The Robust Hybrid Self-tuning Controller using Biological Protection System

Jong-won Kim[1], Sang-hyung Ha[2], Jae-yong Seo[1], Hyun-chan Cho[1], Hong-Tae Jeon[3]
1. School of Information Technology, Korea University of Technology and Education
2. Seoul Technology Appraisal Center of Korea Technology Credit Guarantee Fund
3. School of Electrical & Electronics Eng, Chung Ang University
email : kamuiai@kut.ac.kr

***Abstract* - In This paper, we propose a methodology of efficient control for the non-linear dynamic systems by biological protection system. Existing neural network and genetic algorithm are for the control of nonlinear system state control. But it is not particularly good changeable state and must be re-learning for the control of the system in change state. In this case, system needs a lot of time until tuning. Therefore, we suggest intelligent hybrid self-tuning controller for this problem. That consists of neural network, genetic algorithm and immune system. Immune bank used computer memory can save dynamic error pattern and decrease re-learning time of neural network by searching dynamic error pattern use genetic algorithm. We verify the effect of methodology in two joint link robot-manipulator by computer simulations.**

I. INTRODUCTIONS

In field of adaptive modern control system, the development of robust-controller is being researched a identification parameter.

Specially, the controller that is used intelligent algorithm has a lot of advantages [1][8].

This paper will propose the methodology of increasing robustness in non-linear dynamics system. That is depended on change value of parameter in non-linear dynamic system using neural network.

Adaptive controller using neural network have disadvantage witch is re-learning time to adapt system. To improve this problem, reconstruction of neural network, training algorithm and usage of reference signal on neural network by PID controller are studied [1].

But we solved this problem by biological protection system using immune-system-model and Genetic algorithm technique.

We assumed change value parameter to antigen and against identification parameter to antibody.

We apply the immune system in non-linear dynamic system that using genetic algorithm for searching antibody in immune bank.

This studies consists of three parts. First is non-linear dynamic system design, second is adaptive controller design last one is immune system and genetic algorithm.

We try to prove the methodology through computer simulation.

II. METHODOLOGY

A. Procedure

Neural network is useful with adaptive controller for non-linear dynamic system.

Immune system is for change value of parameter. It saved antibody for antigen and used genetic algorithm for searching antibody in immune bank.

Genetic algorithm is search antibody fast. Therefore, used searching antibody, we can decreasing relearning time of neural network by antibody.

Process of Methodology is following.

a. Define the non-linear system and neural network
b. Create antibody set used known antigen error pattern
c. Save weight set with antigen error pattern
d. Neural Input relearning by new antigen
e. Search antibody used genetic algorithm
f. Change initial weight set of neural network
g. Relearning neural network
h. Non-linear dynamic system control by output

B. Non-linear dynamic system

The non-linear dynamic system which is used in this paper is tow-link manipulator. It moves along desired path (trajectory) in X,Z planes that applied by gravity. The trajectory refers to time history of position, velocity, and acceleration [2].

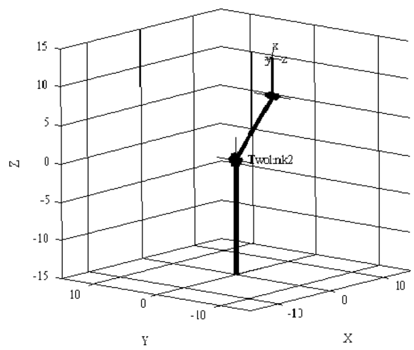

Fig.1. Two-link manipulator

We assume the manipulator is rigid body and link length L1 is 50cm and L2 is 25cm. Payload M is on the end-effector.

The end-effector of manipulator moves with velocity and acceleration from initial position to goal position as Fig2.

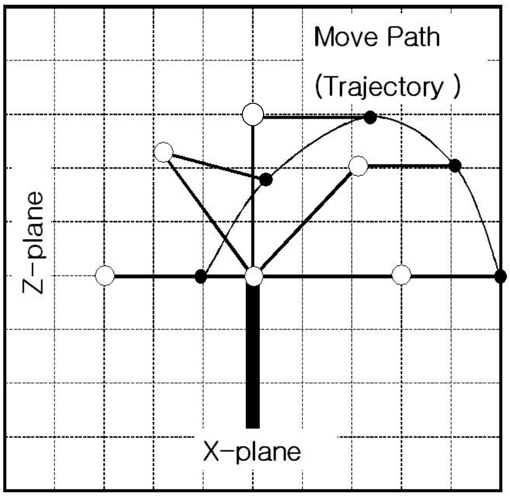

Fig.2. Trajectory and payload

The position of end-effector moves along trajectory.

Joint (actuator) torque calculated depended on payload M[2].

Trajectory of manipulator can be represented to joint angle (Q).

Payload M which is influence trajectory used antigen in immune system.

C. Adaptive Neural Network Controller

1. Structure

Adaptive controller using neural network have been developed in variety ways.

We use BPN(Backpropagation Network) as Fig3 [3][4].

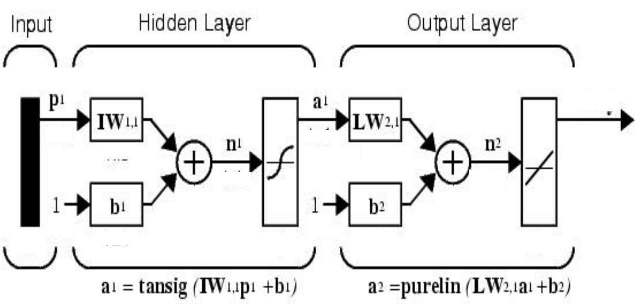

Fig. 3. Backpropagation Neural network

Some parameters which consist of neural network as follows. That is consist of two layers and six input element(p).

Input p is Q(θ1,θ2), dQ(dθ1, dθ2), ddQ(ddθ1, ddθ2).

Output y is torque(tau1, tau2) all neurons full connected.

Q is joint angle, radian.

dQ is velocity.

ddQ is acceleration.

Activation function of first layer is hyperbolic tangent sigmoid and second is linear transfer function.

Learning methodology is gradient descent weight/bias learning algorithm.

Learning-complete-condition is that performance goal satisfies less than 0.001 error range and max iteration is 500 steps [4][5].

D. Immune system

1. Methodology of immune bank generation

Dynamic system occur trajectory error while the manipulator moves along the desired trajectory. That is saved movement angle, velocity and acceleration by input antigen.

Input p uses parameter to make antibody pattern consist binary. And saves pattern. It's calling Q-error pattern.

a. Output Q compares with desired Q each element.
b. If desired Q < output Q than error pattern value is 0.
c. If desired Q >= output Q than error pattern value is 1.
d. Make error code(pattern) consist of 1 or 0.

Desired Q set	0	0.1	0.2	1	1.12	…
Real output Q	-0.1	0.12	0.31	0.8	0.67	…
Q Error pattern	0	1	1	0	0	…

Fig. 4. Sample of Q error pattern

We make Q pattern by same method. And make Q error pattern set.

Q1 Error pattern	0	1	1	0	0	…
Q2 Error pattern	1	0	1	0	0	…
…	…	…	…	…	…	…
Qn Error pattern	0	1	0	1	0	…

Fig. 5. Sample of Q error pattern set

Also, use Q to learn neural network and save learning weight-set.

We makes immune bank matched error pattern with weight-set.

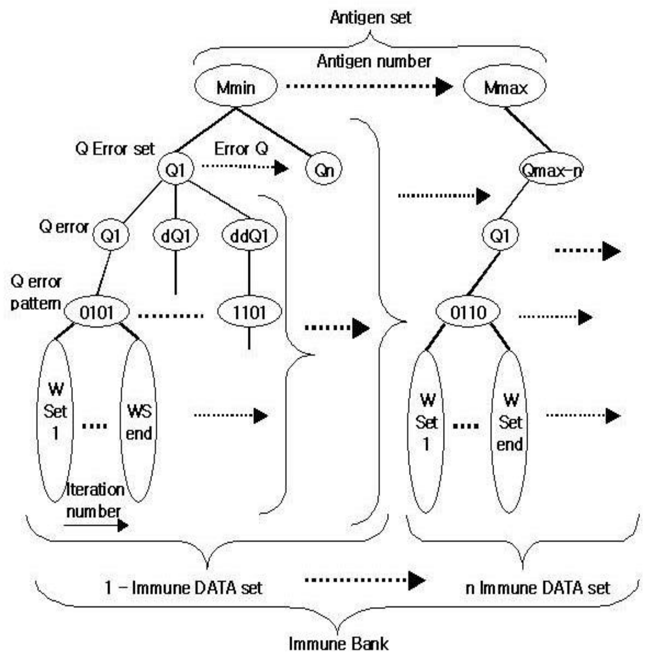

Fig. 6. Immune Bank structure

2. Genetic Algorithm

Known antigen is continuously learned 1 to n, neural network finished learning controller input unknown antigen Mx, and save output Q set.

Saved Q set used bias to make antigen Q-error pattern organized 1 and 0 [6].

We make population as follows.

a. Counting 1 and 0 from Q-error pattern.
b. Accumulate number of 1 and 0.
c. Save population pattern.
d. Search same pattern from other Q-error pattern set in immune bank

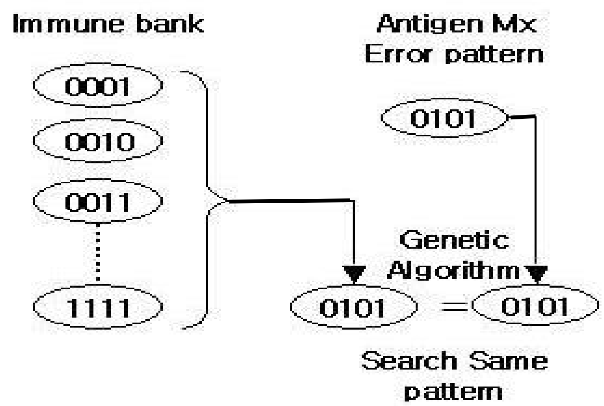

Fig. 7. Search of antibody pattern in immune bank

Replace Neural network initial weight set by searched antibody weight set.

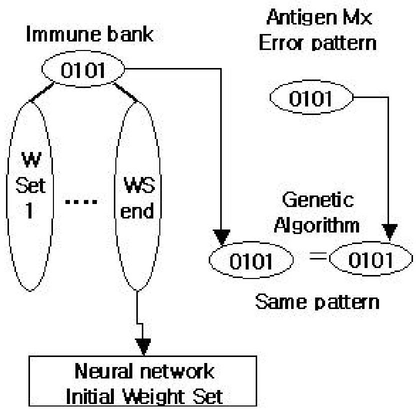

Fig. 8. Select weight set in immune bank

After replace, we check the neural network learning iteration and time.

III. COMPUTER SIMULATION TEST

A. Computer simulation

1. Two-link manipulator

As Fig 9, we designed the computer simulation model. The system is consist of 4 block

.

a. Desired Q set generator (Trajectory Generator)
b. Inverse-Dynamics (Calculate tau)
c. Dynamic system (Two-link manipulator has error)
d. Output of dynamic system (Real Q set)

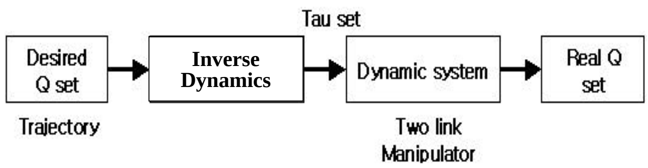

Fig. 9. Dynamic system structure

Simulation time is 10 second.

Data save sampling time is 0.1 second.

Calculated Q error by desired Q set compared with real Q set [5][9][10].

2. The learning of neural network by known antigen set

We input known antigen set to neural network regularly.

Known antigen set range is 1 to 2kg. (Mmin is 1 kg and Mmax is 2kg)

The initial weight set value of this neural network is all one [7].

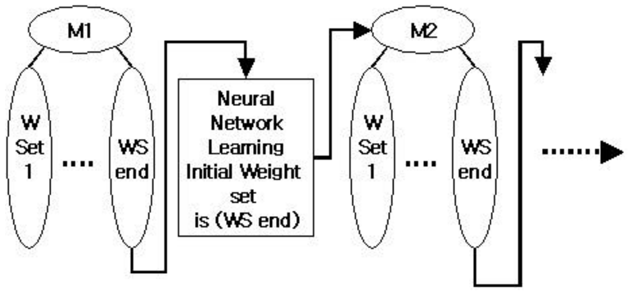

Fig. 10. Learning neural network (known antigen set)

3. Create new antigen pattern

We input new antigen(Mx) to dynamic system.

Create antigen Q error pattern by output Q set of dynamic system.

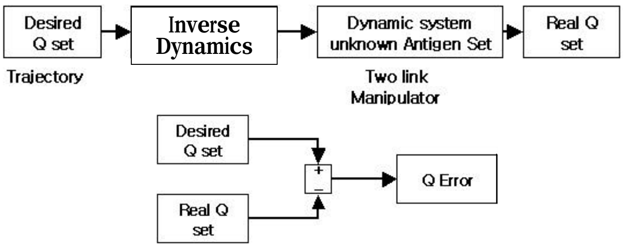

Fig. 11. Create antigen Q error set

3. Search the antibody from immune bank

We used genetic algorithm for search antibody against weight set and select end-weight set from weight set.

Neural network relearn with end-weight set.

If a antibody pattern doesn't exist in immune bank, it select a most similar pattern with a weight set [8].

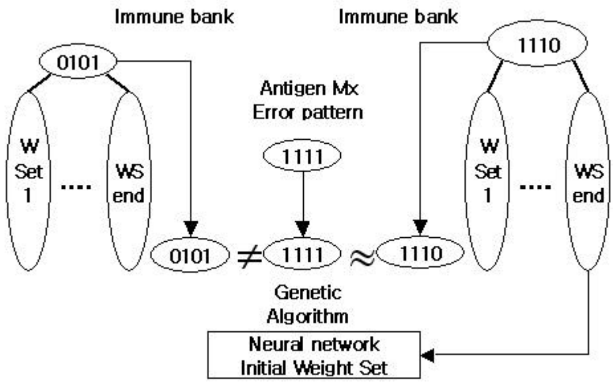

Fig. 12. Most similar pattern selection

Hybrid Self-turning Neural network Controller as Fig14.

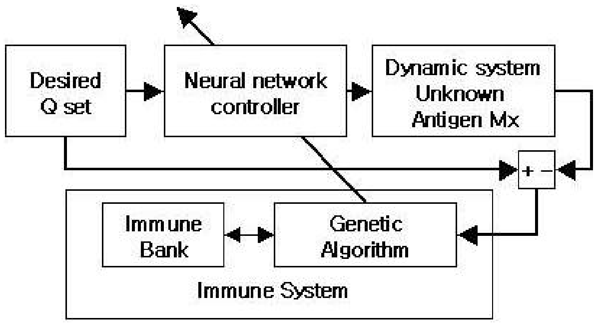

Fig. 13. Hybrid self-tuning controller

4. Computer Simulation Result

We simulate with three unknown new antigens. They are Mx1, Mx2 and Mx3.

a. New antigen Mx1 < Mmin (Mx1 = 0.4Kg)
b. New antigen Mmin < Mx2 < Mmax (Mx2 = 1.5kg)
c. New antigen Mmax < Mx3 (Mx3 = 2.5kg)

We checked re-learning iteration of neural network

1) Antigen Mx1

Antigen Mx1 pattern doesn't exist. Therefore, neural network initial weight set used set used weight-set of Mmin.

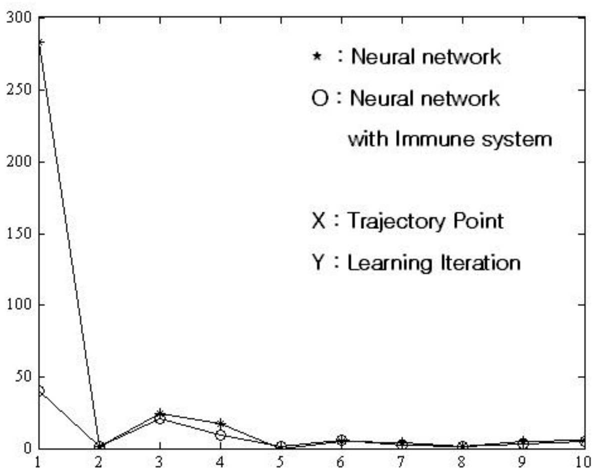

Fig. 14. Compare learning iteration

Learning iteration is similar. Consequently, decrease of iteration is meaningless for reduction of learning time.

2) Antigen Mx2

Antigen Mx2 pattern exist in immune bank. Therefore, neural network initial weight set use weight set in immune bank. Iteration is decreased.

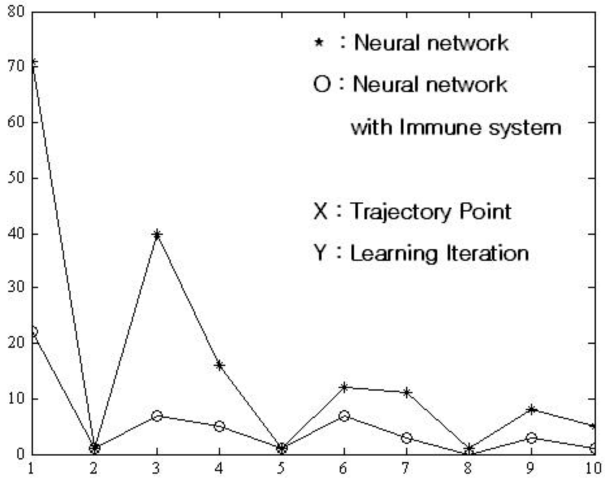

Fig. 15. Compare learning iteration

3) Antigen Mx3

Antigen Mx3 pattern doesn't exist. Therefore, neural network initial weight set use weight-set of Mmax.

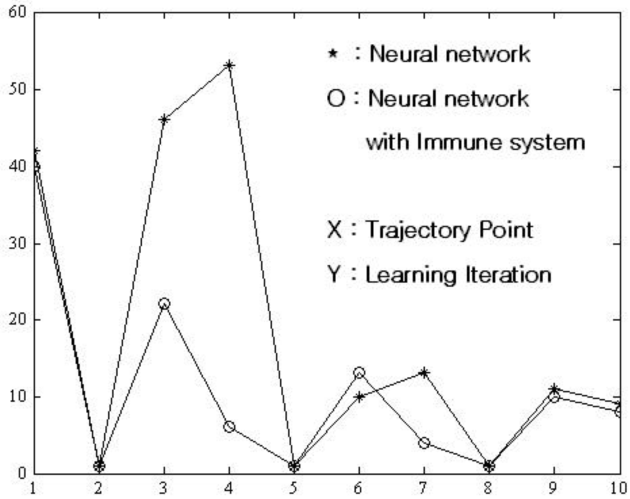

Fig. 16. Compare learning iteration

IV. CONCLUSION

This paper is about hybrid self-tuning controller design based on neural network and immune system. We try to solve time reduction problem using neural network controller with intelligent algorithm from a model, biological protection system.

By the simulation, the methodology has many advantages. But it has some important problems.

First, if pattern size(bit size) is short, error pattern numbers reduced immune bank size. Thus, similar patterns appear. Several patterns are similar also in this simulation.

Second, seek time of error pattern using genetic algorithm is not included in real control time.

This paper focus on simply iteration decline. However, it is robust against antigen changing.

The studies which refers to genetics technique for solution of the real-time calculation problem, and high-speed parallel processing must be formed for immune system.

REFERENCE

[1] Huailin Shu, Xiucai Guo, Hua Shu, "PID neural networks in multivariable systems," Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium on, pp.440-444.

[2] John J. Craig, "Introduction to Robotics mechanics and control," Addison-wesley.

[3] Philip D. Wasserman, "Neural Computing Theory and Practice," VNR, 1989.

[4] Skapura. D. M, "Building Neural Network," Addison & wesley.

[5] Hagan, Deumth, Beale, "Neural Network Design," PWS publishing, 1999.

[6] Moon, "Genetic Algorithm," Doo-yang-sa, 2003.

[7] John F. Gardner, "Simulation of Machines," Books / Cole, 2001.

[8] Yuval. Davidor, "Genetic Algorithm and Robotics," World scientific.

[9] N. K. Bose & P. Liang, "Neural Network Fundamentals with Graphs, Algorithms, and Applications," McGraw-hill, 1996.

[10] Mohand Mokhtari, Michel Marie, "Applications de MATLAB5 et Simulink2," springer, 1998.

ACKNOWLEDGMENT

This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce, Industry and Energy of the Korean Government.