are connected to the connectors of its neighbors. Suppose that every module has n different connectors, and every connector can either have no connection or connect to any of the n connectors of another module. Since each of the n connectors can have n+1 possible connection choices, the total number of local topology for every module is $(n+1)^n$. As n increases, $(n+1)^n$ will increase exponentially. Therefore, a simple representation of local topology using numbers[6] would be too expensive, and we propose a vector format as follows. Consider a module that has six connection side as front(F), left(L), right(R), up(U), down(D), back(B). The local topology is represented by a vector that specifies the connection direction of the six connectors in the order of "F L R U D B". If some connectors have no connection, its connection is N. For example, if a module's topology type is [N,U,D,N,N,F], it means that the module has a topology as link[F]=N, link[L]=U, link[R]=D, link[U]=N, link[D]=N, link[B]=F. Here, link[x]=y, where (x=F,B,U,D,L,R; y=F,B,U,D,L,R,N), means that the modules' x connector is connected to the y connector of another module.

If a robot's configuration is acyclic, then it can be viewed as a tree with a root selected in a distributed fashion[2,3]. Each node denotes a module and the link corresponds to a physical connection between two modules. A hormone can propagate through the entire tree and cause different modules to react differently.

In a robot's configuration contains loops, then a scheme is needed to prevent messages from circulating indefinitely. Since there isn't any identifier for each module, it is challenging to decide where to stop the hormone transmission. Inspired by the spanning tree algorithm, here we propose the idea of virtual disconnection to break communication loops.

Starting from an arbitrary module as the root, we use breadth-first-search (BFS) to explore all the modules and get a spanning tree, and then, virtually cut all the connections that are not in the tree edge. This is like lifting the entire graph by its root, and cutting all the loops at their bottom. Here, virtual cut means that we change the corresponding elements in the local topology vector from uppercase to lowercase, so that the modules will not communicate between this physically connected link. For example, given the robot configuration as Fig. 1, we start from module 1 and traverse all the modules by BFS. Three links that are not in the traversal path (shown by the crosses in Fig. 1) are virtually cut, and thus all the cycles in the robot are broken. Accordingly, the local topology of module 2, 3, 4, 5 and 6 will be changed accordingly into [N,N,N,R,F,f], [b,D,b,N,N,N], [D,N,B,N,F,r], [N,N,N,R,r,N], [D,N,d,N,N,N]. Based on the changed local topology, it can be assured that the hormone message will not propagated through these links. For example, when the hormone message was propagated down from module 4 to 5, module 5 will check its local topology as [D,N,d,N,N,N]. Since link[D] is r instead of R, module 5 will know that its D connector is connected to another module's R connector, but the message can not be sent out through it. So, there won't be any direct communication between module 5 and module 6, and thus the hormone message will not circulating any more.

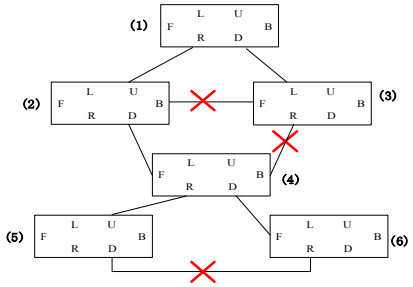

Fig. 1 Robot configuration with loops

B. Communication Protocol

Although the robot is viewed as a tree, it is not required to globally build the tree for communication. What we need is a selected root and the local topology information for each module. Every module is regarded as an independent subsystem that runs the same program and works in a distributed way, as shown in Fig. 2. Once receiving a hormone, a module will check its topology to find all its the connectors

that have upper case values but N, and send out the message to its neighbours except its parent, namely the one from which the hormone is received. Every module will be automatically included in the tree, and the hormone signal will stop at the leaf modules. Moreover, since every module in the tree has only one parent, and there is no loop in the tree, the hormone will be received once and only once by any module.

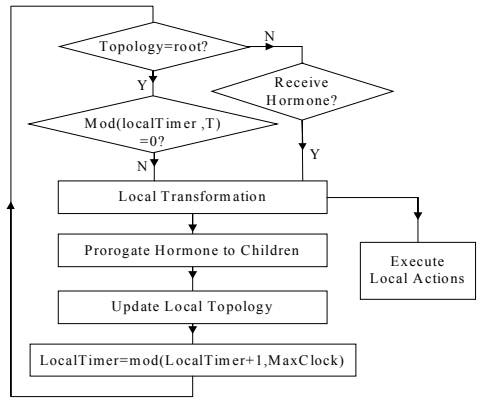

Fig. 2 Communication Protocol

Basically, the main procedure of a module is a loop of receiving/generating and sending hormones, and executing local actions First, the module will check its topology to see whether it is the hormone generator, or the root of the tree. The head module will generate a hormone once its local timer reaches a multiple of predefined time period T, while non-head modules will have actions triggered once receiving hormones. Different modules will react differently according to the rule of local transformation, which is described in section III..

Every time during hormone propagation, the module can also detect its local configuration change and will update its topology according The updating time depends on the communication speed between modules. The capability of Local topology updating makes the robot robust and flexible to its configuration change, since the tree's structure is automatically updated once the local topology information is changed. For example, if a caterpillar robot is bifurcated into two parts, the two disconnected module will quickly update its topology. One of them will become the head based on its new topology, and start to generate another sequence of hormone. So, there comes two trees, and the two small caterpillars will keep moving individually. Moreover, since modules coordinate their actions by the local topology, not by assigned IDs, a module can automatically change its action if its position is changed, and the hormone message will be directed to the modules having a specific role rather than to a specific module. So, if we exchange the position of the modules, the robot will still work well.

III. Modern Control for Modules

This section is to develop the mathematical model to describe local transformation for each module. How hormones trigger actions? How the hormone is modified during its propagation? How the local state information is changed during the process? To solve these problems, we develop a mathematical model to explore the modules' local behaviors.

Since no prior model of the modules is available, we can use the black-box approach to analyze the system, and get its local transformation as Fig. 3. Each module runs the same control protocol, but reacts differently to the received hormone or sensor information based on its topology connection, and state information etc. Without lost of generality, we assume the sensor part is empty in this paper. Since the module has multiple inputs and multiple outputs, a matrix transfer element is needed to relate the input and output. So, instead of input-output transfer functions model, the discrete state space model is chosen to describe the module system.

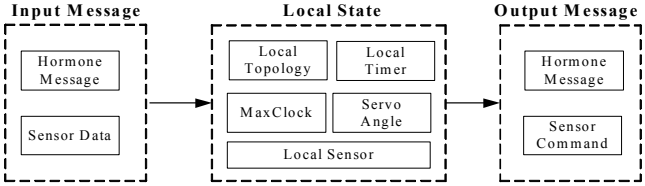

Fig. 3 Local transformation of each module

The first step to construct the state space model is to define the input, output and state vectors. According to Fig. 3, we define U=[Hormone] (the received hormone data) as the input vector, and Y=[Hormone] (the emitted hormone) as the output vector. As for the state vector, it is defined as X=[DOFs, Hormone], where DOFs are the desired motor angles, and Hormone is the hormone message sent out in last step. The reason to remember hormones history is that the hormones are sent out in sequence, and the hormone to be generated is decided by the one sent out last time. Some other state information, like the MaxClock, LocalTimer etc, is not the element of state vector since they are not used for the input and output transformation but for the communication protocol.

The state space representation of each module can be as

$$X_k=AX_{k-1}+BU_k, \quad Y_k=CX_k+DU_k \qquad (1)$$

where Y=[Hormone], U=[Hormone], X=[DOFs, hormone]. For each module, once it runs into the "Local Transformation" process in Fig.2, it will act according to (1). Then, it will send out the modified hormone Y_k, and move the motors to the target position according to the DOFs in X_k.

There are two special attributes for the state space model in (1). First, k is not a time step, but the number of steps that has received or sent out hormones. It is meaningless to relate k with time the local timer because different modules are asynchronous with a global timer. Second, the hormone signal is a command to the modules rather than an energy signal in the regular control system.

A. Algorithm to construct the state space model

The process to construct the state space model for different locomotion is as below:

1. Virtually cut the configuration loops if exist, figure out the local topology for each module, decide the topology of head module, and get the tree for the robot.

2. Analyze the robot locomotion, and list a RULEBASE table to specify how the modules act based on its local state and received hormones. The RULEBASE table is in the form

of table I, where p is the number of different topologies in the robot, m is the hormone sequence length, mT is equal to the MaxClock, and the target DOFs are servo angles for all the modules with n degree-of-freedom. DOF_{ij}^{k} denotes the ith motors desired angle of the module with topology V_k after it receives hormone H_j. Please refer to [6] on how to construct the RULEBASE table in detail.

TABLE I RULEBASE TABLE

Module Topology	Local Timer	Received Hormone Data	Target DOFs	Sent Hormone Data
V_1	T		DOF^1_{11} … DOF^1_{n1}	H_1
	…			
	mT		DOF^1_{1m} … DOF^1_{nm}	H_m
.....				
V_p		H_1	DOF^p_{11} … DOF^p_{n1}	H_i
	…			
		H_m	DOF^p_{1m} … DOF^p_{nm}	H_j

3. Determine the hormone values by 1-of-n encoding method. Since a hormone can take on m different values, we make it a binary vector of length m. The relevant bit on is turned on, and all the others are left off. Therefore, the dimensions of X, Y and U are m+n, m and m respectively.

4. Suppose the topology vector V_1 is the topology type of the head module. Since the hormone is generated in sequence, we can get the dynamic model and measurement equation for the root module as

$$X_k=AX_{k-1}+BU_k\ ,\quad Y_k=CX_k+DU_k \tag{2}$$

Where $A=\begin{bmatrix} 0_{(n+m)xn} & \begin{matrix} DOF^1_{11} & DOF^1_{1m} \\ \cdots & \\ DOF^1_{n1} & DOF^1_{nm} \end{matrix} \\ \hline \begin{matrix} 0^T \\ \hline I_{(m-1)\times(m-1)} \end{matrix} & \begin{matrix} 1 \\ \hline 0 \end{matrix} \end{bmatrix}_{(n+m)\times(n+m)}$ $B=0_{(n+m)\times m}$,

$C=\begin{bmatrix} 0_{m\times n} & I_{m\times m} \end{bmatrix}_{m\times(n+m)}$ $D=0_{m\times m}$.

5. For modules with other local topology V_i, since their actions are determined by the received hormone, the state space model would be like

$$X_k=AX_{k-1}+BU_k\ ,\quad Y_k=CX_k+DU_k \tag{3}$$

where $A=0_{(n+m)\times(n+m)}$ $B=\begin{bmatrix} \begin{matrix} DOF^i_{11} & \cdots & DOF^i_{1m} \\ & \cdots & \\ DOF^i_{n1} & \cdots & DOF^i_{nm} \end{matrix} \\ \hline \begin{matrix} 0^T & I_{hxh} \\ \hline I_{(n-h)x(n-h)} & 0 \end{matrix} \end{bmatrix}_{(n+m)xm}$

$C=\begin{bmatrix} 0_{m\times n} & I_{m\times m} \end{bmatrix}_{m\times(n+m)}$ $D=0_{m\times m}$

The number of h in B determines how the hormone message is modified during propagation, namely the step difference between the emitted hormone and the received hormone in the hormone sequence is h. By remembering the last issued hormones in the state vector X, a module can easily switch to the transformation model of (2) if its role has been changed to the head.

6. By Comparing (2) and (3), we find that matrices C and D in the measure equations are equal. So, the general state space model of (1) can be rewritten as

$$\begin{bmatrix} X_k \\ Y_k \end{bmatrix}=\Phi_{(2m+n)\times(2m+n)}\begin{bmatrix} X_{k-1} \\ U_k \end{bmatrix} \tag{4}$$

where $\Phi_{(2m+n)\times(2m+n)}=\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]=\left[\begin{array}{c|c} A & B \\ \hline 0_{m\times n} \quad I_{m\times m} & 0_{m\times m} \end{array}\right]$

7. As we can see, the modules with different topology will respond differently to the incoming hormone. So, the topology can be regarded as a selector for the transformation matrixes of the dynamic equation. A more general equation for the hormone control is

$$\begin{bmatrix} X_k \\ Yk \end{bmatrix}=\begin{bmatrix} T_1 & \cdots & T_n \end{bmatrix}\begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_n \end{bmatrix}\begin{bmatrix} X_{k-1} \\ U_k \end{bmatrix} \tag{5}$$

where Φ_i is the transformation matrix for the modules with topology V_i, and $T_i=I_{(2m+n)\times(2m+n)}$ if the modules' topology is topology V_i, otherwise $T_i=0_{(2m+n)\times(2m+n)}$.

Equation (5) gives a general state space model that describes the behaviors of all the modules in the robot. With this mathematical model, we can use many existing control theory tools to analyze all the modules' behavior, and get a more thorough understanding of it. For example, to achieve some specific locomotion, we can control the hormone inputs to get the desired DOFs in the state vector X_k according to (5)

B. Locomotion examples

To illustrate the algorithm to construct the state space model, we will consider some locomotion examples. Here, all the modules are supposed to have 3 rotational degrees of freedom as pitch, yaw and roll.

First, let us consider the crawling gait. The robot configuration is as Fig. 4. Intuitively, the legs should be lifted when moving forward and touching the ground when moving backwards. The spine module between two pairs of legs should bend from side to side to increase the length of each step. The rules for the robot to walk are listed as table II. Because the hormone data is changed during propagation, the two left legs will have opposite actions triggered by different incoming hormones, even though they have same topology. The same is for the front and back right legs.

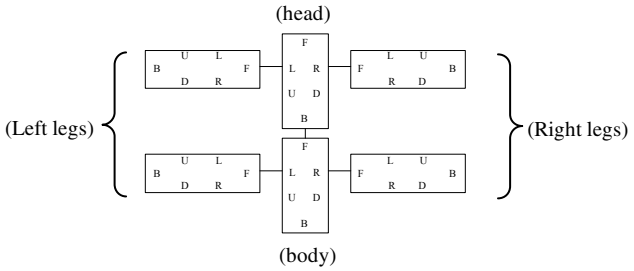

Fig. 4 Quadruped robot configuration

TABLE II THE RULEBASE FOR A CRAWLING GAIT

Module Topology	Local Timer	Received Hormone Data	Target DOFs [pitch, yaw, roll]	Sent Hormone Data
[N,F,F,N,N,F]	T		[0, 0, 0]	H_1
	2T		[0, 25, 0]	H_2
	3T		[0, 0, 0]	H_3
	4T		[0, -25, 0]	H_4
[B,F,F,N,N,N] or [B,F,F,N,N,N]		H_1	[0, 0, 0]	H_3
		H_2	[0, -25, 0]	H_4
		H_3	[0, 0, 0]	H_1
		H_4	[0, 25, 0]	H_2
(left leg) [L,N,N,N,N,N]		H_1	[-45,40,0]	
		H_2	[-60, 0, 0]	
		H_3	[-45, -40, 0]	
		H_4	[-30, 0, 0]	
(right leg) [R,N,N,N,N,N]		H_1	[-45, 40, 0]	
		H_2	[-30, 0, 0]	
		H_3	[-45, -40, 0]	
		H_4	[-60, 0, 0]	

According to our algorithm, the transformation equations for different local topology (we will use TOP to denote Local topology in the following) is:

TOP=[N,F,F,N,N,F](head)

$$X_k = \left[\begin{array}{c|cccc} & 0 & 0 & 0 & 0 \\ & 0 & 25 & 0 & -25 \\ & 0 & 0 & 0 & 0 \\ 0_{7\times3} & 0 & 0 & 0 & 1 \\ & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \end{array}\right] X_{k-1}$$

$$Y_k = [0_{4\times3} \mid I_{4\times4}] X_k$$

TOP= [L,N,N,N,N,N](left legs)

$$X_k = \begin{bmatrix} -45 & -60 & -45 & -30 \\ 40 & 0 & -40 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} U_k$$

$$Y_k = [0_{4\times3} \mid I_{4\times4}] X_k$$

TOP= [B,F,F,N,N,N] or [B,F,F,N,N,F] (body):

$$X_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -25 & 0 & 25 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} U_k$$

$$Y_k = [0_{4\times3} \mid I_{4\times4}] X_k$$

TOP=[R,N,N,N,N,N](right legs)

$$X_k = \begin{bmatrix} -45 & -30 & -45 & -60 \\ 40 & 0 & -40 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} U_k$$

$$Y_k = [0_{4\times3} \mid I_{4\times4}] X_k \qquad (6)$$

where $X=[\text{pitch,yaw,roll},H_1,H_2,H_3,H_4]^T$, $Y=[H_1,H_2,H_3,H_4]^T$, $U=[H_1,H_2,H_3,H_4]^T$.

The above equations are robust to changes in robot configurations. The robot can have any number of modules, as long as they are arranged in the same way with four topologies of head, left leg, right leg and body. One can also dynamically add or delete legs from this robot, or cut the robot into halves. All the modules will automatically detect their local topology changes, and act according to above control equations.

To show that the state space model can be applied to the robot with loop configuration, here we will show another application to rolling track. The robot's configuration is shown as Fig. 5. According to the virtual cut idea in section II, we virtually cut the connection between module 1 and 8, and change their topology to be [b,N,N,N,N,F] and [B,N,N,N,N,f] respectively. Choosing module 1 as the hormone generator, we get the RULEBASE as in Table III.

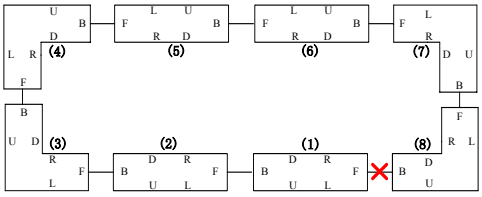

Fig. 5 Rolling track configuration

TABLE III THE RULEBASE FOR A ROLLING TRACK GAIT

Module Topology	Local Timer	Received Hormone Data	Target DOFs [pitch, yaw, roll]	Sent Hormone Data
[b,N,N,N,N,F]	T		[0, 0, 0]	H_1
	2T		[0,0, 0]	H_2
	3T		[90, 0, 0]	H_3
	4T		[90, 0, 0]	H_4
[B,N,N,N,N,F] or [B,N,N,N,N,f]		H_1	[0, 0, 0]	H_2
		H_2	[90,0, 0]	H_3
		H_3	[90, 0, 0]	H_4
		H_4	[0,0, 0]	H_1

So the state space equations for rolling track can be:

TOP=[b,N,N,N,N,F] (head):

$$X_k = \left[\begin{array}{c|cccc} & 0 & 0 & 90 & 90 \\ & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 \\ 0_{7\times3} & 0 & 0 & 0 & 1 \\ & 1 & 0 & 0 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \end{array}\right] X_{k-1}$$

$$Y_k = \left[\begin{array}{c|cccc} & 0 & 0 & 0 & 1 \\ & 1 & 0 & 0 & 0 \\ 0_{4\times3} & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \end{array}\right] X_k$$

TOP= [B,N,N,N,N,F] or [B,N,N,N,N,f] (non-head)

$$X_k = \begin{bmatrix} 0 & 90 & 90 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} U_k$$

$$Y_k = \left[\begin{array}{c|cccc} & 0 & 0 & 0 & 1 \\ & 1 & 0 & 0 & 0 \\ 0_{4\times3} & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 0 \end{array}\right] X_k \qquad (7)$$

where $X=[\text{pitch,yaw,roll},H_1,H_2,H_3,H_4]^T$, $Y=[H_1,H_2,H_3,H_4]^T$, $U=[H_1,H_2,H_3,H_4]^T$.

When the hormone information reaches module 8 through its F connector, module 8 will check its local topology to explore its connection with other modules. Since link(B)=f, no message will be sent out through its B connector. So, the hormone message is stopped, and won't circulate indefinitely.

IV. SIMULATION AND EXPERIMENTAL RESULTS

Our control methods have been implemented both in a physics-based GALINA simulation environment and our self-reconfigurable robot called SuperBot.

A. SuperBot Module

SuperBot[13] is a new reconfigurable robot combines the flexibility of M-TRAN[11], ATRON[12] and CONRO[6] into one. As shown in Fig. 6, each module has two segments joined by a central rotation. Each segment has three universal connectors on three different sides so that a module can connect to others in any of the six directions in 3D (up, down, front, back, left, and right). The module has 3DOF, two pitch/yaw connected by a roll, and can pitch and yaw for up to 180° and roll for 90°

in each direction. The middle roll enables a module to change its moving direction on its own, and have the flexibility for many different locomotion and reconfiguration without any helper modules. Every module is able to detect if a connector is docked or not, and communicate with up to six neighbour modules with a high-speed and reliable communication link.

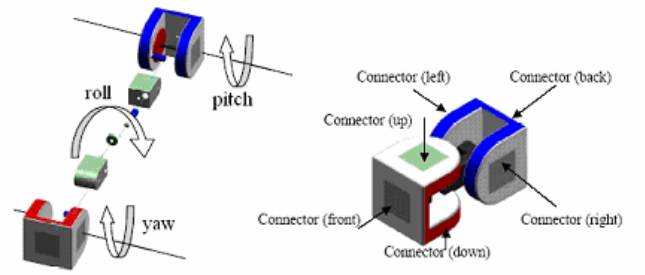

Fig. 6 Schema of SuperBot module

B. *Simulation Result*

We developed a simulation software called GALINA to create SuperBot modules and the testing environments as realistic as possible. To emulate concurrent execution of control programs for different modules and resulting communication issues, we use the threads mechanism to emulate simultaneously running modules in GALINA.

Multi-locomotion, including caterpillar, sidewinder, rolling track, and crawler gait etc, is implemented. In the implementation of all these behaviors, all the modules are loaded with the same program for the communication protocol and control algorithm. What we changed is just the transformation matrix in the state space model.

To test the ability of self-healing, we deliberately "cut" an 8-module caterpillar, a sidewinder, and an eight-legged crawler into two halves. Without any modification to the program, all these segments adapt to the new configuration immediately and continue to move as two small caterpillars, sidewinders, or crawlers. This is because all the modules can find out its topology change immediately, and adapt to the new control transformation autonomously.

C. *Experimental Results*

We have also loaded our program into the real SuperBot modules for different locomotion. As of the writing of this paper, only two SuperBot modules are built, so the possible locomotion is very limited. With one module, we can make it move and flip. After connecting the two modules in a line, we can make it have different motion, like caterpillar, creep, or S moving etc. The speed of creep can reach 12.5cm/s. By changing the head module, the robot can move forward or backward. Both of the two modules are loaded with the same program, and running as autonomous system without any off-line computation.

For the video of the behaviors in simulation and in real robot, please visit http://www.isi.edu/robots/superbot/movies/

IV. Conclusion

Inspired by the modern control theory, this paper presents a mathematical model for the hormone control of reconfigurable robot, and improves the communication protocol for both cyclic and acyclic configuration with the idea of virtual disconnection. The state space model give a general mathematical description of the module's dynamic behavior, and make the hormone-inspired control method applicable to other self-reconfigurable systems in general. Effectiveness of our algorithm is demonstrated by our SuperBot modules both in physics-based simulation and in real robots. Multimode locomotion modes are implemented without increasing the hardware or software complexity. One of our future work is to consider the robustness when hormone message is lost between two connected modules. Another future work can be adding sensor information in the control so that the robot can react to the environment with multiple hormones,.

References

[1] M. H. Yim, Y. Zhang, D. G. Duff, "Modular robots," *IEEE Spectrum*, 39(2), pp. 30-34, February 2002.

[2] B.Salemi, P. Will, W-M. Shen, "Distributed Task Negotiation in Self-Reconfigurable Robots," International Conference on Intelligent Robots and Systems. Las Vegas, October 2003

[3] B. Salemi, P. Will, and W.-M. Shen, "Distributed Task Negotiation in Modular Robots," Robotics Society of Japan, Special Issue on "Modular Robots", 2003.

[4] Stoy, K, W.-M. Shen, P. Will, "Using Role-Based Control to Produce Locomotion in Chain-type Self-Reconfigurable Robots," *IEEE Transactions on Mechatronics*, 7(4), 410-417, Dec. 2002.

[5] Stoy, K., W.-M. Shen, P. Will, "Implementing Configuration Dependent Gaits in Self-Reconfigurable Robots," International Conference on Robotics and Automation, Taiwan, 2003.

[6] W.-M. Shen, B. Salemi, and P. Will, "Hormone-Inspired Adaptive Communication and Distributed Control for CONRO Self-Reconfigurable Robots," *IEEE Transactions on Robotics and Automation*, 18(5), October 2002.

[7] W.-M. Shen, B. Salemi and P. Will, "Hormone for self-reconfigurable robots," Proc. Intl. Conf. Intelligent Autonomous Systems, IOS Press, pp. 918-925, 2000.

[8] W.-M. Shen, Y. Lu and P. Will, "Hormone-based control for self-reconfigurable robots," Proc. Intl. Conf. Autonomous Agents, 2000.

[9] B. Salemi and W.-M. Shen. "Distributed Behavior Collaboration for Self-Reconfigurable Robots," International Conference on Robotics and Automation, 2004.

[10] Butler, Z., K. Kotay, D. Rus, K. Tomita. "Generic Decentralized Control for a Class of Self-Reconfigurable Robots," Intl. Conf. on Robotics and Automation. 2002. Washington DC.

[11] S. Murata, E. Yoshida, etc, "Hardware Design of Modular Robotic System," Proc. of 2000 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, 2000.

[12] MW Jorgensen, EH Ostergaard, HHLund, "Modular ATRON: Modules for a self-reconfigurable robot," proceedings of IEEE/RSJ International Conference on Robots and Systems, 2004.

[13] W.-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, J. Venkatesh, "Mutlimode locomotion via SuperBot reconfigurable robots," International Conference on Robotics and Automation, 2006.

[14] F. Hou, W.M. Shen, "Hormone-inspired Adaptive Distributed Synchronization of Reconfigurable Robots," The 9th International Conference on Intelligent and Autonomous Systems, 2006.

[15] R.C. Dorf, *Modern control systems*, 9th ed., NJ : Prentice Hall, 2001.