AIAA-2003-0457

Artificial Immune System Approaches for Aerospace Applications

K. KrishnaKumar
NASA Ames Research Center
Moffett Field, CA

41st Aerospace Sciences Meeting & Exhibit
6 - 9 January 2003
Reno, Nevada

Artificial Immune System Approaches for Aerospace Applications

K. KrishnaKumar
Research Scientist, NASA Ames Research Center, Moffett Field, CA.
kkumar@mail.arc.nasa.gov

Abstract

Artificial Immune Systems (AIS) combine *a priori* knowledge with the adapting capabilities of biological immune systems to provide a powerful alternative to currently available techniques for pattern recognition, modeling, design, and control. 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 "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. In this paper, we outline AIS models that are immediately applicable to aerospace problems and identify application areas that need further investigation.

1 Introduction

Biological immune systems use a complex of cells, molecules, and organs to overcome foreign invasion of living beings [1, 2]. In doing so, they have proven to be capable of performing several tasks, like pattern recognition, learning, detection, optimization, etc. Biological immune systems use a finite number of discrete detector "building blocks" derived from the DNA molecule to achieve these capabilities. These detector blocks, that are evolved, can be thought of as pieces of a puzzle that must be put together in a specific way to neutralize, remove, or destroy each unique disturbance the being encounters. One can attempt to define and learn these detector blocks for various engineering problems of interest. The detector blocks can be processed using the features from the biological immune systems captured into computational models.

A substantial amount of research in intelligent systems has concentrated on models of intelligence and learning as they occur in human beings. This research often overlooks intelligent systems that are not *explicitly* related to the processes of human brains and minds. A good example from nature that is not anthropomorphic, but still exhibits high levels of intelligence, is the biological immune system. The immune system actively exploits memory (both long term and short term), executes strategies, divides tasks hierarchically, recognizes patterns, deals with unforeseen conditions, adapts to changing conditions, etc. However, its operation is markedly different from that of a neural or symbolic processing system.

How does the immune system metaphor fit under aerospace applications? There are several problem domains in aerospace engineering that need complex solutions. For example, future needs of space exploration, homeland security, etc will need novel aircraft designs that are drastically different from currently available designs. Although these designs will be different, they will have to exploit the current knowledge bases that exist in aerodynamics, controls, etc. To achieve these complex designs, one can define and learn computational (or information) building blocks, off-line and on-line, to design complex solutions for problems at hand. The building blocks are processed on-line for system adaptation. These building blocks can be identified using either learning or by incorporating *a priori* knowledge. Various immune system features such as clonal selection, bone marrow models, etc can be used to process these building blocks to arrive at good design choices. In addition, various AIS features can be combined to form a robust design tool for a wide variety of design problems.

Another potential area will be security of computational systems that drive the multitude of functions on current and future aircraft and spacecraft. The dangers (such as computer viruses) are ill defined and approaches based on the immune system are ideally suited.

There are several significant benefits of AIS approaches as compared to other artificial intelligence approaches currently being studied. These are:

Vast a priori knowledge: There is a vast amount of knowledge available in the DNA molecule for arriving at solutions in the adaptive immunity. The idea of efficiently storing and using the a priori knowledge is a powerful metaphor.

Rapid evolution from an acceptable solution to an optimal solution: In AIS, the search for a solution is modeled after the generation of an immune response wherein the optimal solution is achieved by rapid mutation and recombination of a genetic representation of the solution space. During the generation of the immune response, the system receives a continuous feedback from the antigen-antibody complex resulting in a generation of an increasingly specific antibody response. This represents a learning paradigm that is used in AIS to develop solutions that continually increase in accuracy.

Learning: The learning paradigm in AIS is based on the interaction between populations of antibodies and antigens. This provides a unique way of arriving at self-organizing network structures.

Uniqueness and variety: Each AIS response is unique implying an enormous number of possibilities in the available solutions.

Representation transparency: It is easy to accommodate both objects and continuous numbers in the same representation enabling an easy way to mix various data types.

Memory: The ability to create new solutions in a short time and inherent memory management are other attractive features of AIS.

Other attractive features include:

- Robust Recognition
- Reinforcement learning
- Distributed/parallel processing
- Multi-layered
- No centralized control
- Self-tolerance

2 Immune System-A Short Description

The immune system is made up of two major divisions (see Figure 1), the innate immune system, and the adaptive immune system. The innate immune system is composed of static defenses such as skin and mucus that serve to separate the individual from potential threats. These are supplemented by pre-formed biochemical barriers and other defensive elements such as phagocytes that are widely distributed in the blood and body tissue. These also have the ability to signal the appearance of a threat and call in more of these pre-formed elements. All of these elements are broadly reactive to general categories of problems and have a limited and predetermined set of responses. If the defenses of the innate immune system are breached, the adaptive immune system is called upon to produce a specific reaction to the infectious agent.

The adaptive response is driven by the presence of the threat and those cells that nullify the threat the most effectively receive the strongest signal to replicate. The basic components of the immune system are white blood cells, or lymphocytes. Lymphocytes are produced by the bone marrow. Some lymphocytes only live for a few days and the bone marrow is constantly making new cells to replace the old ones in the blood. There are two major classes of lymphocytes: B-cells, produced in the *bone marrow* in the course of so-called clonal selection (described later), and T-cells, processed in the *thymus*. B-lymphocytes secrete antibodies and some B-cells survive as memory cells. T-cells are concerned with cellular immunity: they function by interacting with other cells. T-cells divide into helper T-cells, that activate B-cells, and killer T-cells, that eliminate intracellular pathogens. Activated B-cells present pieces of the antigens to killer T-cells.

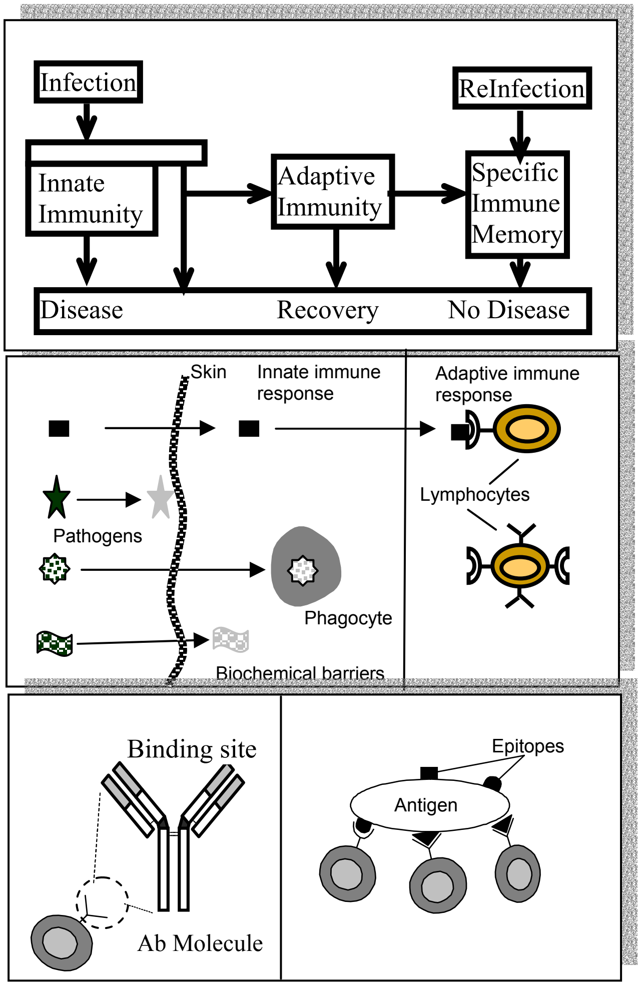

Figure 1. Immune System Functional Flow (Top); Layers of Defense in the Immune System (Middle); Immune Recognition (Bottom)

The immune recognition is based on the *complementarity* between the binding region of the receptor and a portion of the antigen called *epitope*. Antibodies do not bind to the whole infectious agent, but rather to one of the many molecules on the agent's surface. This means that different antibodies can recognize a single antigen (See Figure 1).

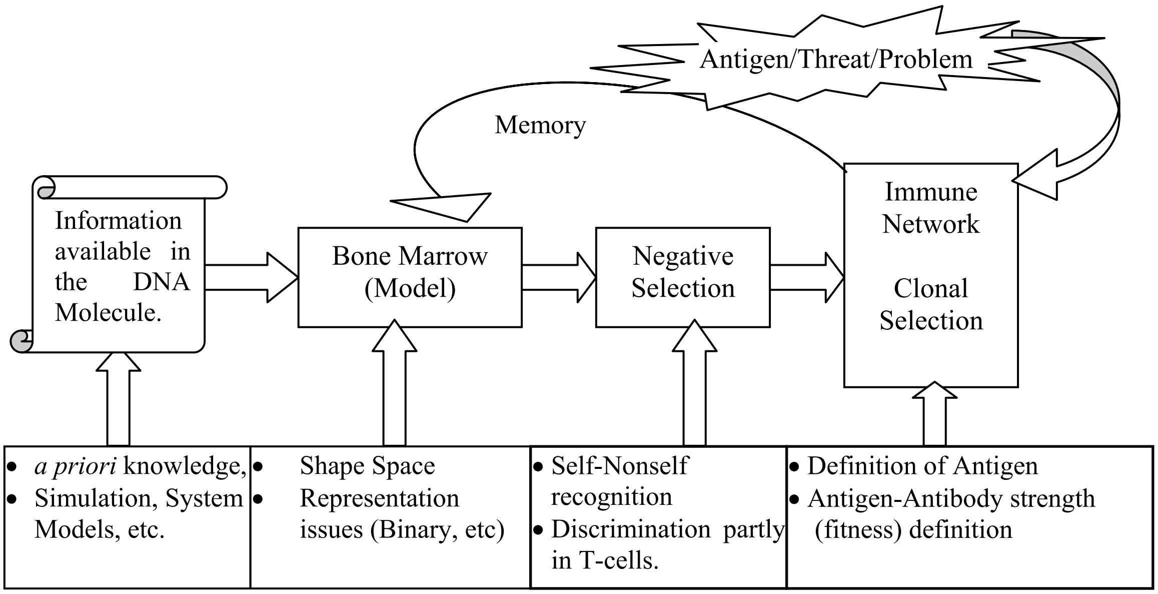
Antigen/Threat/Problem
Memory
Information available in the DNA Molecule.
Bone Marrow (Model)
Negative Selection
Immune Network
Clonal Selection
• *a priori* knowledge,
• Simulation, System Models, etc.
• Shape Space
• Representation issues (Binary, etc)
• Self-Nonself recognition
• Discrimination partly in T-cells.
• Definition of Antigen
• Antigen-Antibody strength (fitness) definition

Figure 2. A system-level description of the Immune System Metaphor

Antibodies are essentially bifunctional molecules with a variable (V) region and a constant (C) region. It is the variation in the (V) domain that gives the immune system the power and speed in the adaptation process. Studies conducted to examine the antibody diversity produced during the immune response have demonstrated that the number of somatic mutations in the (V) region increases with time. This increase in somatic mutations correlates with an increase in antibody affinity for the antigen. Hypermutation (high levels of mutation), although an important factor in immune system maturation, by itself is not sufficient. Actually, assuming that the hypermutation mechanism is totally random, many of the mutations will destroy the affinity for the antigen. One way the immune system overcomes this is by selectively increasing the population of the high affinity antibodies. Thus, selection also plays a major role in determining high-affinity antibodies.

3 Immune System Computational Models

Figure 2 presents a system-level description of the immune system metaphor. There are several computational models that are based on the principles of immune systems. These are:

- Bone marrow models
- Negative-selection theorem
- Clonal Selection Algorithm
- Immune Network model
- Immunized Computational Systems

The assumption of usability of these models is preceded by the assumption that some understanding of the problem exists. This is akin to the vast source of information available to the immune system. Once this knowledge exists, one can use the immune sub systems (see Figure 2) individually or in combination.

In the next subsections, we present details of these sub systems (and combinations of these subsystems) and problems in which they are suited for.

3.1 Bone Marrow Models

In the bone marrow models, the following ideas are encapsulated:

- Gene libraries are used to create antibodies from the bone marrow (Gene examples are shown in Figure 3).
- Antibody production is through a random concatenation of genes from the gene library (Figure 4)

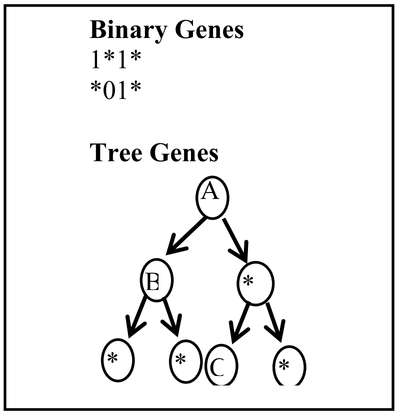

Figure 3. Different gene representations

- Simple or complex libraries can be used
- Antibodies produced are evaluated using a fitness function that defines the affinity of the designed antibody.

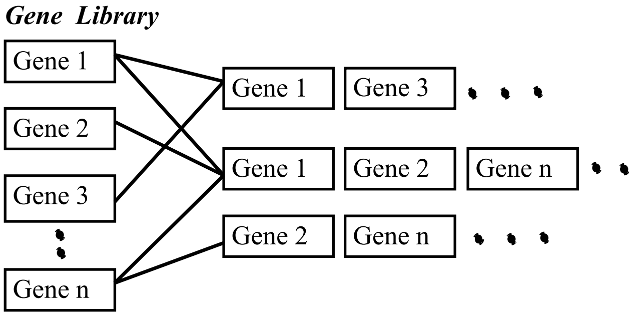

Figure 4. Creation of solutions using the Gene Library

The gene library contains pieces of a solution that has been predetermined using a priori knowledge. Representations could include simple Binary strings to more complicated neural chunks. Examples of two different types of Genes are shown in Figure 3. Figure 4 presents the formation of the antibodies from the gene library.

Once the solutions are produced from the library, they can be processed using standard genetic operators.

3.2 Negative-selection Algorithm

Forrest et al [8] developed this algorithm based on the principle of self-nonself discrimination in the immune system. This discrimination is achieved in part by T-cells, which have receptors on their surface that can detect antigens. T-cells are generated by a random genetic rearrangement process and then they undergo a censoring in the *thymus* where the T-cells that react against self-proteins are destroyed. This algorithm is summarized as follows:

- Define self as a collection *S* of strings of length *L* over a finite alphabet--Needs to be protected.
- Generate a set *R* of detectors, each of which fails to match any string in *S*
- Monitor *S* for changes by continually matching the detectors in *R* against *S*. If any detector is matched, then a change is known to have occurred. Candidate detectors can be generated randomly or in an intelligent fashion.

Some of the applications of negative selection include: Color Image Segmentation [11]; Anomaly detection in time-series data [4, 6]; and Computer virus detection [5, 9].

3.3 Immune Network Model

In the immune network theory, originally proposed by Jerne [see references 1, 2, 20, & 21], antibodies recognize both antigens and other antibodies. Antibodies recognizing other antibodies form a network within the immune system. It is interesting to note that this theory does not require the presence of antigens to stimulate an immune response. The dynamics are governed by both the presence of antigens and antibodies. The figure below shows the basic principle of the immune network theory.

As the antibody matures, it recognizes the antigen with a higher degree of accuracy (see clonal selection principle below). Once the antigen is completely removed, the network between like-antibodies helps in keeping the immune system from extinguishing itself. A stable population is maintained as the memory that will be useful for future encounters of similar antigen.

This network of B-cells occurs due to the matching of the paratopes against the idiotopes on other B-cells. As shown in Figure 5, the cell (not shown) producing antibody #1 has a complementing idiotopes for antibody #2. Similarly antibody #2 has a complementing idiotope for cell #3. Same antibody could interact with more than one antibodies. This interaction actually produces a network of coupled stimulation-suppression phenomenon that maintains a stable equilibrium of good antibodies for future use. The increase or decrease (dynamics) of the concentration of a set of lymphocyte clones and the corresponding B-cells can be simulated using a non-linear differential equation.

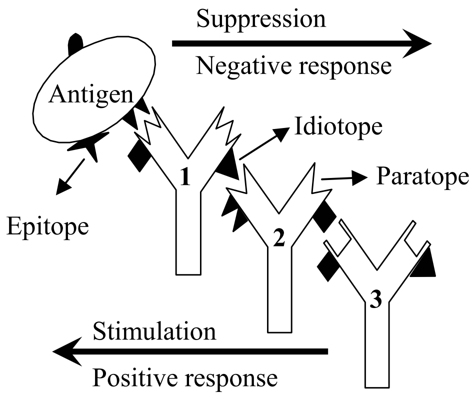

Figure 5. Immune Network suppression and stimulation

Applications of Immune network models range from structural design [12] to fraud detection [3] to robotics [7].

3.4 Clonal Selection Algorithm

A distinct difference between biological evolution and evolution based on clonal selection principle is the time scales. The goal of the clonal selection in the immune system evolution is to find the most suitable member of a population in very short periods of time.

The clonal selection algorithm uses three basic operators: (1) selection, (2) cloning, and (3) maturation. The operators are implemented to perform the basic tasks of discovering and maturing good antibodies from the population of available solutions in an orchestrated fashion.

The cloning operator is a process in which antibody with high performance receive correspondingly large numbers of copies in the new population of antibody. In cloning, the antibody with high performance is given higher probability of reproduction.

The maturation operator enhances the ability of the algorithm to tune the antibody in the population. Maturation is the occasional alteration of antibody at a particular part by using the method called hypermutation (high levels of mutation). This procedure is designed to prevent the permanent loss of specific information in the antibody, and thus allows for the diversity of the antibody. Although hypermutation is an important factor in immune system maturation, by itself is not sufficient. The reason is that the hypermutation mechanism is totally random and result in destroying the affinity for the antigen. Therefore, the immune system must produce a large population of the high affinity antibodies to overcome the information loss.

Together, the three operators of selection, cloning, and maturation provide an effective mechanism for searching complex spaces. An algorithm is outlined below.

Clonal Selection Algorithm

(1) Generate an antibody population either randomly or from a library of available solutions.
(2) Select the *n* best performing antibody population by evaluating a performance index.
(3) Reproduce the *n* best individual by cloning the population.
(4) Maturate the Pc antibodies by hypermutation.
(5) Re-Select the best performing antibody population;
(6) Stop if antibody generates satisfactory performance. Otherwise, start over from (1) using Pm (probability of mutation).

3.5 Immunized Computational System

Immunized computational systems (ICS) incorporate bone marrow models along with clonal selection to reproduce the robustness and adaptability of a biological immune system. The concept of a gene is replaced with the concept of a building block (described later). Figure 6 presents a functional block diagram of the immunized computational system. The system proposed by the author [13-17] has the following attributes:

Known Solutions, etc.: The possible solutions contain both a base-line computational system (BCS) and a changeable computational system (CCS). The computational building blocks, defined later, are used to arrive at the changeable computational systems. The BCS is designed to represent an average behavior of the solution (this could be known or approximated). Since this design is carried out off-line, any standard technique can be used for its synthesis. The base-line system is analogous to static portion of the antibodies. The CCS represents the variable region of the antibody and epitope equivalents. This structure must be adapted on-line. To include the innate immunity equivalence,

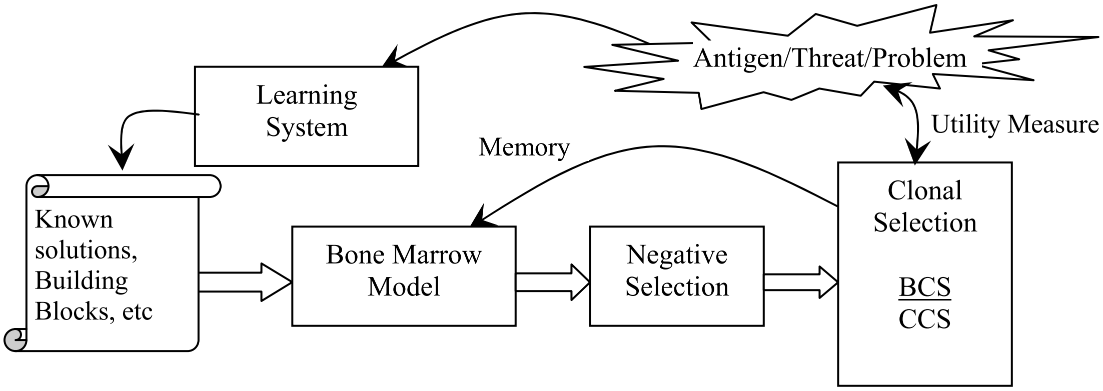

Figure 6. Immunized Computational System Architecture

the changeable computational systems that are known *a priori* can be stored in look-up tables and can be used to produce the right antibody and CCS models. Similar to the variable region of the antibodies, the changeable computational systems provide diversity to the immunized computational system.

Clonal Selection (Exploratory System): The exploratory system is basically an evolutionary algorithm variant that uses recombination, selection, and mutation to arrive at a suitable CCS.

Learning System: The learning system consists of a suit of learning paradigms to learn and store important computational building blocks that are not available a priori.

Utility Measures, etc: This component provides the definition of what is a good solution for the problem at hand.

3.5.1 Computational Building Blocks

Computational building blocks are defined as segments of a computational system topology (for example, a neural network (NN) connection, or family of connections, along with its associated weights) that contributes in establishing a good mapping for a class of input-output characteristics. Building blocks can be of different order. Examples of building blocks using neural connections are shown in Figure 7. The building blocks are specified using a universal representation scheme that uses a neuron as the basic processing element. Thus the order 1 building block consists of two neurons and the relationship between them. In the case of neural networks, the relationship is the connection strength and the neurons are characterized by their type (input, hidden, output), the type of aggregation, and activation functions. In the case of a fuzzy system, the relationship is AND, OR, or THEN, and the neurons represent the input or output variables with their associated parameters (such as the fuzzy membership function). Determining important building blocks is problem dependent and *a priori* knowledge will guide this process.

3.5.2 The Role of Evolutionary Algorithms

Analogous to the DNA molecule, the computational building blocks need to be coded as a string of building blocks. An example of a neural network with the corresponding genetic representation is shown in Figure 8. It may be noted that for a fuzzy system, the variables are the input or output variables with their associated fuzzy membership function (FMF) and other parameters (such as implication, aggregation, and defuzzification operators) and the relationship could be AND, OR, or THEN (indicates beginning of the consequent part of a rule). In the case of a neural network, this relationship will be in the form of the connection strength between two neurons. For on-line processing, a population of CCS is randomly constructed using juxtapositioning of the building blocks, forming a population of strings similar to the concept of the bone marrow model presented earlier (See figure 4). Next, we find the best string that will represent the CCS via clonal selection. Typical steps are as follows:

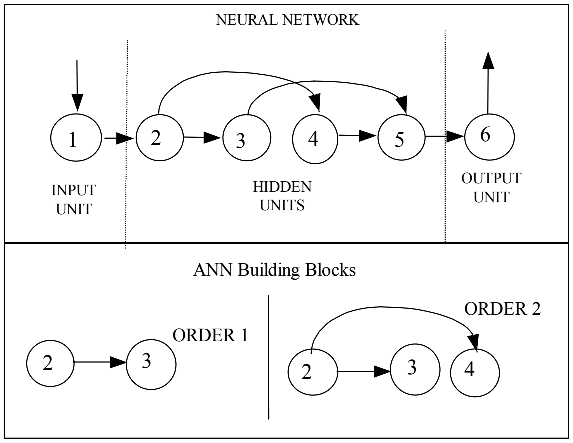

Figure 7. NN Computational Building Blocks

- From a population of N strings, arrive at a near-optimal CCS using recombination, mutation, and selection. An on-line critic that can provide a fitness value for each candidate solution 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.
- Ideally, it will be desirable to have a population of all the possible CCS synthesized from the building blocks. Since this is infeasible for most applications, we select a finite population of N strings and to ensure diversity, in every generation we introduce N/2 new strings drawn randomly from the available building blocks (other innovations are possible).
- The CCS chain is then decoded and the resulting computational system is superimposed with that of the BCS. The resulting computational system is evaluated for its optimality. The BCS can also be one of few possibilities. Once a near-convergence system has been found, hyper-mutation (or another preferred local learning scheme) is applied to the parameters. In this step, no recombination is conducted, and no new strings are introduced. The selection operator is retained as it increases the number of high-affinity (or highly fit) CCS in the population.

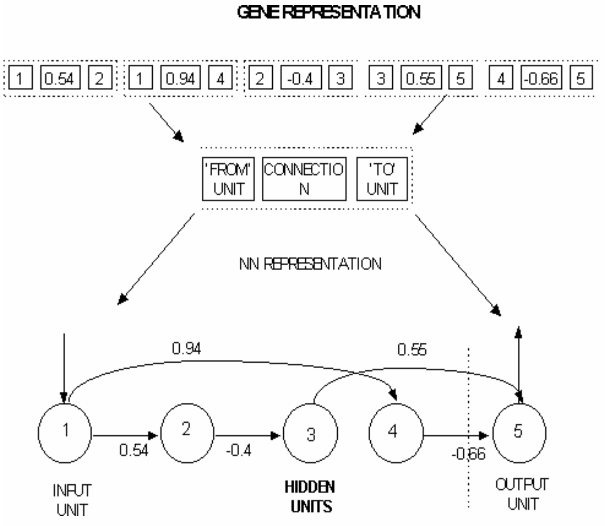

Figure 8. Neural Network genetic coding. The cut and splice operators do not disrupt individual genes. Also, the mutation operator is used only on the connection strengths.

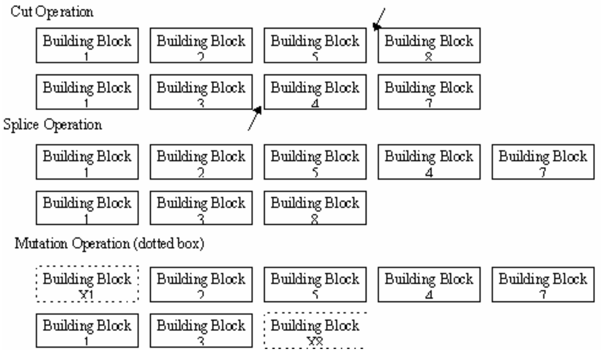

Figure 9. Evolutionary Algorithm Operators

The actual on-line processing is conducted using clonal selection. The operators used for the recombination phase are cut, splice, and mutation. The cut operation is conducted with a specified gene-wise cut probability, p_k, such that the actual probability is $p_c=\min\{p_k(L-1),1\}$, where L is the current string length. The splice operator concatenates two strings at random with specified probability, p_s. Mutation is conducted only on the connection strengths (weights or association) using a small mutation rate. Figure 9 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 p_{hm}. The probability distribution for the mutation operator is assumed to be Gaussian and is derived using building block data gathered off-line.

The ICS model has been successfully applied to adaptive control problems in Reference [13-17].

4 Applications Domains

In the past few years, AIS has been successfully applied to many different engineering problems. Some of these include:

- Security and virus detection [5,9,24]
- Fault identification [3]
- Anomaly detection [4,6]
- System ID [15]
- Adaptive control [13-17]
- Design [12]

There are few issues that a user must address in getting ready to use any of the AIS computational models. These include:

- Representational issues such as the choice of the alphabet (Binary, integer, tree structure, neural networks, etc) and proper coverage of the operating space.
- Description of how antigens and antibodies interact in terms of an affinity measure. Affinity measures could be in terms of Euclidean distances, Hamming distances, or a more complex definition of their interaction via a fitness function.

In the next few paragraphs, we outline areas in aerospace where the AIS paradigm is well suited as an alternative to existing techniques.

4.1 Automated Design

Aerospace systems of tomorrow will be complex and their design interdisciplinary. For example, design optimization conducted individually on subsystems such as wing, propulsion, and control will not integrate without extensive redesign. A unified design that integrates all facets of a system is difficult if not impossible to find using current optimization techniques. Traditional design approaches rely on cut-and-try approaches adopted by designers that can be conducted very well using high performance computers. The bottleneck for computer implementation is the lack of (1) a universal representation of the design features and (2) a procedure for the design features to be cut-and-tried *by a computer* in an optimal way. There are other potential problems associated with computer-based automated design. These are:

- Time required for a system simulation is large prohibiting use of full fidelity simulation of the problem.

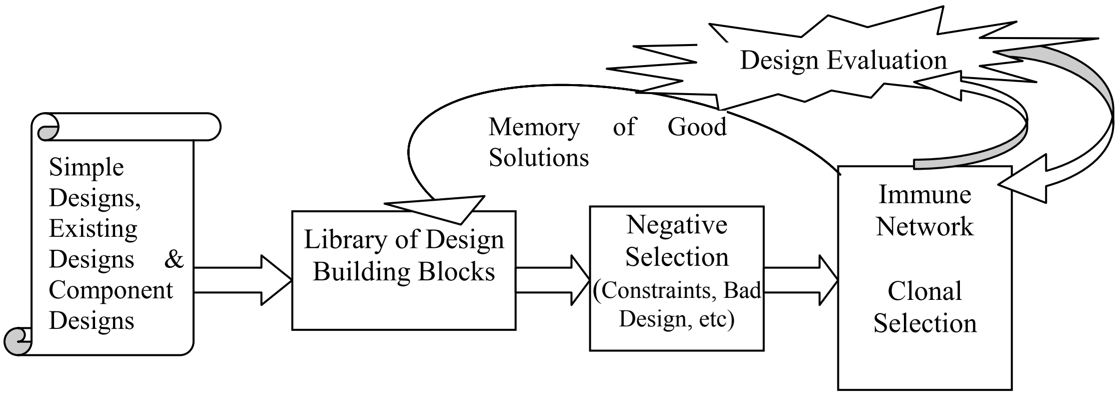
Design Evaluation
Memory of Good Solutions
Simple Designs, Existing Designs & Component Designs
Library of Design Building Blocks
Negative Selection (Constraints, Bad Design, etc)
Immune Network
Clonal Selection

Figure 10. A system-level description of Automated Design using AIS

- Analytical derivatives of the objectives of design are frequently unavailable and numerical gradients are expensive to compute.
- Design space consists of both continuous and/or discrete parameters
- Design response surface is non-linear, discontinuous, or undefined in some regions. Presence of several local extrema is common in many applications.
- Final design has to satisfy multiple objectives

Although the choice of using AIS technologies for solving the design problems stated above is not obvious initially, a closer look reveals several similarities between the problem of counteracting an external threat and the problem of automated design. After all, biological immune system is very good at designing antibodies for various types of antigens. Figure 10 presents a concept-level diagram of an AIS-based design system. The discussion in the next paragraphs point to various features of the system.

To obtain a universal representation, we introduce the concept of a design building block. A design building block is a way to represent a feature as an input-output function with tunable parameters. These functions can be polynomials, pieces of a neural network, look-up tables, etc. These building blocks are identified first and a library of these building blocks is constructed. This library is continuously updated as more and more designs are created. This library is then used to construct design solutions for existing problems using clonal selection algorithm and other AIS features.

We see several critical benefits of this approach to the design community. These are:

- Library of Design building blocks: Library techniques are commonly used in integrated circuit (IC) design packages. This greatly reduces the number of combinations with which the designer must deal, thereby speeding up the search process. The design building blocks concept tailors this idea to the specific design.
- Innovation is achieved by the use of evolutionary search with stochastic operators (Clonal selection). This greatly reduces the chance of reaching a local solution and solves many of the problems associated with gradient search techniques.
- Long-term memory can be provided to the design package to remember good design solutions for later use via the use of the immune network theory and other micro-features of the immune system.

Yoo and Hajela [12] have investigated the use of AIS techniques for multicriterion structural design. In their approach, clonal selection along with explicit definitions of antigens as specific points on the Pareto set was used to arrive at antibodies (good solutions) that satisfy the design criteria in a generalized way.

4.2 Vehicle Health Management

Vehicle Health Management (VHM) is the capability to efficiently perform checkout, testing and monitoring of air and space transportation vehicles, subsystems and components before, during and after operation. This includes the ability to perform timely status determination, diagnostics and prognostics. Areas of VHM in which the AIS paradigm is well suited are:

- Sensor validation
- Anomaly detection
- Real-time and post-test event detection algorithms.
- Virus protection for on-board computer systems

Anomaly detection using AIS has been the most active research area related to AIS and VHM. Dasgupta et al [4, 6] have investigated the use of negative selection algorithms for anomaly detection with good success. In their study they have used both binary and real numbers to represent the antigen and antibody equivalents.

One of the benefits of using an AIS approach is in the choice of choosing the detector sets based on either, (a) normal behavior (where it is well known and the detector sets are small in size) or (b) abnormal behavior (where normal behavior has too many characteristics to capture into detector sets). The normal behavior of a system is often characterized by a series of observations over time.

The problem of detecting novelties, or anomalies, can be viewed as finding deviations of a characteristic property in the system. For computer scientists, the identification of computational viruses and network intrusions is considered one of the most important anomaly detection tasks. This problem will be prevalent in future aerospace systems in which computer networks will manage most of the tasks now currently done by the pilot and ground crew.

4.3 Real-time maneuvering

Typically, for tactical maneuvering, the pilot compares the commanded flight-path with that of the current aircraft and selects the maneuvers capable of achieving that command [23]. Pilots use their knowledge of aircraft capabilities and of near-optimal maneuvering strategies in order to select the necessary actions. For unmanned flight or for highly automated man-in-the-loop flight, an immunized computational system as outlined in section 3.5 could be used for tactical maneuvering. Figure 11 presents the immunized maneuver selection concept. A maneuver database, also representing the “experience” of a pilot, could be used to provide pre-canned or automatically generated, maneuvering elements and established sequences. Vehicle models can be used to provide the necessary predictive information for decision-making, representing the equivalent of a pilot’s “understanding” of the internal performance of the aircraft. Appropriate flight modes and targets would be sent to the autopilot system, when necessary to initiate the desired actions.

5 Summary and Future Directions

This paper presented several possible models of AIS that can be immediately used in some of the problem domains identified. AIS is a powerful alternative to many existing techniques in the arena of biologically inspired computing paradigms. More work is needed in certain core areas of the biological immune systems to fully exploit the IAS metaphor. The applications of the future will also rely on successful modeling of some of the underlying reasons for the success of the biological immune systems. One area of immediate need is the ability to arrive at a complete set of building blocks (completeness of the repertoire) that cover the operational space.

Attempting to prove that an available set of building blocks along with recombination can completely cover the operating space is a difficult task. Letting the size of this set to be "P" and with "N" total number of building blocks needed to define a complete system, we have P^N number of possible systems. If for example P=20 and N=200, we have 20^{200} possible systems. At this point it is clear that a space of this size cannot be stored a priori. If we define "K" to be the number of building blocks that are correlated, we can see that for K=0, one can, with at most P*N mutations find the optimal system. Even this takes 4,000 evaluations. Now, if K=N, the extreme case, we have a nearly impossible task at hand. There are two things that make this picture not as bad as it seems: (a) for each building block, only a subset of the P values will be important; and (2) some of these can be pre-learned and stored as building blocks. Still some questions remain to be answered:

- For a given problem, what is the average size of the building block necessary to effect a quick adaptation

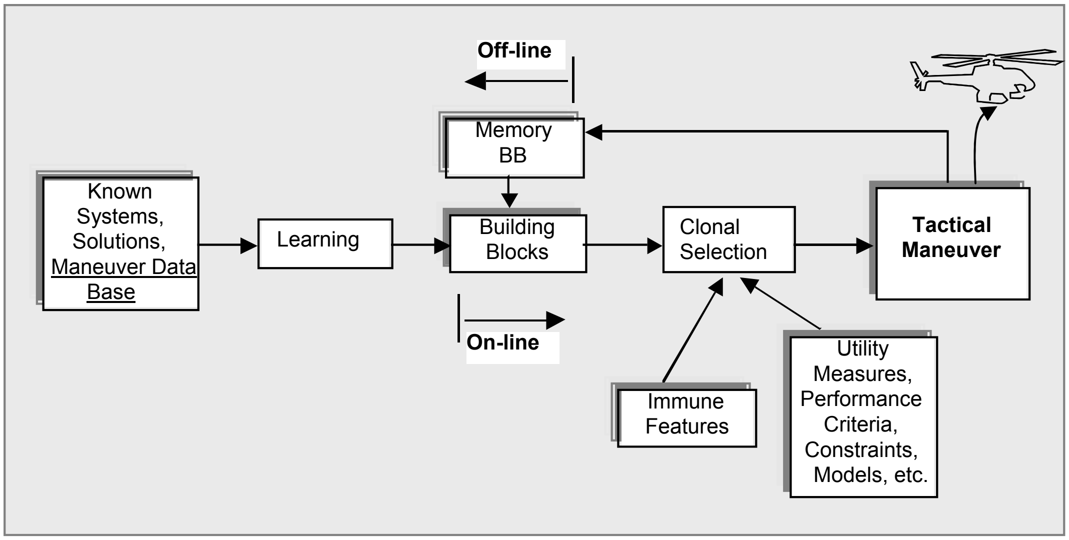
Off-line
Memory BB
Known Systems, Solutions, Maneuver Data Base
Learning
Building Blocks
Clonal Selection
Tactical Maneuver
On-line
Immune Features
Utility Measures, Performance Criteria, Constraints, Models, etc.

Figure 11. Tactical Maneuver Selection using AIS

process?

- What role does the recombination play in alleviating the complexity in the search?
- What problems exhibit a complexity catastrophe phenomenon?

Another area that needs more understanding is the immune system's ability to discriminate between self and non-self. The question in the engineering system context is one of discriminating normal behavior from an abnormal one.

6 References

1. Benjamini, E. (1992). Immunology, A Short Course. Wiley-Liss Publications, New York, NY.
2. Roitt, I., Brostoff, J., and Male, D. (1989). Immunology, C. V. Mosby Company, Washington, D.C.
3. Costa Branco, P.J., Dente, J.A., Martins, J.F., Pires, A.J., (2000) Fault detection using immune-based systems and formal language algorithms, Proceedings of the 39th IEEE Conference on Decision and Control, Volume: 3.
4. Dasgupta, D. (1998). An Anomaly Detection Algorithm Inspired by the Immune System. Chapter 14 in the book entitled Artificial Immune Systems and Their Applications. Publisher: Springer-Verlag, Inc., pp 262-277.
5. Dasgupta, D. and Gonzalez, F. (2002). An Immunity-Based Technique to Characterize Intrusions in Computer Networks, In the Journal of the IEEE Transactions on Evolutionary Computation, Vol. 6, No. 3, June.
6. Dasgupta, D.; Gonzalez, F.; Kozma, R. (2002). *C*ombining negative selection and classification techniques for anomaly detection, Proceedings of the 2002 Congress on Evolutionary Computation, Volume: 1 , 12-17 May.
7. Dong-Wook Lee., Jin-Hyung Jun, Kwee-Bo Sim, (1999) Realization of cooperative strategies and swarm behavior in distributed autonomous robotic systems using artificial immune system, 1999 IEEE International Conference on Systems, Man, and Cybernetics, Volume 6.
8. Forrest, S., Perelson, A. S., Allen, L., and Cherukuri, R. (1994). Self-nonself discrimination in a computer. IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Almitos, CA, PP. 202-212.
9. Harmer, P. K., Williams, P. D. Gunsch, G. H. Lamont, G.B. (2002). An artificial immune system architecture for computer security applications, IEEE Transactions on Evolutionary Computation, Vol. 6, No. 3, June.
10. Hightower, R, Forrest, S. & Perelson, A. S. (1993). The evolution of secondary organization in immune system gene libraries. In Proceedings of the Second European Conference on Artificial Life.
11. Sathyanath, S. & Sahin, F. (2001). An AIS Approach to a Color Image Classification Problem in a Real Time Industrial Application, Proceedings of the 2001 IEEE systems, Man, and Cybernetics Conference.
12. J. Yoo, J., and Hajela, P. (1999). Immune Network Simulations in Multicriterion Design, Structural Optimization, 18, 2/3, pp 85-94.
13. KrishnaKumar, K & J. Neidhoefer (1997). Immunized Neuro-control, Expert Systems with Applications.
14. KrishnaKumar, K. & Neidhoefer, J. (1999), Immunized Adaptive Critics for an Autonomous Aircraft Control Application, In (Ed.) D. Dasgupta, Artificial Immune Systems and Their Applications, Springer-Verlag, pp. 221-241.
15. KrishnaKumar, K. (1992). Immunized Neurocontrol: Concepts and Initial Results", COGANN-92, A Workshop on Combinations of GA and NN. June 6.
16. KrishnaKumar, K. (1993). Immunized Artificial Neural Systems, World Congress on Neural Networks, Portland, July 11-15
17. KrishnaKumar, K. Satyadas, A. & Neidhoefer, J. (1995), 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.
18. Okamoto, T., and Ishida, Y. (2000). A Distributed Approach against Computer Viruses Inspired by the Immune System, IEICE Trans. on Comm., Vol. E83-B, No. 5, May, pp. 908-915, 2000.
19. Opream, M. & Forrest, S. (1999). How the immune system generates diversity: Pathogen space coverage with random and evolved antibody libraries, In proc. of the GECCO' 99, 2 pp1651-1650.
20. Perelson, A. S., (1989). Immune Network Theory, Immunological Review 1989, 110, pp. 5-36.
21. Stewart, J. and Varela, F. J. (1989). Exploring the Meaning of Connectivity in the Immune Network, Immunological Review, 1989, 110, pp. 37-61
22. Perelson, A. S., and Kauffman, S.A. (1991). Molecular Evolution on Rugged Landscapes: Proteins, RNA, and the Immune System, Addison-Wesley Publishing Company.
23. KrishnaKumar, K., Kaneshige, J, Satyadas, A (2002). Challenging Aerospace Problems for Intelligent Systems, Presented at the von Karman Lecture series on Intelligent Systems for Aeronautics, Belgium, May 2002.