Proceedings of the 2004 IEEE
Conference on Robotics, Automation and Mechatronics
Singapore, 1-3 December, 2004

Immunology-based Control Framework for Multi-jointed Redundant Manipulators

H. Y. K. Lau
Department of Industrial and Manufacturing Systems
Engineering, The University of Hong Kong,
Pokfulam Road, Hong Kong, PRC.
hyklau@hku.hk

A. K. S. Ng
Department of Industrial and Manufacturing Systems
Engineering, The University of Hong Kong,
Pokfulam Road, Hong Kong, PRC.
kamseng_alex@graduate.hku.hk

Abstract — **Artificial Immune System (AIS) has recently been actively researched with a number of emerging engineering applications that has capitalized from its characteristics including self-organization, distributive control, knowledge mapping and fault tolerance. This paper reports the development of an AIS paradigm for the distributive control of a multi-jointed, redundant manipulator. Traditionally, manipulator control is achieved by analytical solutions. By adopting a multiagent-based control paradigm, a multi-jointed manipulator can be thought of as a group of separately controlled agents. In this paper, we investigate the viability of a multiagent immunology-based control framework for the trajectory control of a multi-jointed redundant manipulator.**

Keywords—artificial immune system, distributed control, redundant robot control

I. INTRODUCTION

Artificial Immune System (AIS) has recently been actively researched with a number of emerging engineering applications including the control of mechantronics systems, in particular, robots, which has capitalized from the special characteristics of the immune system including self-organization, distributive control, knowledge mapping and fault tolerance. Some examples of applying AIS in robotics include: Michelan and Von Zuben [1] investigated an autonomous control system of a mobile robot based on the immune network theory; Lau and Wong [2] developed a control framework to improve efficiency and robustness of a distributed material handling system; Meshref and VanLandingham [3] proposed the use of the immune network properties to model the robot joint behaviours, and Ko et al. [4] developed a distributed motion control framework for modular self-reconfigurable robots.

This paper introduces an AIS-based, multiagent, distributive robot control paradigm for the real-time control of redundant manipulators. In this paper, Section 2 outlines the classical approach to robot control and Section 3 describes the multiagent-based paradigm. An immunology-based control framework is presented in Section 4 and Section 5 describes the configurations of the redundant 4 degree-of-freedom WAM manipulator for the performance of experimental studies. Section 6 presents the experimental results and Section 7 conclude and discusses our findings.

II. MANIPULATOR CONTROL

Traditionally, a typical manipulator control system derives its trajectory based on the command given by the trajectory generator through well-defined analytical solutions. Inverse kinematics and Jacobian transpose are common approaches to derive such solutions [8, 9]. The analytic solutions enable computation of essential joint variables for a trajectory including position, velocity and orientation, etc. of a manipulator. In particular, Jacobian Transpose maps the Cartesian space velocity to joint space velocity that can be used to derive the control parameters such as the location of new set points of a manipulator:

$\theta_t = \theta_{t-1} + \Delta\theta$ where $\Delta\theta = J^{-1}\Delta x$ and J^{-1} is the Jacobian Inverse

Traditionally, a manipulator is controlled by inverse kinematics resolution [8, 10, 11]. In the case of the cable-driven teleoperator systems developed in the Intelligence Systems Laboratory at the University of Hong Kong, Lau et al. [12] propose a Jacobian-based control strategy for solving the inverse kinematic problem for the redundant WAM robot. However, there are limitations with kinematics solutions where singularities and redundancy are often major difficulties and concerns. Mori et al. [13] experimented with a multiagent-based manipulator control for 2D tasks by task decompositions. Ramdane-Cherif et al. [14] proposed a multiagent paradigm to solve for the kinematics solution. However, few studies have been done on applying multiagent-based paradigms to the practical control of redundant robots.

III. A MULTIAGENT-BASED CONTROL PARADIGM

Individual robot joints can be thought as spatially distributed agents that are controlled independently and concurrently. This suggests a totally distributed agent control paradigm for manipulator control. An agent [15-17] can be considered as a logical and autonomous entity which tries to contribute to the overall task achievement based on its own strategy. The agent is assumed to be able to recognize the conditions of the progressing task. Each agent tries to drive the assigned actuators with its own strategy by which it can improve the degree of task achievement. The overall system can be deemed as a set of distributed control variables which are directly or indirectly managed by these agents. Each joint controlling agent decides and carries out their own behaviors in response to external sensory signals. In this context, Lau and Ng [18] proposed a multiagent-based control paradigm for the slave arm control in a teleoperator system.

The proposed multiagent-based control paradigm, as shown in Figure 1, is a logically distributed system for manipulator control. Under this control approach, the execution of a trajectory involves the interaction between agents; and a conventional manipulator can therefore be thought of as a system composed of a group of agents, each of which is independently controllable. Conventional methods for robot control have largely been dependent on an exact kinematics solution. In contrast, a multiagent based paradigm has multiple agents and each agent may have individual goal(s). All cooperating agents may contribute to achieve a particular goal such as to move the end-effector of a multi-jointed robot along a particular trajectory. In the proposed control paradigm, cooperative strategies are adopted from the artificial immune system that each joint agent's behavior is suppressed or stimulated by the neighboring agents in order to achieve optimal control.

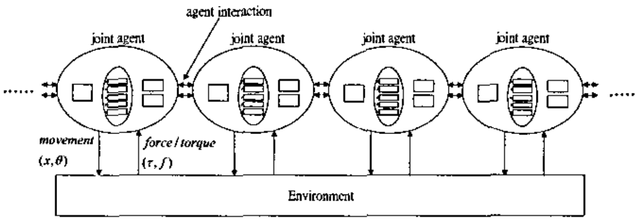


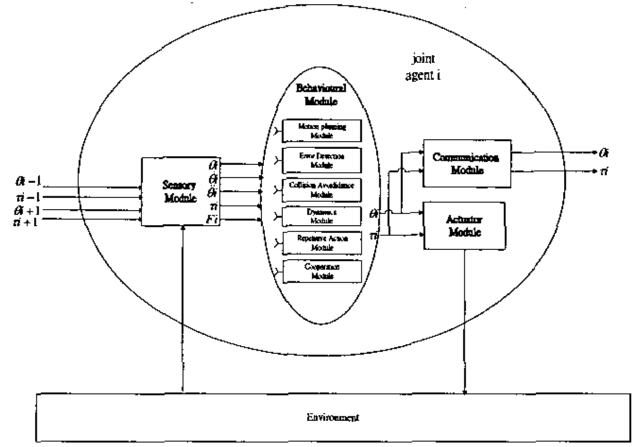
Figure 1. A multiagent-based control paradigm for the modeling and control of a multi-jointed manipulator

IV. AN IMMUNOLOGY-BASED CONTROL FRAMEWORK

The biological immune system consists of the tissues, cells and molecules that involve in adaptive immunity and totality of host defense mechanisms. The immune system is a natural, rapid and effective defense mechanism for a given host against various infections. An Immune action can often been classified into innate and adaptive immunity [19, 20].

Innate immunity refers to the abilities to recognize all pathogens specifically, where as the enhanced protection against re-infection are the unique features of adaptive immunity that is based on the clonal selection of lymphocytes bearing, antigen-specific receptors. Each lymphocyte carries cell-surface receptors of a single specificity each bearing a distinct receptor so that the total repertoire of receptors can recognize virtually any antigen. Adaptive immunity is initiated when an innate immune response fails to eliminate a new infection and antigens and activated antigen-presenting cells are delivered to the draining lymphoid tissues. Each lymphocyte bears a single type of receptor with a unique specificity. Interaction between a foreign molecules and a lymphocyte receptor capable of bringing that a molecule with high affinity leads to lymphocyte activation. The differentiated effector cells derived from an activated lymphocyte will bear receptors of identical specificity to those of the parental cell from which that lymphocyte was derived [19, 20]. The characteristics of distributed control, distributed memory, specificity, stimulation and suppression of human immune system inspired mathematical analogies for solving engineering problems, in particular, robot control problems described in this paper.

An intelligent joint agent in the proposed immunology-based control framework for multi-jointed manipulator control, as shown in Figure 2, is a distributed control, distributed memory and specific behavioral control agent for each joint of a robot. Each joint agent models a self-contained entity requiring minimal information from others. The framework consists of a number of basic modules including sensory module, behavioral module, communication module and an actuator. The sensory module and the actuator interact with the environment. The sensory module can be considered as a group of sensors such as encoders which are used to obtain the joint position or the joint torque sensor for deriving the joint torque via the motor current. The sensory module includes filters for filtering the noises and also includes the abstraction model for abstracting the actual sensed data to some sensory states. The actuator is used to output commands to the joint motor. The communication module is used for the communication between neighboring agents.



θ	Joint angle
$\dot{\theta}$	Joint velocity
$\ddot{\theta}$	Joint acceleration
τ	Joint torque
F	Force

Figure 2. Principal components of a joint agent in the immunology-based control framework

In this control framework, communication between neighboring joint agents are considered, where information are not gained or communicated beyond its two nearest neighbors. The behavioral module obtains sensory information from the sensory module and output to the communication module for stimulation and suppression of its neighboring agents for the actuation of the individual robot joint. The behavioral module is derived according to the artificial immune system. Each generic behavior is specialized for different categories of tasks where the interaction between the behavior and the relevant sensory state that initiate such behaviors imitate the mechanism of clonal selection. The behavior set refers to the capabilities of joint agent in association with robot manipulation tasks that imitate the antigen receptor repertoire storing the pattern for recognized antigens. Repetitive actions are learnt and stored in the behavior module which can be retrieved for tackling similar tasks that may come up subsequently. The mapping between the immune system and the AIS-based multi-jointed robot control framework is given in Table I.

TABLE I. MAPPINGS BETWEEN THE BIOLOGICAL IMMUNE SYSTEM AND THE AIS-BASED MULTI-JOINTED ROBOT CONTROL FRAMEWORK

Immune System	Artificial Immune System
Antigen	Sensory states
Antigen recognition	Behavior initiate condition
Behavior set	Antigen receptor repertoire
Behavior	Antibody
Repetitive actions	Immuno-memory

In the proposed intelligent joint agent, six modules are defined for the behavioral module, namely, the Motion Planning Module, Error Detection Module, Obstacle Avoidance module, Dynamic Model Module, Repetitive Action Module and Cooperation Module.

A. *Motion Planning Module*

The Motion Planning Module is designed for the trajectory generation for each joint agent. There are 2 trajectory generator adopted by the control framework, namely, the B-spline, (Equation 2) and straight-line interpolation (Equation 3). The inputs to the module are the current locations of the end-effector, $\vec{x}$, the target location, $\vec{s}$ and the distance $|\vec{s}\text{-}\vec{x}|$, the module then computes the trajectory in Cartesian coordinate using Cartesian trajectory interpolation. The Motion Planning Module produces output for the Error Detection Module.

Given a target S_i:

$$\mathrm{S_i} = [\mathrm{x} \quad \mathrm{y} \quad \mathrm{z}]^{\mathrm{T}} \text{ with reference to local frame}\{\mathrm{i}\} \ s.t.\ \mathrm{S_i} = \mathrm{S_w} T_1^0 \ldots\ldots T_i^{i-1}$$

where (1)

$\vec{\mathrm{S}}_{\mathrm{w}}$	The set point in the world frame
T_i^{i-1}	The i^{th} Denavit-Hartenberg transformation matrix

B-spline Interpolation:

$$y = \sum_{i=0}^{3} P_i \square N_{i,m} \tag{2}$$

where

$$N_{0,4} = (1-t)^3$$
$$N_{1,4} = 3t(1-t)^2$$
$$N_{2,4} = 3t^2(1-t)$$
$$N_{3,4} = t^3$$

Straight Line Interpolation:

$$\vec{y} = nk(\vec{s} - \vec{x}) + \vec{x} \tag{3}$$

where

n	represents the coefficient
k	represents the cycle
$\vec{s}$	represents the set point
$\vec{x}$	represents the current position

B. *Error Detection Module*

The Error Detection Module is designed to derive the joint output for each joint agent with a view to minimize the error between the set point and the current position. The module controls the movement of the single joint variable (θ_i) that moves within the joint limit (θ_{si}).

The module has knowledge about the link dimension of the joint and the corresponding joint range. It can be informed of the locations of the end-effector ($\vec{x}_i$) and target ($\vec{\mathrm{S}}_i$) and the distance $|\vec{S}_i\text{-}\vec{\mathrm{x}}_i|$. It also controls the joint angle according to its objective function (Equation 4) and it stimulates the activities of the Motion Planning Module such that the Motion Planning Module then generates the next control set point to achieve the desired movement.

The objective function:

$$\theta_i(\mathrm{t}) = \vec{\mathrm{k}}_i \left\lfloor \vec{S}_i(\mathrm{t}) - \vec{\mathrm{x}}_i(\mathrm{t-1}) \right\rfloor \tag{4}$$

where

$$\vec{\mathrm{x}}_i(\mathrm{t}) = \mathrm{fwdkinematics}(\theta_1, \theta_2, \theta_3, \theta_4)$$

$$\vec{\mathrm{k}}_i = [\mathrm{k_{ix}} \quad \mathrm{k_{iy}} \quad \mathrm{k_{iz}}]$$

s.t. $\vec{\mathrm{k}}_i$ is the row vector and is a constant for agent_i

C. *Obstacle Avoidance Module*

This module provides the capability to maintain a safety distance, $\mathrm{D_s}$, from an obstacle, $\vec{\mathrm{o}}$, in order to prevent collision. This module has the capacity to suppress other behaviors when it is initiated.

$$|\vec{\mathrm{x}} - \vec{\mathrm{o}}| < D_s \tag{5}$$

$$\theta_i = k_o(\vec{\mathrm{x}} - \vec{\mathrm{o}})$$

where

$\mathrm{k_o}$	is the obstacle avoidance coefficient
$\vec{\mathrm{o}}$	is the obstacle position

D. *Dynamic Model Module*

The module provides the capability for high precision and high speed control of the robot joint. The joint position, velocity and acceleration are required to compute the corresponding joint torque of the robot joint according to the equation of motion (Equation 6).

$$\tau = M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) + G(\theta) \tag{6}$$

where

$M(\theta)\ddot{\theta}$	is the dynamic force

$V(\theta,\dot{\theta})$ is the Centrifugal and Coirolis forces

$G(\theta)$ is the gravitational force

E. Cooperation Module

This module is activated when the end-effector is close to the target such that the distance between the target and the current position is smaller than a predefined distance. The module generates suppression signal to deactivate the activities of its neighboring agents. If the end-effector is far away from the target such that the distance between the target and the current position exceeds a predefined distance, D_c, the cooperation module then stimulates the activities of neighboring agents to achieve optimal control.

The definitions for the distances parameters are given as follows:

$$|\bar{s}-\bar{x}| \le D_c \quad (7)$$

$$n_{i-1}=n_c$$

$$n_{i+1}=n_c$$

where

n_{i-1} is the coefficient of Error Detection Module in joint $_{i-1}$

n_{i+1} is the coefficient of Error Detection Module in joint $_{i+1}$

n_c indicates the suppression signal

$$|\bar{s}-\bar{x}| > D_c$$

$$n_{i-1}=n_s$$

$$n_{i+1}=n_s$$

where

n_s is the stimulation signal

V. SYSTEM CONFIGURATION

To realize the practicability of the proposed AIS-based control framework, a robot system that is controlled by real-time hardware and software is deployed. Our system installed at the Intelligent Systems Laboratory of the University of Hong Kong is a four degrees-of-freedom, cable-driven redundant WAM robot that is capable of performing whole arm manipulation (Figure 3). WAM is an extremely light weight, fully cable-driven and low friction manipulator capable of very high speed motion [21]. In addition to joint torque sensing for force control applications, an ATI force torque sensor is installed at the end effector of the manipulator to derive the force feedback signal. The design of the WAM control system is based on the QNX real-time multi-tasking operating system. A number of prioritized control tasks including joint servo and trajectory generator and these tasks are communicating in a synchronized metaphor. Joint servos are designed to compute the exact torque required to actuate the joint motors. In the distributed control, each joint is controlled by its joint agent that incorporates its trajectory generator, servo and sensors.

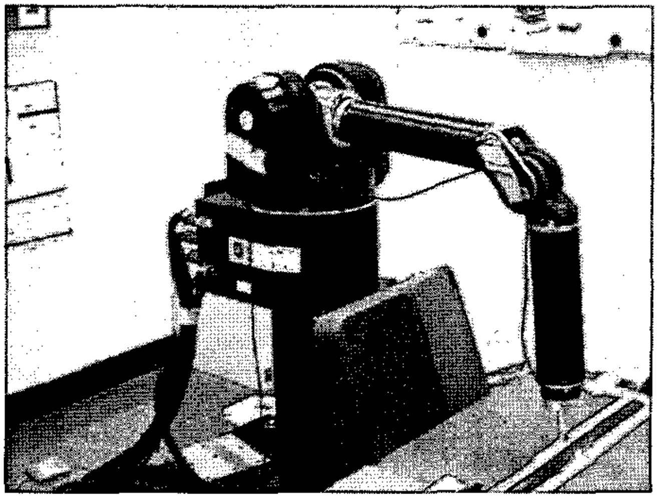


Figure 3. The Whole Arm Manipulator (WAM)

VI. EXPERIMENTATION

Simulation study is performed using a simulator developed using MATLAB for the WAM system. The manipulator trajectory plots are generated by the modified Corke's Robotic Toolbox [22] for the WAM robot. In the simulation study, the WAM is initialized with the joint configuration of $(\theta_1, \theta_2, \theta_3, \theta_4) = (0°, 90°, 0°, 90°)$. Under this starting configuration, the corresponding end-effector locations are at $\bar{x}_i=\{x, y, z\}$. In a particular trajectory movement, the target locations of the end-effector is set to $\bar{x}_2=\{70, 10, -43\}$ cm. The target position of the WAM is (x: 70cm, y: 10cm,z: -43cm). 4 agents are deployed to control the 4 Degree-of-freedom redundant WAM. Each joint agent controls a single joint. Figures 4 - 7 show the joint-based trajectory of the manipulator when executing this trajectory between the 2 set points. Figure 8 shows the trajectory of manipulator from the simulation.

The computation complexity of motion control using analytical solution and that using AIS-based control paradigm can be compared based on the number of trigonometric computation required. The number of trigonometric computation for the motion control using the AIS-based control paradigm is of the order of 10^2 whereas the number of trigonometric computation for a typical analytical solution is in the order of 10^8. Further investigation of individual joint-based trajectory shows that the AIS-based control paradigm results in a highly efficient means for computing the intermediate joint set points for executing a desired trajectory. The proposed AIS-based control paradigm shows the ability to deal with singularities and joint limit avoidance as well as implicit redundancies control while the motion control using analytical solution requires explicit strategies to resolve for singularities and to prevent solutions being trapped in sub-optimal situations.

VII. CONCLUSION

This paper presents an AIS-based manipulator control paradigm. The immunology-based control framework for a redundant multi-jointed manipulator is described with the essential algorithms deployed outlined in the paper.

Simulation studies were carried out to establish the practicability of the multiagent immunology-based control framework and it is shown that this approach reduces the computational complexity that does not require computing the pseudo-inverse Jacobian matrix as in the core of an analytical approach. The simulation study shows that desirable joint trajectories were produced to control the redundant WAM and results show that the multiagent immunology-based control framework can control the redundancy without resorting to solving high order polynomial functions. In addition, computational efficiency is improved, which is shown in the results given in the previous section. These findings indicate that the proposed multiagent immunology-based control provides a promising approach to the control of redundancy. Joint dynamic model and obstacle avoidance strategies have been investigating and will be included in the future implementations. Currently, further studies to investigate the coordination and cooperation between the joint agents under the AIS framework are being undertaken.

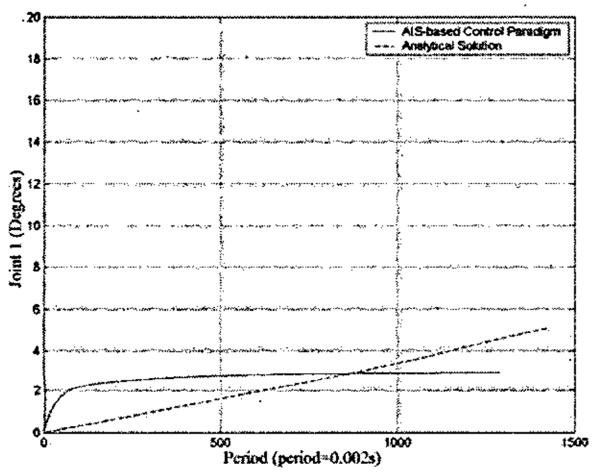


Figure 4. Joint 1 trajectory

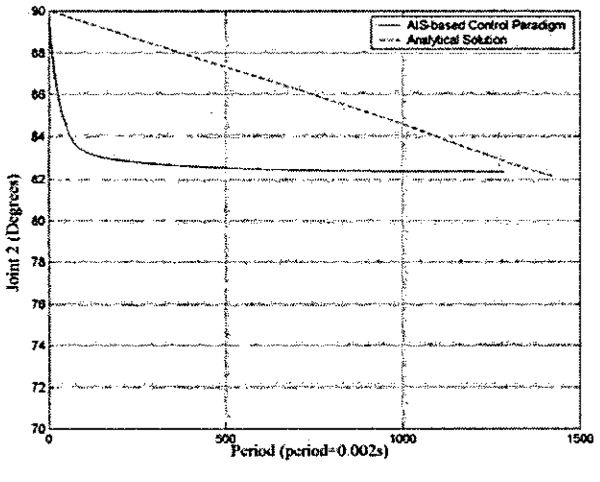


Figure 5. Joint 2 trajectory

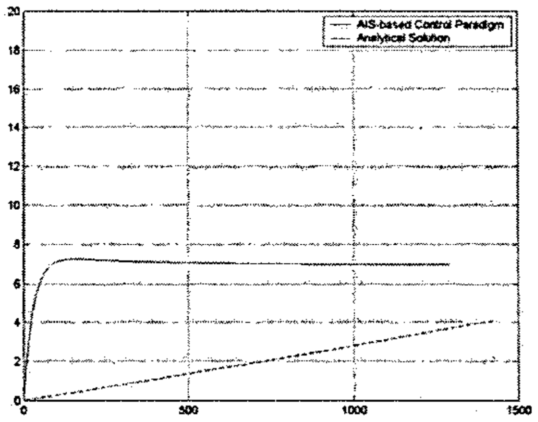


Figure 6. Joint 3 trajectory

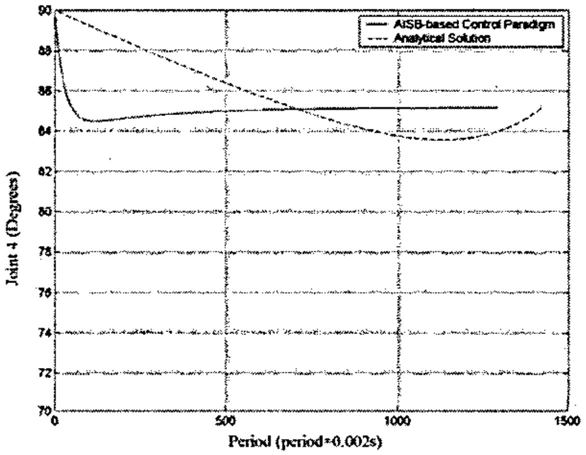


Figure 7 Joint 4 trajectory

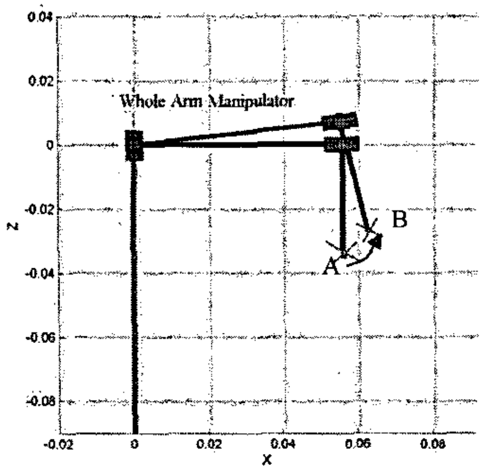


Figure 8. Simulated trajectory of WAM as it transverse from location A to location B

ACKNOWLEDGMENT

This research is partly supported by the Research Grant Council of the Hong Kong Special Administrative Region, PRC under the CERG Project No. HKU7079/02E.

REFERENCES

[1] Michelan, R. and Von Zuben, F. J. (2002) Decentralized Control System for Autonomous Navigation based on an Evolved Artificial Immune Network. *In the proceedings of the special sessions on artificial immune systems in Congress on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Honolulu, Hawaii, May, pp. 1021-1026

[2] Lau, H. Y. K. and Wong, V. W. K. (2003) Immunologic Control Framework for Automated Material Handling. *Proceeding of Second International Conference on Artificial Immune Systems (ICARIS)*, September 1-3, Napier University, Edinburgh, UK, pp. 57-68

[3] Meshref, H. VanLandingham, H. (2001) Immune network simulation of reactive control of a robot arm manipulator. *In the Proceedings of the 2001 IEEE Mountain Workshop on Soft Computing in Industrial Applications*, SMCia/01, pp. 81-85. June 25-27.

[4] Ko, A. Lau, H. Y. K. and Lau, T. L. (2004)An artificial immune system based distributed motion control framework. *In the Proceedings of The 2004 IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, Louisiana, USA, April 26 - May 1.

[5] Bolmsjo, G., Olsson, M. and Cederberg, P. (2002) Robotic arc welding – trends and developments for higher autonomy *Industrial Robot*, Vol.29, No.2, pp. 98-104

[6] Rooks, B. (1997) Robot welding in shipbuilding. *Industrial Robot*, Vol.24, No.6, pp. 413-417

[7] Yagi, T. (2004) State-of-the-art welding and de-burring robots. *Industrial Robot*, Vol.29, No.6, pp. 495-499

[8] Craig, J. J. (1986) *Introduction to robotics- Mechanics & Control*. Addison-Wesley.

[9] Paul, R. P. (1981) *Robot manipulators: mathematics, programming, and control*. The MIT Press.

[10] Asada, H. and Slotine, E. J. (1986) *Robot Analysis And Control*, 1 ed. Wiley-Interscience.

[11] Murray, R. M., Li, Z. and Sastry, S. S. (1994) *A mathematical introduction to robotic manipulation*, CRC Press.

[12] Lau, H. Y. K., Lau, T. L. and Wai, L. C. C. (2002) A Jacobian-based redundant control strategy. *Proceedings of the Seventh International Conference on Control, Automation, Robotics and Vision, ICARCV 2002*, Singapore December, pp. 1060-1965.

[13] Mori, A., Naya, F., Osato, N. and Kawaoka, T. (1996) Multiagent-Based Distributed Manipulator Control. *Proceedings of the 1996 IEEE/SICE/RSJ International conference on Multisensor Fusion and Intergration for Intelligent Systems*, pp. 289-296

[14] Ramdane-Cherif, A., Levy, N., Djenidi, H. and Tadj, C. (2002) Agents Paradigm to Solve a Complex Optimization Problem. *Proceedings of the First IEEE International Conference on Cognitive Informatics, ICCI' 02*, pp. 247-256

[15] Dastani, M., de Boer, F., Dignum, F., van der Hoek, W., Kroese, M. and Meyer, J.-J. Ch. (2002) Implementing Cognitive Agents in 3APL. *Proceedings of the The 14th Belgian-Dutch Conference on Artificial Intelligence, BNAIC2002*, Leuven, Belgium, pp. 515-516

[16] d'Inverno, M., Kinny, D., Luck, M. and Wooldridge, M. (1998) A Formal Specification of dMARS, *In Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages, Singh, Rao and Wooldridge (eds.), Lecture Notes in Artificial Intelligence*, 1365, pp. 155-176, Springer-Verlag.

[17] Jennings, N. R. and Wittig, T. (1992) ARCHON: Theory and Practice, *Distributed Artificial Intelligence: Theory and Praxis*, Kluwer Academic Press, pp.179-195.

[18] Lau, Y. K. H. and Ng, A. K. S. (2004) Teleoperation with a distributed multiagent-based slave manipulator control system, *Proc. 1st IFAC Symposium on Telemetric Applications in Automation and Robotics (TA2004)*, Helsinki, Finland.

[19] Janeway, C. A., Travers, P., Walport, M. and Shlomchik, M. (2001) *Immunobiology: the immune system in health and disease*, Garland Publishing.

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

[21] Townsend, W. T. and Guertin, J. A. (1999) Teleoperator slave – WAM design methodology. *Industrial Robot*, Vol.26, No.3, pp. 167-177

[22] Corke, P. I. (1996) A Robotics Toolbox for MATLAB, *IEEE Robotics and Automation Magazine*, Vol.3 Issue 1, March, pp. 24-32