Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

A Decentralized Control Framework for Modular Robots

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

Intelligent Systems Laboratory
The University of Hong Kong
Pokfulam Road, Hong Kong SAR
E-mails: aux1496@hkusua.hku.hk

Abstract

Distributed control paradigm offers robustness, scalability, and simplicity to the control and organization of module based systems. MSR (Modular Self-Reconfigurable) robot is a class of robot that best demonstrate the effectiveness of distributed systems as all modules in the robot are individuals that perform their own actuation and computation; the behavior of the complete robot is a collective behavior of all independent modules.

In this paper, a general control framework, named General Suppression Framework, is proposed and a distributed control system based on the framework is presented. The control system is designed to control a set of MSR robots configured into a planar manipulator arm. All modules in the manipulator arm contain their own processing and actuation units, which allow them to evaluate and react to the environment independently. The modules can perform passive communication with their immediate neighbors and can exhibit aggressive or tolerant behavior based on the environment change to generate emergent group behaviors. A simulation program is developed to demonstrate the effectiveness of the distributed system in controlling the module based planar manipulator arm.

Keywords

Modular Robots, Artificial Immune Systems, Distributed Control, Multi-agents.

1. Introduction

Distributed module based agent systems [19] has acquired a high level of interest among researchers attempting to develop a robust and scalable system. Artificial Immune Systems (AIS) [7] that function like their biological counterparts is inspiring many research activities in different disciplines. Lau et al. [9] developed a control framework to improve efficiency and robustness of a distributed material handling system. Segel et al. [12] examined how biological ideas can help to solve engineering problems, and inversely how the artificial system can inspire new conjectures to unrecognized methods by which the immune system is organized. de Castro et al. [6] presented the application of AIS in computer network security, machine learning, and pattern recognition. Tarakanov et al. [15] introduced Immunocomputing as a new computing approach based on the fundamental concept, the formal protein (FP). The virtual world, Tierra, created by Ray [11] contains virtual viruses and hosts that develop artificial immune systems to defend them. This work may not directly classify under AIS, but also share similar conceptual ideas. The term AIS was even mentioned in a novel [17] as early as 1992.

This paper presents the development of a new distributed control framework, named General Suppression Framework. The framework inspired by the suppression hypothesis in discrimination theory has a *Suppression Modulator* that contains many suppressor cells with different functions. To evaluate the effectiveness of the suppression mechanism, a seven-module MSR robot configured in the form of a hyper-redundant planar manipulator is constructed in MATLAB for experiment.

This paper proceeds as follow. **Section 2** provides a brief introduction to MSR robots and explains the unique difficulties in controlling a MSR manipulator arm against a purpose-built robot arm. **Section 3** introduces the suppression hypothesis in discrimination theory and presents the General Suppression Framework. **Section 4** discusses how the modules generate useful emergent behaviors using only simple local rules and how the suppression mechanism improves the efficiency of the system. The design of the simulation and setup of the experiment are also briefly described. **Section 5** concludes the work in this paper and discusses future works to be taken.

2. Modular Self-Reconfigurable Robots

Modular Self-Reconfigurable (MSR) robots [5][8][19] are robots made up of many identical but independent mechatronic modules that can be disconnected and reconnected autonomously and to rearrange into different structures that can facilitate the robot to complete its tasks more effectively. Each individual module is a self-contained unit equipped with its own processor to control the module's movement and to facilitate communication with neighboring modules.

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

In spite of all the advantages MSR robots has to offer, there are many challenges to overcome before these robots can have practical applications outside of research. One of the biggest challenges is to develop decentralized control system that does not require a designated leader for coordination, so the robot will not have a concentrated weak point (i.e. the leader) of failure and can achieve total homogeneity in module design.

Many researchers around the world have examined different control algorithms. Butler [4] et al. at Dartmouth College proposed a distributed goal recognition technique called "Trace" to generate global shape using only local knowledge and local communication. Unsal et al. [16] at Carnegie Mellon University have presented a multi-layered planner for the motion of modules with a combination of distributed approaches at the high-level with low-level pre-defined rules for trajectory motions. Bojinov et al. [3] at PARC applied multi-agent control to randomly generate stable structures based on local rules.

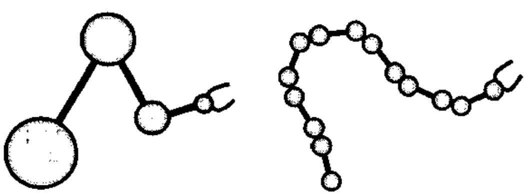

Figure 1: A purpose-built robot arm (Left) has larger torque at the base joint and smaller torque near to the end-effector. A MSR robot arm has identical joint-torque at each joint.

The primary focus of this research is to exploit the T-cell suppression mechanism to control a seven-link hyper-redundant manipulator arm. Conventionally a purpose-built articulated robot arm would have joints designed to provide the torque necessary for the task. i.e. a robot arm for lifting heavy parts would have a high-torque joint near to the base to manipulate the load and the links, whereas, the joint-torque near to the end-effector would be smaller because it is designed to carry the load only. Such robot arms are very unlikely to run out of joint-torque under designed working conditions (see Figure 1). However, when a MSR robot configures into an articulated arm, the joint acting as the "base" can only exert as much torque as the joint acting as the "end-effector". Therefore the decentralized system must prevent each joint from exceeding its torque-limit while controlling individual modules to generate emergent behavior as a whole to reach the objective.

3. General Suppression Framework

Mother Nature has inspired many fascinating systems to solve engineering problems [14], artificial neural networks enable systems to learn effectively, and genetic algorithm [2] creates diversified answers for complex problem. Human Immune System is an extremely effective system that can identify abnormal activities, solve the problem using existing knowledge, and generate new solutions for unseen events; in short it is a network of players who cooperate to get things done [13].

Immuno discrimination is perhaps the most unique and important function in the immune system; its duty is to discriminate Non-Self Cells from Self Cells. Self Cells are the good cells that exist and work inside our body. Non-Self Cells are external elements that does harm to the system (antigen). The distinction and the recognition of foreign antigen is done by B-Cells and T-Cells, which allows the system to identify harmful molecules to response (to kill) and leave the good molecules (self-cells) untouched.

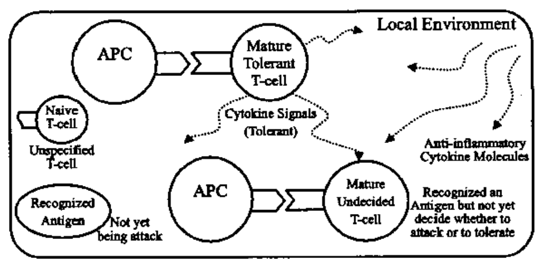

Figure 2: Cell Suppression Mechanism.

The *General Suppression Framework* developed in this research is based around the analogy of the suppression hypothesis in the discrimination theory [1]. When a T-cell receptor binds to a peptide with high affinity presented by an APC (Antigen Presenting Cells), the T-cell recognized the antigen become mature and it has to decide whether to attack the antigen aggressively or to tolerate it in peace. An important decision factor is the local environment within which the T-cell resides. The present of inflammatory cytokine molecules such as interferon-gamma (INF-γ) in the environment tend to elicit aggressive behaviors of T-cells, whereas the anti-inflammatory cytokines like IL-4 and IL-10 tend to suppress such behavior by blocking the signaling of aggression. In brief, a T-cell matured after recognizing an antigen does not start killing unless the environment also contains encouraging factors for doing so.

In addition, after a mature T-cell developed a behavior, it will emit humoral signals to convert others to join. The mechanism is illustrated in Figure 2.

Our analogy infers each module (joint) of the manipulator is an autonomous T-cell that continuously reacts to the changing environment and affects the functioning of other cells through the environment. The framework consists of five major components. The most notable difference between the natural mechanism shown in Figure 2 and the proposed framework shown in Figure 3 is that the T-cell's functions are divided into three separate components, the *Affinity Evaluator*, *Cell Differentiator* and the *Cell Reactor*. Delegating the three unique functions into separate components enables the system to be organized in modular manner and that when programming for an application, the result and effect of each component can be traced easier. The key functions of the five components are explained below.

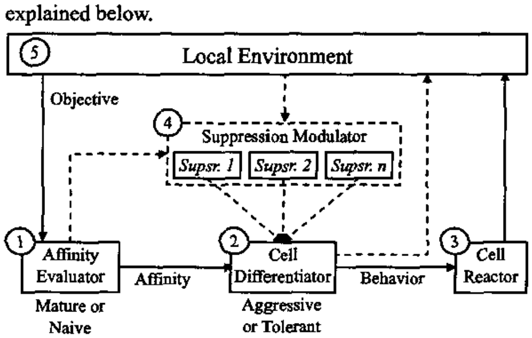

Figure 3: The General Suppression Framework. Dashed lines represent humoral signal transmissions, where solid lines represent cellular signals.

(1) *Affiliation Evaluator* – is responsible for determining the affinity level according to the torque status of the neighboring modules. The affinity is highest when the torque load of both neighboring modules plus the module itself is all under working limit, otherwise the affinity is low. The function of this component is similar to the immune discrimination function, which helps to differentiate between self and non-self cells. *Affinity Evaluator* can affect the *(2) Cell Differentiator* in two ways, the first is to directly send an cellular signal that indicates the affinity level, the second is to send humoral signals to *Suppressor Cells* in the *(4) Suppression Modulator*. The cellular signals give a spontaneous effect that last only one cycle, whereas humoral signals can remain effective until another humoral signal is released to reverse or to neutralize the effect. Notice the information about the neighboring modules is obtained from the *(5) Local Environment*, this infers that each module is an independent unit and everything else, to the module's concern, belongs to the external environment.

(2) *Cell Differentiator* – is responsible for deciding the module's behavior by evaluating the affinity index from *(1) Affinity Evaluator* and the suppression index from *(4) Suppression Modulator*. In general, the cell becomes aggressive if the affinity index is high and become tolerant if the affinity index is low. However, the suppression index can act as a suppressant to force the (3) *Cell Differentiator* to become tolerant even when the affinity index is high.

The component can also sent humoral signals directly to influence the *(5) Local Environment*. The functioning of the *(2) Cell Differentiator* is similar to the cell differentiation mechanism, in which cells develop aggressive or tolerant behavior in response to the type of cytokines present in the environment. When activated, these cells also release humoral signals to convert nearby cells to copy their behavior.

(3) *Cell Reactor* – reacts to the cellular signal from the *(2) Cell Differentiator* and execute the corresponding behaviors which take effect in the *(5) Local Environment*. This component is the part that actually does the killing like activated aggressive T-cells in the immune system.

(4) *Suppression Modulator* – is a collection of *Suppressor Cells*, which response to different stimulations and exhibit specific suppression effects to *(2) Cell Differentiator's* decision process. The function of *Suppression Modulator* is comparable to the cytokine signaling mechanism which uses IFN-γ, IL-4, IL-10, etc to perform intercellular communication and to cause the environment to inflame, so to stimulate or suppress aggressive behaviors in the T-cells. In this framework, the *(4) Suppression Modulator* acquire information from the *(5) Local Environment* and the *(1) Affinity Evaluator*. This information is available to all Suppressor Cells within the modulator. There can be 0, 1, 2, 3... *n* number of *Suppressor Cells* and their response to stimulation may also influence other *Suppressor Cells* inside the modulator. When the number of *Suppressor Cell* is 0 (zero), the Modulator still exists but can no longer process information acquired from the *(5) Local Environment*, therefore all information will be feed directly to the *(2) Cell Differentiator*. This experiment contains only one *suppressor cell*, which reads in ls_dst from the light sensor and releases a humoral signal to suppress the *(2) Cell Differentiator* from choosing to become aggressive. This suppression mechanism is simple but very useful for controlling systems that has the tendency to over react to occasional conditions.

(5) *Local Environment* – is where interactions between different components take place. The importance of this component within the framework is to act as an interface that links to the *Global Environment* which contains other *Local Environments* with different sets of *Suppression Modulators*. In addition it provides a theoretical space to integrate the physical objects and the abstract system in an analyzable form.

The *Cell Differentiator* and the *Suppression Modulator* are the heart of the system; the former is responsible for integrating complex information from different sources into simple instruction, whereas the later plays an important role in turning intricate problems into quantitative outputs.

4. Simulation and Implementation

4.1 The Simulation

A simulation program concerning the control of a MSR robot configured in the form of a planar hyper-redundant manipulator arm is constructed using MATLAB to demonstrate how the proposed framework can be applied to control a distributed system.

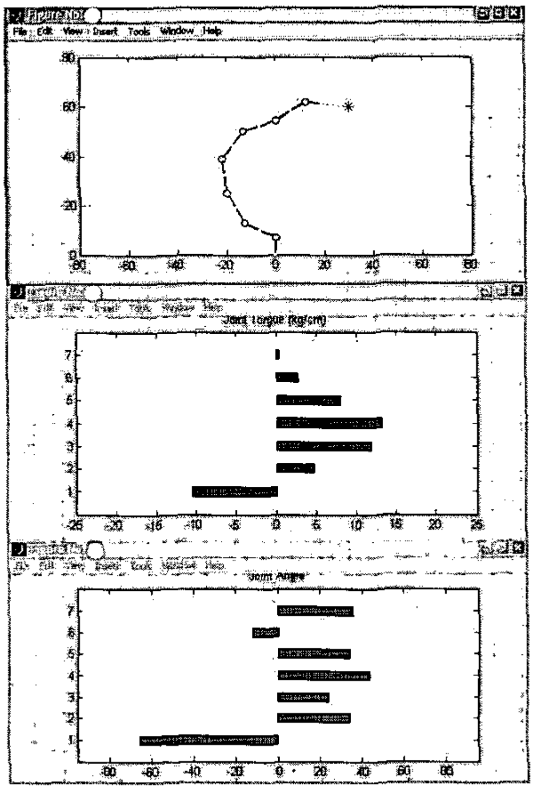

Figure 4: The simulated seven-link hyper-redundant manipulator arm.

The MSR manipulator arm (Figure 4) is constructed of seven modules. Each module has one degree-of-freedom and a motion range of +/- 90 degree from the center line (Figure 5). The system is completely distributed as each module has its own processing unit for logic evaluation and motion control. Each module constantly displays its local angle and joint torque for its immediate neighboring modules to see, inversely the module can read the same information from other modules. However, no active request of information is allowed. The working torque is set at 15 kg.cm (identical to the physical module), the module can adjust the joint angle freely under this limit, beyond which, the joint can no longer adjust itself and will not hold when the torque exceeds 20 kg.cm.

The seven modules are configured in such a way that module-one (md_1) is grounded as the base of the manipulator, followed by md_2, md_3, md_4, md_5, md_6, and md_7 is the end-effector equipped with a light sensor to track the light. The ultimate objective is to minimize the distance between the light source and the light sensor on md_7 without any joint exceeding the maximum torque limit. i.e. 20 kg.cm.

Theoretically, all modules possess the same ability but the sensors they carry differentiate their role within the system. In this simulation, the light sensor carried by md_7 allows the module to determine its distance from the light source, hence, giving it the ability to affect the entire system. However, it should be noted that other modules are also capable of producing similar effect if equipped with the necessary sensor.

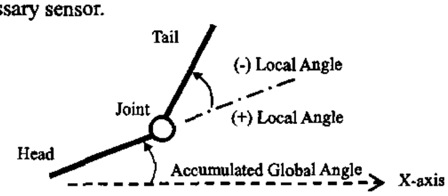

Figure 5: An illustration of a single module in the simulation.

The simulation consists of two separate programs, the main program, **msr_manipulator** contains the control algorithm, and the sub-program, **graphical_outputs** handles the arithmetic and produces the three output windows shown in Figure 4. The first window displays the motion of the manipulator and the location of the light source. The second window displays the joint torque of each module in kg/cm, and the third window displays the modules' local angle ranging from -90 to +90 degree.

The **graphical_outputs** plots and simulate the manipulator's motion by calculating the *Head, Joint* and *Tail* locations and the (AGA) *Accumulated Global Angle* (the angle between the *Head-Joint-Link* and the X-axis). A module's *Head* location is the same as the preceding module's *Tail* location when they are docked, except for md_1's, which is grounded to the origin of the graph space

(0, 0). The AGA of each module is calculated by the preceding module's AGA minus its *Local Angle*. The *Joint* location is found by converting the polar coordinate (σ, ρ) into Cartesian coordinate [x, y], where σ is the AGA in radian, and ρ is the length of the *Head-Joint-Link*, and add the result to the *Head* coordinate. The *Tail* location is found by the same method but substituting *Local Angle* for AGA, and *Joint* coordinate for *Head* coordinate.

When the locations of all components are known, the torque at each joint can be easily calculated by summing the moments created by the modules between itself and the hanging end-effector. In this simulation, the length of each link is 7 centimeter and the mass is 100 gram each.

The suppression index (Supp_idx) grows stronger when away from the origin (light sensor). When a module receives a suppression index, it adds a number (i.e. 10) to the index number and displays the result for the preceding module to consult. Since each module can only read information from the neighboring modules, the suppression index value will become higher as it passes from md_7 to md_1. The initial value can begin with any real number depending on the value of ls_dst, and all modules receiving a suppression index greater than zero will be affected. For example the suppression index in Figure 7(A) begins from 0 at md_7 and reaches 60 at md_1; therefore all modules except md_7 will be affected. In Figure 7(B), the suppression index begins from -50 at md_7; hence, only md_1 will be affected because it has a suppression index greater than 0.

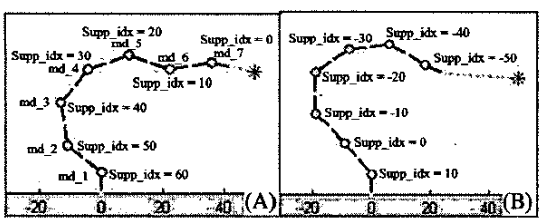

Figure 7: When the value of ls_dst is low (A) the initial suppression index is high. Inversely the initial suppression index is low (B).

The movement of the manipulator arm is indeed a reflection of the emergent behavior of the seven autonomous modules. Each module can exhibit aggressive or tolerant behavior; the decision is governed by two factors, the distance of light source from the light sensor (ls_dst), and the joint_torque of the neighboring modules in relation to the module's own. A module will exhibit tolerant behavior if the joint_torque of itself or one or more of its neighbor has exceeded the limit. In another word, a module cannot exhibit aggressive behavior unless the joint_torque of all two neighboring modules plus itself are below the limit (Table 1). The ls_dst is a signal send from the sensor which acts as suppressant to prevent over aggressive behaviors.

To reach the furthest point, the system needs to adjust all modules' *Local Angles* as near to zero as possible. Following the same logic, when the *Local Angle* of md_7 (the module equipped with the light sensor) is different then zero, its preceding module with aggressive behavior will adjust its *Local Angle* to absorb the difference. For the example in Figure 7(A), the *Local Angle* at md_7 is +30°; md_6 will increase its *Local Angle* slowly to facilitate md_7 to adjust to zero-degree while still pointing at the light source. At the same time, md_5 will adjust its *Local Angle* to absorb the angular differences at md_6. The same happens simultaneously in all other modules. Notice, at this point, md_5 will adjust its *Local Angle* towards the negative direction because currently md_6 has a negative *Local Angle*. The movement is in fact counter productive as md_5 is actually moving the light sensor further away from the light source. However, this counter productive movement is compensated by md_4, md_3, md_2, and md_1 as their *Local Angle*s are all adjusting towards the positive direction. The movement of md_5 will change as md_6 corrects the direction of its *Local Angle* after absorbing the angular difference in md_7.

Joint_torque Neighbor #1	Joint_torque Neighbor #2	Self Joint_torque	Self Behavior	Self Action
Good	Good	Good	Aggressive	Adjust Local Angle
Over limit	Good	Good	Tolerant	Help Neighbor #1
Good	Over limit	Good	Tolerant	Help Neighbor #2
Over limit	Over limit	Over limit	Tolerant	No Possible Action

Table 1: Status of neighbors in relation to self behavior. A module cannot exhibit aggressive behavior when its joint_torque exceeded the limit.

The tolerant behavior is a balance of the aggressive behavior and is the safeguard mechanism that prevents all modules in the system from running out of joint_torque together and put the system into a dead loop. A tolerant module with its joint_torque under working limit, instead of adjusting its *Local Angle* to reach the goal, it would adjust its *Local Angle* towards the direction that alleviates the load on its neighboring modules.

4.2 The Experiment

The purpose of the experiment is to demonstrate how the suppression mechanism in the General Suppression Framework can assist the manipulator arm to reach the goal more effectively. The experiment consists of 10 preset test conditions and a total of 20 runs were done. Each test condition is run twice, once with the Suppression Modulator in effect and once without. The position of the light source is fixed in all ten test conditions but the initial form of the manipulator changes after every two runs.

As expected, when the manipulator first starts to reach the light source form its initial position, it behaved almost identical with or without the Suppression Modulator in effect. However, as the value of ls_dst decreases (approaching the light source), the manipulator without suppression requires many more extra steps to reach the goal. This is because when the light sensor is close to the light source (Figure 7(A)), the modules near to the end-effector (i.e. md_6 and mdc_7) require finer movement to approach the light source, but every minor movement of the modules near to the base (i.e. md_1 and md_2) would cause the end-effector to swing beyond the position of the light source and causes the light sensor to have to turn back and search for the light source again from another direction.

For the manipulator with Suppression Modulator turned on. The problem is easily solved because Suppression Modulator emits a suppression signal to limit the motion of the modules near to the base when approaching the light source. Therefore the modules near to the end-effector can approach the light source in fine motion. The suppression signal grows stronger as ls_dst decreases; hence, more modules near to the ground will slow down or even freeze to allow the modules near to the end-effector to move accurately. The number of steps required for the manipulator with suppression mechanism to reach the goal is approximately 1/2 to 1/3 of the steps required for the manipulator without suppression mechanism.

5. Conclusion and Future Work

We have proposed the General Suppression Framework that emphasizes on the use of suppressor cells to control a modular manipulator arm. The system is highly scalable as all communication is based on interaction with the environment. A MATLAB simulation program has shown the suppression mechanism can effectively prevent the arm from over shooting the goal at close range.

Future work will continue to focus in the development of the suppressor cell functions and the interaction between suppressor cells. We are working to develop other decentralized systems based on the framework to control different type of autonomous robots. Currently, a seven-module MSR manipulator arm is under construction, which will be used to verify the control system in real-time and to help visualize the dynamic effect that was not considered in the simulation.

References

[1] E. Benjamini, G. Sunshine, S. Leskowitz. Immunology: A Short Course. Wiley-Liss, New York, USA, 1996.

[2] P.J. Bentley. Evolution Design by Computers. Morgan Kaufmann, Bath, U.K., 1999.

[3] H. Bojinov, A. Casal, and T. Hogg. Multiagent Control of Self-reconfigurable Robots. In Proc. of the *Intl. Conf. on Multiagent Systems*. IEEE, 2000.

[4] Z. Butler, R. Fitch, D. Rus, and Y. Wang. Distributed Goal Recognition Algorithms for Modular Robots. In Proc. of the *Intl. Conf. on Robotics & Automation*. IEEE, 2002.

[5] A. Castano, and P. Will. Mechanical Design of a Module for Reconfigurable Robots. In Proc. of the *Intl. Conf. on Intelligent Robotics & Systems*. IEEE, 2000.

[6] L.N. de Castro, and J. Timmis. Artificial Immune Systems: A New Computational Intelligence Approach. Springer-Verlag, New York, 2002.

[7] D. Dasgupta. Artificial Immune Systems and Their Applications. Springer-Verlag, Germany, 1999.

[8] A. Ko et al. Topological Representation and Analysis Method for Multi-port and Multi-orientation Docking Modular Robots. To appear In Proc. of the *Intl. Conf. on Robotics & Automation*. IEEE 2004.

[9] H. Y.K. Lau, and V.W.K. Wong. Immunologic Control Framework for Automated Material Handling. In Proc. of the *Intl. Conf. on Artificial Immune Systems*. 2003.

[10] J. H. L. Playfair, B. M. Chain. Immunology at a Glance. Blackwell Science, Bodmin, Cornwall, 2001.

[11] T.S. Ray. Overview of Tierra at ATR, in "Technical Information, no. 15, Technologies for Software Evolutionary Systems", ATR-HIP. 2001. <http://www.isd.atr.co.jp/~ray/pubs/overview/Overview.doc>

[12] L. A. Segel, and I. R. Cohen. Design Principles for the Immune System and Other Distributed Autonomous Systems. Oxford University Press, Oxford, 2001.

[13] J. Sharon. Basic Immunology. Williams & Wilkins, Pennsylvania, USA, 1998.

[14] M. Sipper. Machine Nature: the Coming Age of Bio-inspired Computing. McGraw-Hill, New York, 2002.

[15] A.O. Tarakanov, V.A. Skormin, and S.P. Sokolova. Immunocomputing: Principles and Applications. Springer-Verlag, New York, 2003.

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

[17] V. Vinge. A Fire Upon the Deep. Tor, New York, 1992.

[18] M.J. Wooldridge. An Introduction to multiagent systems. John Wiley & Sons, West Sussex, England, 2002.

[19] M. Yim, D.G. Duff, and K.D. Roufas. PolyBot: a Modular Reconfigurable Robot. In Proc. of the *Intl. Conf. on Robotics & Automation*. IEEE, 2000.