Linear Quadratic Regulator Using Artificial Immunize System

Abdulla I. Abdulla

College of Electronics Engineering
University of Mosul
Mosul, Iraq
E-mail: Abd_63om@yahoo.com

Abstract

This work demonstrates the application of Artificial Immunized System (AIS) in designing on-line the parameters of the Linear Quadratic Regulator (LQR) for SISO plant. The AIS is used in creating the adaptive critics which represent the matrix parameters in the solution of the *Riccati* equation of the LQR. It is found that the AIS possess promising features to deal on-line with the plant parameter variations and reflect acceptable regulating and tracking characteristics.

Key words: Artificial Immunized System, Evolutionary Algorithms, Adaptive Systems, Linear Quadratic Regulator.

المنظم الخطي المتعامد بأستخدام نظام المناعة الاصطناعي

عبد الله أبراهيم عبد الله
كلية هندسة ألألكترونيات
جامعة الموصل

المستخلص

يوضح هذا العمل تطبيق نظام المناعة الاصطناعي (AIS) في التصميم الأني لمتغيرات المنظم الخطي المتعامد (LQR) لنظام ذو إدخال واحد وإخراج واحد (SISO). يستخدم نظام المناعة الاصطناعي (AIS) في تكوين النقاد التكيفي والذي يمثل عناصر المصفوفة الموجودة في حل معادلة *Riccati* للمنظم الخطي المتعامد . ووجد بأن نظام المناعة الاصطناعي يمتلك ملامح واعدة للتعامل الاني مع متغيرات النظام ويعكس خواص تنظيم و تتبع مقبولة.

الكلمات الدالة: نظام المناعة الاصطناعي ، الخوارزميات التطورية، النظام المتكيف، المنظم الخطي المتعامد.

Received: 18 – 10 - 2010 **Accepted: 31 – 7 - 2011**

List of Abbreviations

AIS	Artificial Immunized System
LQR	Linear Quadratic Regulator
ICS	Immunized Computational System
BCS	base-line computational system
CCS	changeable computation system
SISO	Single-input single output

1. Introduction

A practical way to define intelligence of a system is to approach it as having various levels of capabilities for self-improvement. Self- improvement is an important goal of human intelligence. It is quantifiable and measurable in various ways.

Immunology is the science of built-in defense mechanisms that are present in all living beings to protect against external attacks. A biological immune system can be thought of as a robust, adaptive system that is capable of dealing with an enormous variety of disturbances and uncertainties. Biological immune systems use a finite number of

discrete detectors "building blocks" to achieve this adaptiveness. These building blocks can be thought of as pieces of a puzzle, which must be put together in a specific way to neutralize, remove, or destroy each unique disturbance the system encounters.[1]. Researchers attempt to

define and learn these detector blocks for various engineering problems of interest.[2,3,4]

In table 1, an example is presented in terms of intelligence required in an intelligent controller, and compares this example to the levels of intelligence exhibited in an immune system. These levels are motivated by the investigation research in Immunized artificial system (IAS) and the quest to implement autonomous controllers.

Table (1) . Levels of Intelligence (L), Intelligent **Control** and Immune System Response.

Immune Response	L	Intelligent Control
Antigen is present in the system		Disturbance is present in the system
Innate Immunity is the first line of defense. Alerts next level	O	Robust feedback control (non-adaptive; Error tends to zero)
T cells activate the B cells response (adaptation) to counteract the antigen	1	An error critic modifiers the controller parameters to counteract the disturbances
Macrophages present the antigen to the T cells and T cell response (adaptation) is activated	2	Some utility function translates the disturbances into an error function and presents it to the critic. The critic starts the adaptation process
For certain antigen, the macrophages adapt with the end product; anti body that modifies the specificity bread but weak to narrow but strong of controller from.	3	For certain causes of disturbances (Ex: control hardware failure is adapted directly based on a planning function

The detector blocks can be processed using the features from the biological immune systems captured into computational model known as Immunize Computational Model.

2. The Immunized Computation System

Immunized Computational System (ICS) use the immune system metaphor along with computation (both hard and soft computing) techniques to attempt to produce robustness and adaptability of a biological immune system [5] .Figure (1) presents a general block diagram of immunized computational system. The proposed system has the following attributes:

The dashed lines in Figure (1) represent the base-line computational system (BCS). This system is designed to represent an average of the uncertain system. Since this design is carried out off-line, any standard technique can be used for its synthesis. The base-line system is analogues to static portion of the antibodies.

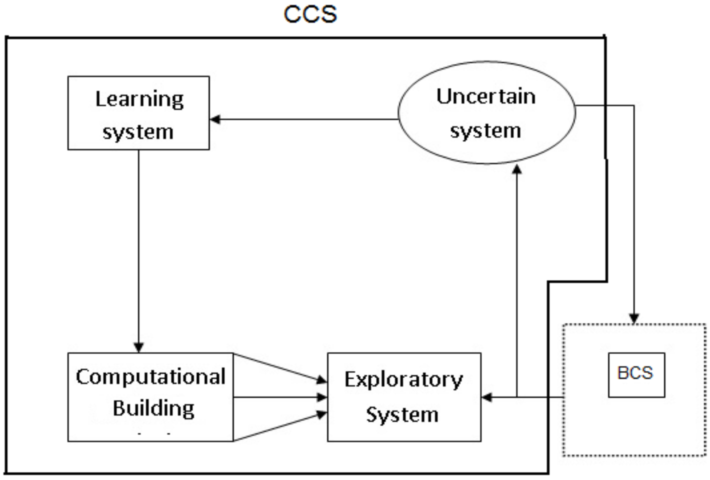


Fig. (1) Immunized Computational System Structure.

The solid lines in Figure (1) represent the changeable computation system (CCS). This represents the variable region of the antibody and epitope equivalents. This structure must be adopted on-line. To include the innate immunity equivalence, the changeable compositional systems that are known a *priori* can be stored in look –up tables and can be used to produce the right anti bodies and (CCS) models. Similar to the variable region of the anti bodies, the (CCS) provides diversity to the immunized computational system. The exploratory system is basically an evolutionary algorithm (EA) variant the uses crossover, selection, and mutation to arrive at a suitable (CCS). The learning system consists of a suit of learning paradigms to learn and store important computational building blocks that are not available a *priori*. A system that is either being emulated or being controlled.This system is constantly changing (time varying or non –linear plant).

2.1 Computational Building Blocks

Computational building blocks are defined as segments of a computational system topology (for example, a neural network connection, or family of connections, along with its associated weights) that contributes in establishing a good mapping for a class of input-output characteristics [6].

Building blocks can be of different order. The order of a building block specifies the number of specific connection defined in the building block. Example of building blocks using neural connection is shown in Figure (2). Building blocks are

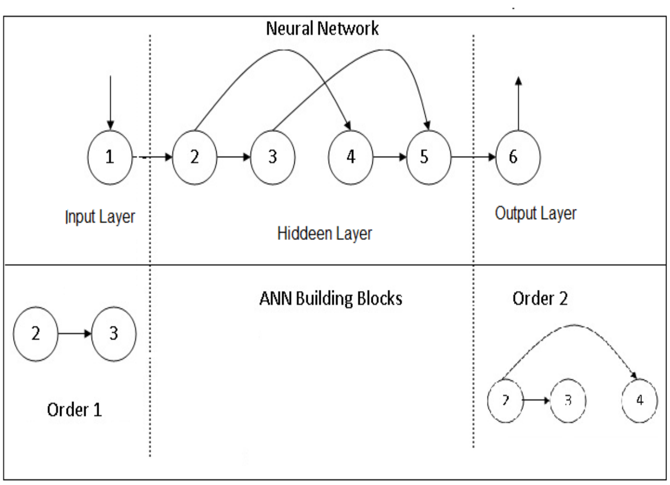


Fig. (2) Examples of Immunized Computational building blocks specified using universal representation scheme that uses a neuron as the basic processing element. Thus the order (1)- building block consist of two neurons and the relationship between them. In case of Artificial Neural Network (ANN), the relationship is the connection strength and the neurons are characterized by their type (input, hidden, output), the type of aggregation, and activation functions. [5]

2.2 The Role of Evolutionary Algorithms

Analogous to the DNA molecule, the computation building blocks need to be coded as string of building blocks. An example of a neural network with the corresponding genetic representation is show in Figure (3).

For on-line processing, a population of (CCS) is randomly constructed using juxta positioning of the building blocks, forming a population of messy string (see figure 4). Next, we find the best sting that will represent the (CCS) through crossover, selection, and hyper-mutation of these strings.

The cut and splice Operators do not disrupt individual genes. Also, the mutation Operator is used only on the connection strengths.

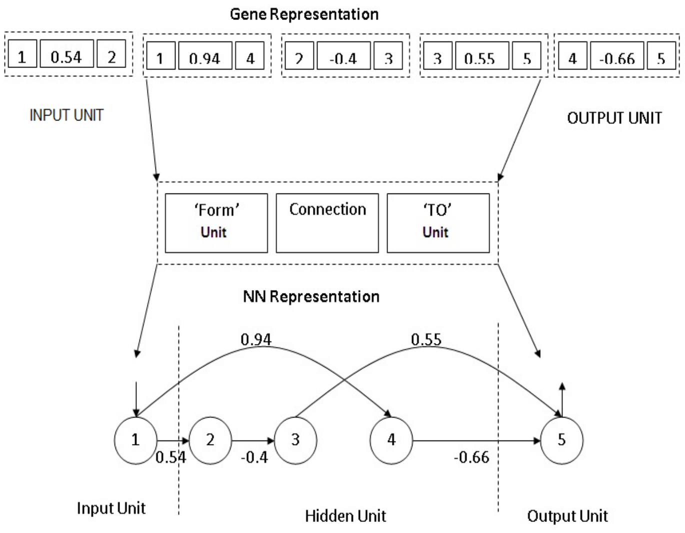


Fig.(3) Neural network genetic coding.

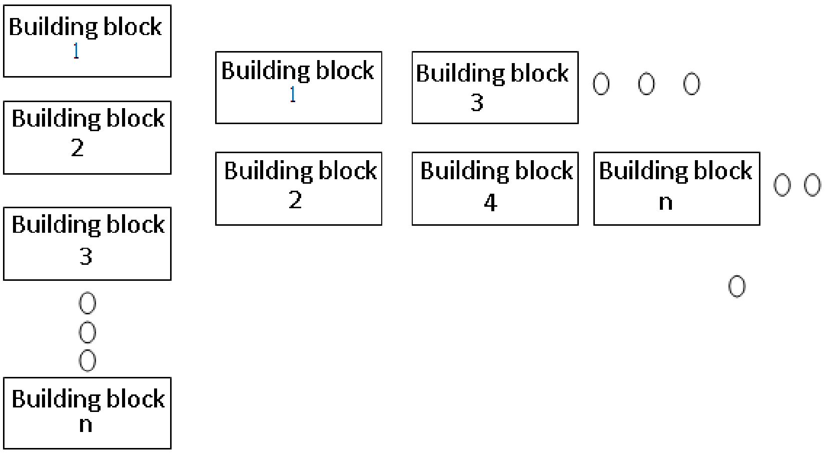


Fig.(4).Creating a population of Computational systems

2.1 The Main Steps for off-line Evolutionary Algorithm are:

1. Form a population of N strings, arrive at a near-optimal (CCS) using crossover, mutation, and selection. An on-line critic that can provide a fitness value for each candidate soluation will decide optimality. Mutation takes place only on the parameters representing the topology (for example, weights for a NN) assuming a certain probability distribution of these parameters derived from the building blocks.
2. Ideally, it will be desirable to have a population of all possible (CCS)

Synthesized from the building blocks. Since this is not possible, we select a finite population of N messy string and to ensure diversity, in every generation we introduce N/2 new messy string drown randomly from the available building blocks.

3. The (CCS) chain is then decoded and the resulting computation system is superimposed with that of the (BCS). The resulting computation system is evaluated for its optimality. The (BCS) can also be one of few possibilities. Once a near – optimal system has been found, hyper-mutation (or another preferred local leaning scheme) is applied to the parameters. In this step, no crossover is conducted, and no new string is introduced. The selection operator is retained as it increases the number of high – affinity (or high fit) (CCS) in the population.

2.2 The Adopted on-Line Processing

An evolutionary variant unlike any other technique, the (EA) achieves its objective through a mix of random swapping and coping guided by a Darwinists "Survival of the fittest" strategy. (EA) possess great similarities to biological evolution and are guided by the notion of building blocks. The operators used for the crossover phase are similar to those used by [2] cut, splice, and mutation. The cut operation is conducted with a specified gene-wise cut probability, Pk, such that the actual probability is pc=min {Pk(L-1)}, where (L) is the current string length. The splice operator concatenates two string at random with specified probability, Ps.

Mutation is conducted only on the connection strengths (weights or association) using a small mutation rare. Figure (5 presents a sample sequence of operators for a computational system. After each cut and splice operation, the strings can grow longer or shorter .Selection is carried out using a binary tournament selection scheme. In the hyper – mutation phase, the connection parameters are mutated using a probability of mutation (Phm).The probability distribution for the mutation operator is assumed to be Gaussian and is derived using building block data gathered off-line.

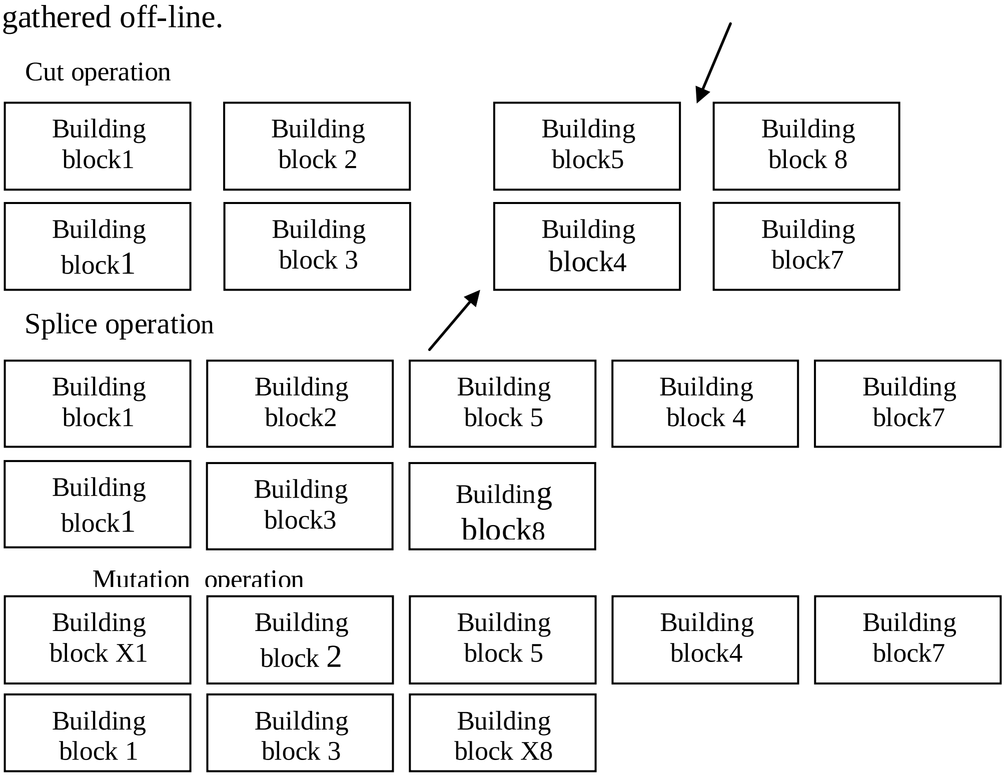


Fig.(5). Evolutionary Algorithm

3. Linear quadratic Regulator

The solution may be analytic only in certain cases and even in these cases the structure of the controlling system (controller) is often non-linear.
When the objective function is quadratic in the state and control variables, it is possible to have analytic solution and a linear controller. Therefore , the optimal response can not only be solution realized simply but it can also be reached easily (or very closely reached). Due to their symbol structure and practical relevance the algorithm represents one of the best Known and important results of modern control theory. Then the application is quite common.

4. Design Example:

For the dynamic system

$$\dot{X} = AX + Bu \qquad \text{…... (1)}$$

Where;

$$A = \begin{bmatrix} -k_1 & 0 \\ -k_2 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} k_2 \\ 0 \end{bmatrix}$$

and k_1 and k_2 are the uncertain system parameter.
Given linear system and quadratic cost function as equation :

$$J(k) = \sum_{k=1}^{M} X^T(k)QX(k) + U^T(k)RU(k) \qquad \text{……………. (2)}$$

Where, M is the number of time steps to update the control law,
R = [0.1] and $Q = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$
We have a closed form of dynamic programming solution for state feedback as follows:
$U(t) = -R^{-1} B^T SX(t)$ ………. (3)
Where; S is the solution of the Riccati equation.
$SA + A^T S - SBR^{-1}B^T S + Q = 0$ ….…. (4)

The optimal parameter solution of the steady state Riccati equation is:

$$S_{initial} = \begin{bmatrix} 0.1706 & 0.3162 \\ 0.3162 & 0.8556 \end{bmatrix}$$

Note that $S_{initial}$ is positive definent symmetric matrix . It should be noted that there must be a limits for the uncertainties in parameters values .In this work ,it is assumed that the limits of uncertainties are ($0.1 \le k_1 \le 2$),($0.1 \le k_2 \le 2$).The strengths of LQR theory is used to arrive to the optimal parameters of S matrix.

1. Population

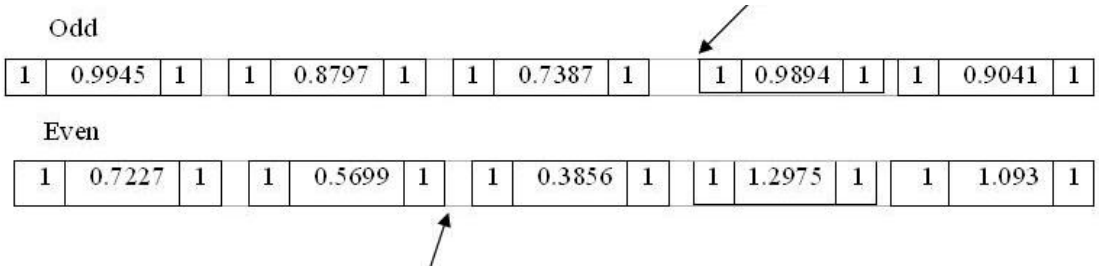
Firstly, generate *n* number of population for S matrix (S_{11}, S_{12},S_{22}) randomly as shown in Table(2).
Note that the column of S_{11} takes values, which are randomly generated with mean value equal 0.1706 (S_{11} initial) and variance 0.1 .Similarly for the column of S_{12} and S_{22}.

Table (2): Generate *n* number of population for S matrix (S_{11}, S_{12},S_{22}) .

S11	S12	S22
0.9945	0.0661	0.3555
0.7227	0.1201	0.261
0.8797	0.2477	0.7483
0.5699	0.2153	0.0128
0.7387	0.1458	0.6572
0.3856	0.1795	0.8308
0.9894	0.2511	0.8473
1.2975	0.0187	0.6751
0.9041	0.1906	0.3754
1.093	0.0159	0.4265

For first iteration:

Now, each odd and even number in the S_{11} column is a building block sorted as follows:



Odd

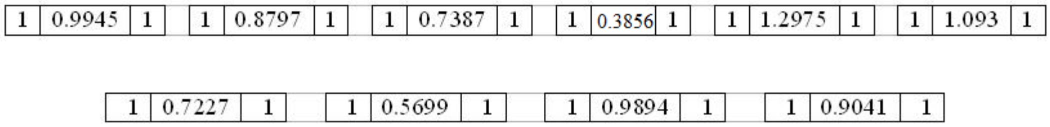
1 | 0.9945 | 1   1 | 0.8797 | 1   1 | 0.7387 | 1   1 | 0.9894 | 1   1 | 0.9041 | 1

Even

1 | 0.7227 | 1   1 | 0.5699 | 1   1 | 0.3856 | 1   1 | 1.2975 | 1   1 | 1.093 | 1

2. Recombination

The recombination step is performed as explained in section (2.2) with probability of p_c = 0.5 and specified probability p_s =0.6 as shown below:

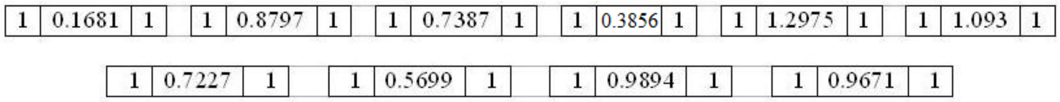


1 | 0.9945 | 1   1 | 0.8797 | 1   1 | 0.7387 | 1   1 | 0.3856 | 1   1 | 1.2975 | 1   1 | 1.093 | 1

1 | 0.7227 | 1   1 | 0.5699 | 1   1 | 0.9894 | 1   1 | 0.9041 | 1

3. Mutation

After that the mutation will take place with probability (p_m =0.09) as shown [see section (2.2)].



1 | 0.1681 | 1   1 | 0.8797 | 1   1 | 0.7387 | 1   1 | 0.3856 | 1   1 | 1.2975 | 1   1 | 1.093 | 1

1 | 0.7227 | 1   1 | 0.5699 | 1   1 | 0.9894 | 1   1 | 0.9671 | 1

Note that the mutation is performed on location (1) and (4). Similar steps will be applied to the S_{12} and S_{22} to get the value of (S_{11}; S_{12} and S_{22}) for first iteration.After the first iteration the best value of(S_{11}; S_{12} and S_{22}) are obtained in Table(3) .

Table (3): The best value of (S_{11}; S_{12} and S_{22}).

S_{11}	E(t)	S_{12}	E(t)	S_{22}	E(t)
0.1681	0.0327	0.1955	0.0615	0.271	0.0759
0.0394	0.0802	0.0587	0.0853	0.6761	0.04731
0.0839	0.662	0.3157	0.01629	0.3764	0.0709
0.1135	0.0473	0.2523	0.4911	0.4255	0.0509
0.1531	0.0386	0.1292	0.0716	0.3565	0.0712
0.1401	0.0412	0.2901	0.03567	0.8493	0.0107
0.0132	0.0865	0.1301	0.0747	0.7493	0.02287
0.0769	0.0738	0.2835	0.02911	0.0138	0.08583
0.132	0.0422	0.2513	0.04311	0.8318	0.0198
0.078	0.0721	0.2924	0.02725	0.6582	0.04872

Note that in this table calculation of the fitness function is given by

$$E(t+1)=[J(t)-J(t+1)]^2 \qquad \text{........ (5)}$$

Where,

$$J(t)=X^T (t)S(t)X(t) \qquad \text{........ (6)}$$

and;

$$J(t+1)=X^T(t+1)S(t+1)X(t+1) \qquad \text{........... (7)}$$

are evaluated according to equation (2) and for M=10 sampling time .To clarify the 1st column of the above table the following steps are performed:

1- Set S_{11} at the value of 0.1681 while S_{12} and S_{22} are fixed as those in the matrix $S_{initial.}$

2- Evaluate the fitness function as in equation 5 [the obtained value is 0.0327].

3- Repeat 1 and 2 for the whole values of the S_{11} column.

4- Select the best value of S_{11} (i.e that posse is minimum J which is 0.0327).

5- Set the value of S_{12} to the 1st value of the S_{12} column (i.e S_{12}=0.1955) while S_{22} of the S matrix is kept as in $S_{initial}$

$$\text{i.e} \qquad S=\begin{bmatrix} 0.1681 & 0.1955 \\ 0.1955 & 0.8556 \end{bmatrix}$$

6- Evaluate the fitness function for this value of S_{12}.

7- Repeat for other values of S_{12} in the S_{12} column and select the best S_{12} that posses the minimum J.

8- Set the S matrix with the best values of S_{11} and S_{12} obtained in step(4) and S_{22} as the 1st value of the S_{22} column i.e.

$$S=\begin{bmatrix} 0.1681 & 0.3157 \\ 0.3157 & 0.271 \end{bmatrix}$$

9- Evaluate the fitness function for the S matrix (i.e J=0.0759).

10- Repeated step 9 for the other values of S_{22} column and select that value which posses minimum J. therefore the best value of the S matrix becomes.

$$S=\begin{bmatrix} 0.1681 & 0.3157 \\ 0.3157 & 0.8493 \end{bmatrix}$$

Now we selected the following value of (S_{11};S_{12}; and S_{22}) which consider the initial value to the 2nd iteration
S_{11}=0.1681 S_{12}=0.3157 S_{22}=0.8493.

The initial (S) critic matrix for the 2nd interaction is

$$S = \begin{bmatrix} 0.1681 & 0.3157 \\ 0.3157 & 0.8493 \end{bmatrix}$$

After (20) iterations for (regulating case) that should be accomplished On-line during M Sampling times

$$S_{best} = \begin{bmatrix} 0.1704 & 0.309 \\ 0.3157 & 0.849 \end{bmatrix}$$

According to equation below:
$K(t)= -R^{-1} B^{T} S(t)$,
which represent the state feedback gain vector. This algorithm is repeated along 20 iterations K=[1.7064 3.1623].
The overall on-line artificial immunized system for regulating case is shown in figure (6)

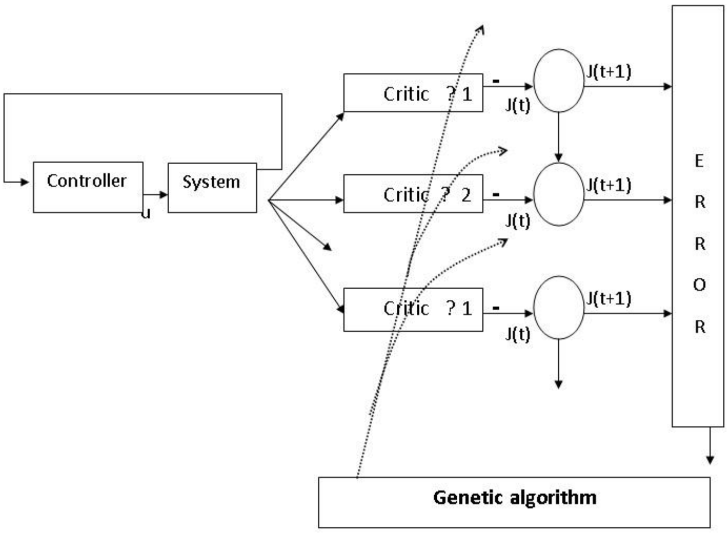


Fig (6): Immunized adaptive critics on-line Implementation. (Regulating case)

The overall concepts of tracking case is shown in figure (7).The two-state responses for regulating case are illustrated in figure (8) in which a conventional PI controller is used. The parameters of the PI controller are added as extra critics to be determined genetically. Each M sampling time firstly obtained the k_p and k_i of the PI controller before selecting the best value of the matrix S. After (20) iteration performed each M sampling time the best PI and S parameters are:

$$S = \begin{bmatrix} 0.1362 & 0.1362 \\ 0.3162 & 1.3162 \end{bmatrix}$$

$$K = \begin{bmatrix} 3.1623 & 3.1623 \end{bmatrix}$$

$$K_p = \begin{bmatrix} 0.005 \; 0.1 \end{bmatrix}$$

$$K_i = \begin{bmatrix} 0.08 \; .007 \end{bmatrix}$$

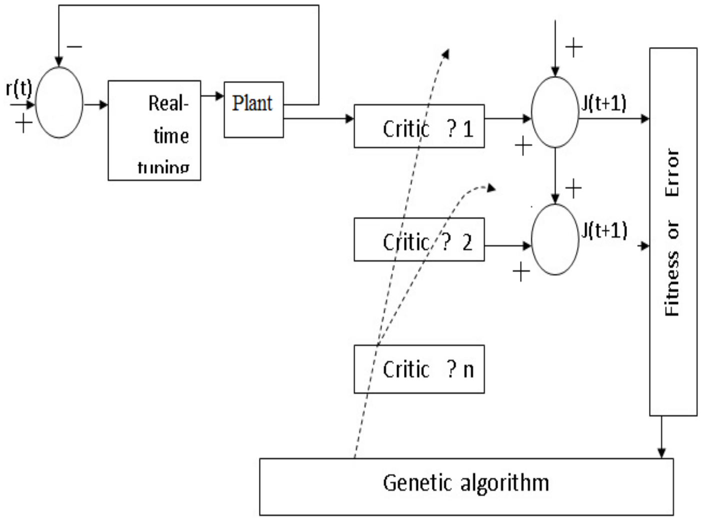


Fig. (7). Immunized adaptive critics on – line Implantation. (Tracking case)

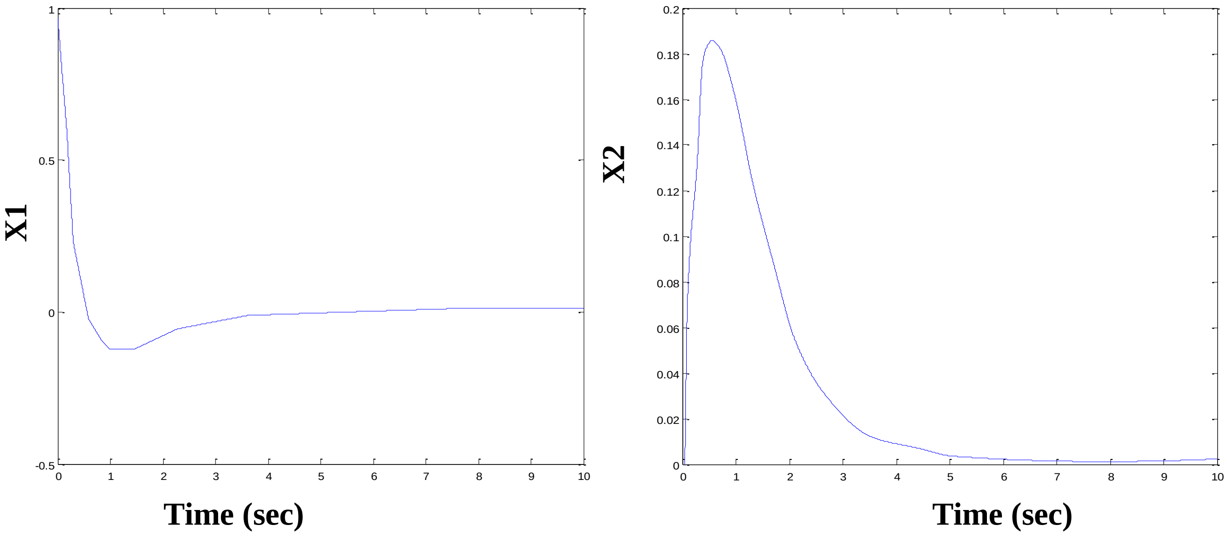


Fig.(8) : The two states response of example (Regulating case).

The two-state response for tracking case is shown in figure (9). Figure (10) shows the gain vector (k) parameters for (tracking case).Figure (11) shows the PI vectors parameters.

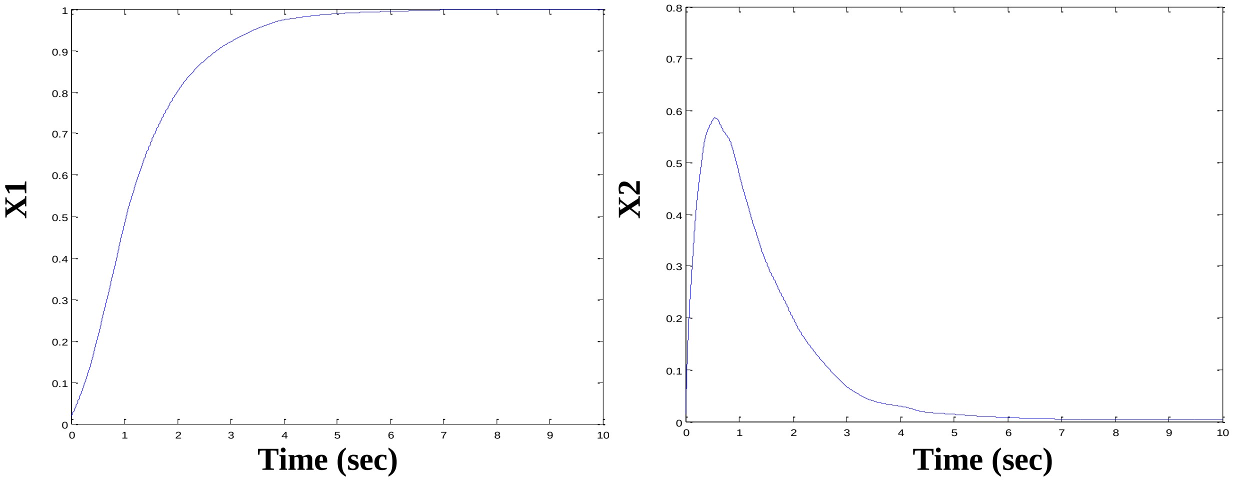


Fig.(9) : The two states response of example (Tracking case).

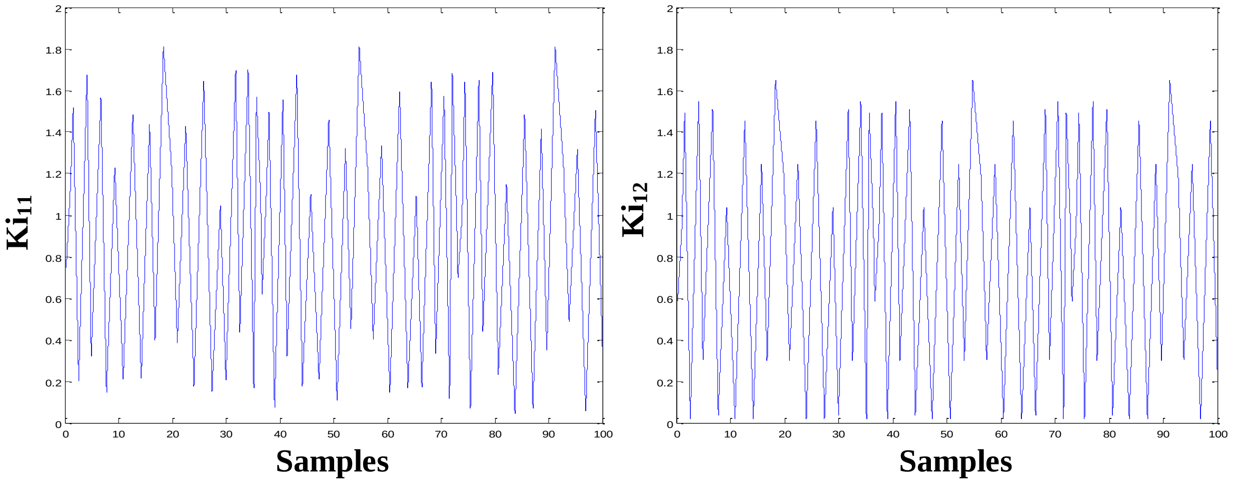


Fig.(10) : The variation of gain matrix K (Tracking Case).

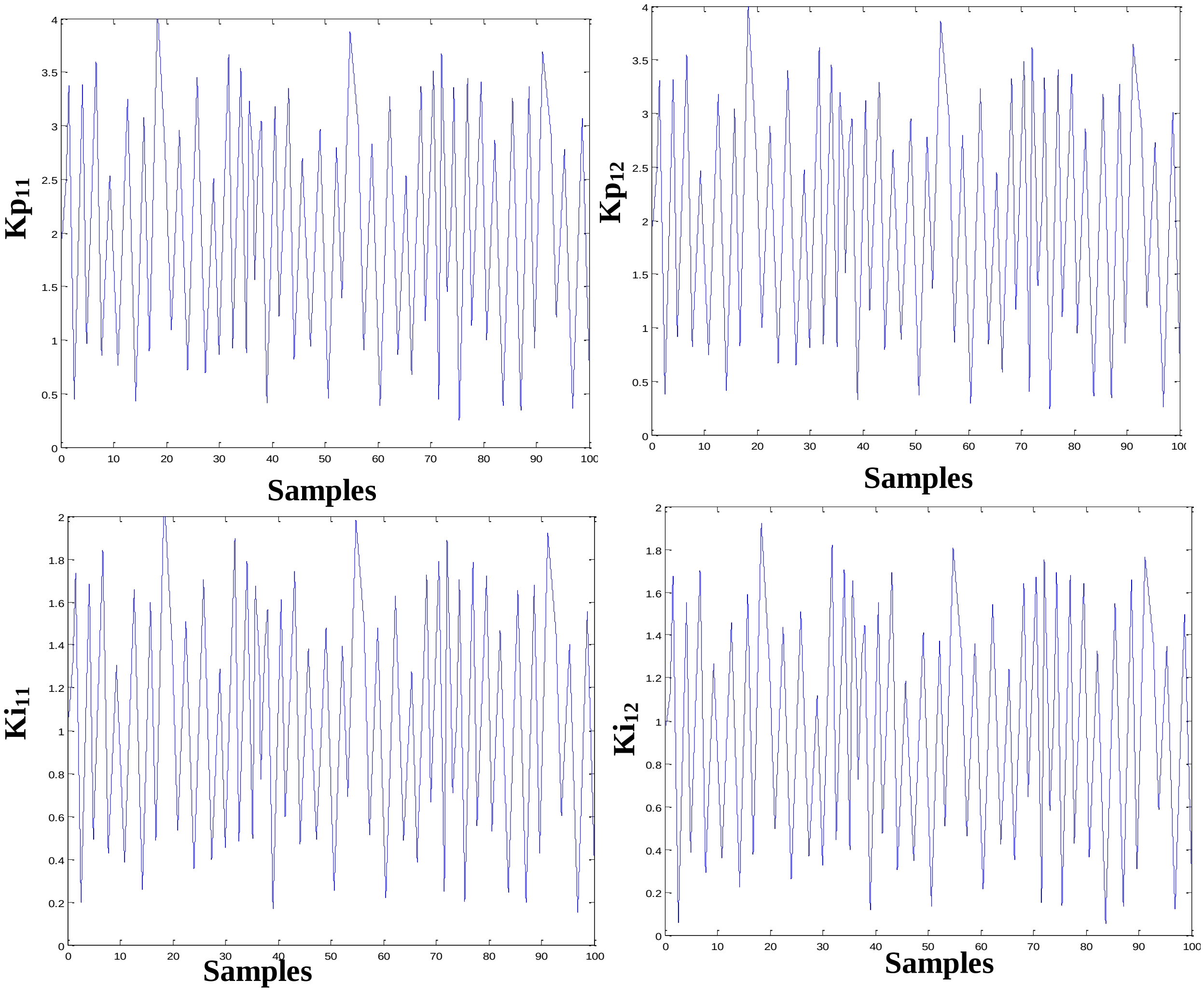


Fig. (11): The variation of PI controller (Tracking Case)

5. Conclusions

The AIS is a step forward toward applying the Evolutionary Algorithms (EA) on-line in control design problems that faces plant parameter variations. In deed the time required for the Genetic Algorithms (GA) operator to perform many generations is not applicable to be done during one sampling time. Therefore, the adaptation has to be accomplished during M sampling time, where M is depend on the variation on the plant parameters , degree of non-linearity ,and the processor speed. It is found that the proposed AIS that vary the coefficient of the state feedback gain each M sampling time is applicable and good adaptable regulating and tracking performances are obtained even in the case of having plant parameter variations.

References:

[1] KrishnaKumar,K , " Artificial Immune System Approaches for Aerospace Applications", 41st Aerospace Sciences Meeting & Exhibit (6 - 9) January, Reno, Nevada, AIAA- 0457, 2003.

[2] Belkacemi,R and Feliachi,A," Immune Based Design of a Load Frequency Controller in Power Systems, Journal of Electrical Systems, 4-2,page 1-12,2008.

[3] Yanfei Zhong, Liangpei Zhang, Bo Huang, and Pingxiang Li , " An Unsupervised Artificial Immune Classifier for Multi/Hyper spectral Remote Sensing Imagery", IEEE Transaction on Geosciences and Remote Sensing , Vol. 44, NO. 2, February, 2006.

[4] Xuanwu Zhou," Research on Immune Pathology in Artificial Immune System", Chinese Control and Decision Conference (CCDC 2009).

[5] K. Kirshnakumar ‘‘Immunized Artificial system- concepts and applications’’, Department of areospace engineering and mechanics Tuscaloosa , 1998.

[6] KrishnaKumar, K. Satyadas, A. & Neidhoefer, J," An Immune System Framework for Integrating Computational Intelligence Paradigms
with Applications to Adaptive Control", In Computational Intelligence A Dynamic System Perspective, (Eds.) M. Palaniswami, Y.Attikiouzel, R. J. Marks II, D. Fogel and T.Fukuda, IEEE Press, pp. 32-45,(1995)..

The work was carried out at the College of Electronics Engineering. University of Mosul