Journal of Network and Computer Applications 33 (2010) 633–645

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Agent-based artificial immune system approach for adaptive damage detection in monitoring networks

Bo Chen [a,b,*]

[a] Department of Mechanical Engineering–Engineering Mechanics, Michigan Technological University, Houghton, MI 49931, USA
[b] Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931, USA

ARTICLE INFO

Article history:
Received 12 November 2009
Received in revised form
4 February 2010
Accepted 18 March 2010

Keywords:
Artificial immune systems
Mobile agents
Multi-agent systems
Pattern recognition
Structural health monitoring

ABSTRACT

This paper presents an agent-based artificial immune system approach for adaptive damage detection in distributed monitoring networks. The presented approach establishes a new monitoring paradigm by embodying desirable immune attributes, such as adaptation, immune pattern recognition, and self-organization, into monitoring networks. In the artificial immune system-based paradigm, a group of autonomous mobile monitoring agents mimic immune cells (such as B-cells) in the natural immune system, interact locally with monitoring environment, and respond to emerging problems through simulated immune responses. The presented immune-inspired monitoring paradigm has been applied to structural health monitoring. The "antibody" of a mobile monitoring agent is a pattern recognition algorithm tuned to a certain type of structural damage pattern. The mobile monitoring agent performs damage diagnosis based on structural dynamic response data. Mobile monitoring agents communicate with each other and collaborate with network components based on agent interaction protocols defined in agent standards, the Foundation for Intelligent Physical Agents standards. A mobile agent system embedded in sensor nodes supports the selective generation, migration, communication, and management of mobile monitoring agents automatically. The active structural health monitoring is achieved by distributing mobile monitoring agents to the sites where they are needed. The structural damage diagnosis using mobile monitoring agents and artificial immune pattern recognition method has been tested using a scaled steel bridge structure. The test result shows the feasibility of using this approach for real-time structural damage diagnosis.

1. Introduction

Technology is taking us to a world where numerous networked devices interact with the physical world in multiple ways and at multiple scales, from the global Internet scale down to micro and nano-devices. The distributed sensing and monitoring systems are one type of these systems, which are playing an important role in U.S. economic prosperity, security, and quality of life. Due to ever-increasing complexity of systems and unpredictable working conditions, the distributed sensing and monitoring systems need to possess high quality of adaptability, autonomy, and reliability. The fundamental research challenge is to establish robust decentralized computing systems that interact with physical world, be capable of operating under changing environments, and exhibit the desired response behavior under physical constraints such as communication bandwidth, energy consumption, and processing power.

To address aforementioned challenges, a number of researchers have made a great effort to improve the flexibility, scalability, and intelligence of the networked systems. Agent technology is one of the promising methods to achieve these goals. From the reported literature, agent approach has been applied for the network fault detection (Al-Kasassbeh and Adda, 2009), data fusion and management (Zhu et al., 2007), interoperable network framework (Jabeur et al., 2009), grid computing (Shi et al., 2006), mobile computing test system (Ilarri et al., 2009), and sensor network middleware to provide dynamic programming environments for flexible network management (Boulis et al., 2003; Szumel et al., 2005; Levis and Culler, 2002).

An artificial immune system (AIS) is suitable to handle the great complexity of the reality (Castiglione et al., 2001). The reason behind this is that the natural immune system incorporates a variety of artificial intelligence techniques, such as pattern recognition through a network of collaborating agents, adaptive learning through memory, and an advanced selection mechanism of the best B-cells (Lanaridis et al., 2008). The

* Correspondence at: Department of Mechanical Engineering–Engineering Mechanics, Michigan Technological University, Houghton, MI 49931, USA.
Tel.: +1 906 487 3537.
E-mail address: bochen@mtu.edu

1084-8045/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2010.03.011

desirable characteristics of the immune system have inspired the development of artificial immune systems for various applications. Basic immune-inspired algorithms include negative selection, clonal selection, immune networks, and dendritic cell. The negative selection and dendritic cell algorithms have been applied for anomaly detection, including intrusion detection (Boukerche et al., 2007; Tarakanov, 2008), computer security (Harmer et al., 2002), and misbehavior detection (Sarafijanovic and Le Boudec, 2005; Dasgupta et al., 2005). The clonal selection and immune network algorithms gained applications in the fields of pattern recognition and data clustering (Watkins et al., 2004; Zhong et al., 2007; Chen and Zang, 2009).

This paper presents an immune-inspired monitoring paradigm, which embodies desirable immune attributes, such as adaptation, learning capability, and self-organization, into distributed monitoring systems. In the presented paradigm, mobile monitoring agents mimic immune cells (such as B-cells) in the natural immune system for the anomaly detection and pattern recognition in distributed monitoring systems. Mobile monitoring agents interact locally by sensing and monitoring the environment, and respond to emerging problems through simulated immune responses. The presented bio-inspired monitoring paradigm has been applied to Structural Health Monitoring (SHM) networks. Adaptive structural health monitoring is critical for a quick response to operational and environmental changes. The presented artificial immune system approach provides adaptive monitoring through the evolution of monitoring agents and memory cells. The active structural health monitoring is achieved by distributing mobile monitoring agents to the sites where they are needed. The monitoring task is managed automatically by an embeddable mobile agent system. The major contribution of the presented work lies in providing a systematic approach to address the adaptive monitoring issues with an integrated network framework consisting of agent-based network middleware, damage pattern recognition through mobile monitoring agents, and the embodiment of desirable immune mechanisms.

The rest of the paper is structured as follows. Section 2 discusses desirable characteristics of natural immune system for adaptive monitoring. Section 3 introduces an agent-based network framework to mimic immune functions. Section 4 presents artificial immune pattern recognition method for damage classification. Section 5 illustrates structural damage detection using artificial immune pattern recognition and mobile monitoring agents. Section 6 concludes the presented work.

2. Artificial immune system approach for adaptive monitoring

The natural immune system is an effective defense mechanism for a given host against infections (de Castro, 2006). The immune system protects living organisms from invading antigens through the collaboration of immune cells (such as B-cells and T-cells), as shown in Fig. 1. The surface receptor (antibody) of an immune cell can recognize and bind to antigens. The adaptive immune response is initiated by an encounter between a B- or T-cell and its specific antigen. The adaptive immune response achieves two goals: the number of cells that are capable of responding to a particular antigen are multiplied (clonal expansion), and these new generated immune cells are able to produce a large number of antibodies for binding to the intruder (Delves et al., 2006). When a B-cell encounters a nonself antigen that has sufficient affinity with its receptors, the antibody of the B-cell binds to the antigen, marking it for destruction. To avoid lymphocyte activation to 'self' substances, the activation of a B-cell requires the co-stimulation by a helper T-cell (TH). Helper T-cells can

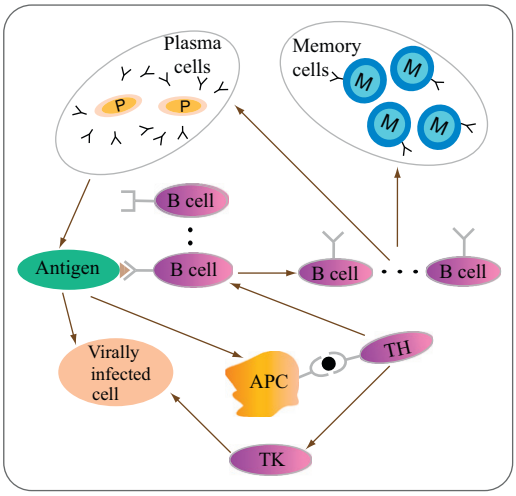

Fig. 1. The adaptive immune response TH: helper T-cell, TK: killer T-cell (Chen, 2009).

recognize antigen in antigen presenting cells. Once receiving a stimulation signal from a helper T-cell, the B-cell is activated. It, therefore, undergoes a clonal selection process in which the B-cell proliferates and differentiates into antibody secreting cells (plasma cells) and memory cells. The memory cells will remain in the immune system for months or years. The first exposure of a B-cell to a specific type of antigen triggers the primary response in which the pattern is recognized and the memory is developed (Castiglione et al., 2001). The secondary response occurs when the same antigen is encountered again. The memory cell for a specific antigen that had stimulated the primary response will respond to a previously recognized antigen in a much shorter time compared to a newly activated B-cell (Carter, 2000). Activated antigen-specific killer T-cells (TK) kill virally infected cells either by lysing them or through the secretion of soluble mediators which act to inhibit viral replication.

Due to the similarities of the human immune system and distributed monitoring systems, the immune system model could be used as the basis for adaptive monitoring (Chen, 2009). This approach is well suited to address monitoring problems because: (1) The AIS-based systems are *autonomous*. The AIS-based systems can automatically manage monitoring tasks by dynamically generating and distributing mobile monitoring agents; (2) The AIS-based systems are adaptive. The adaptive mechanism of the immune system has great value in monitoring networks. The selective generation of mobile monitoring agents based on immune clonal selection is essential for producing large enough amount of specialized mobile monitoring agents in resource-constrained sensor networks (Negoita, 2005); (3) The AIS-based systems are active. The concept of actively dispatching mobile monitoring agents (mimicking B-cells) helps the distribution of specialized monitoring agents to the sites where they are needed. The on-site diagnosis overcomes the delay of system response. The distributed data fusion reduces the raw data transmission, which will save the communication bandwidth in wireless monitoring networks.

To achieve AIS-based monitoring, the desirable immune attributes and mechanisms need to be embodied into distributed monitoring systems to provide properties analogous to those presented in the natural immune system. The idea of the embodiment of immunology into engineered systems is shown in Fig. 2, which shows the direct coupling between the computational system and its environment, and is a rich complex feedback process (Stepney, 2007). The key characteristics of the immunology are modeled in mathematical and computational

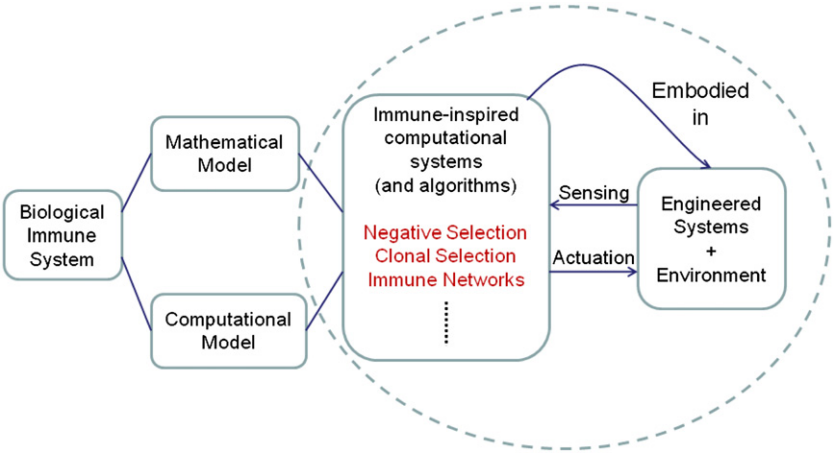

Fig. 2. Embodiment of immunology into engineered systems.

models through biological observations and experiments. The immune-inspired computational systems (and algorithms) based on these models can be embodied into the engineered systems. To achieve this embodiment, both computational system and the engineered system should be open, have sufficient dynamics to be modified by the other party, and there is high enough interaction bandwidth between two parties. For the monitoring systems, the computational system dynamics include the modification of monitoring agent population and memory cells: number, type, and location. The engineered system and the environment are open to the computational system. The monitoring agents can read real-time sensor data and make decisions accordingly. The interaction between the computational system and the engineered system occurs when immune algorithms recognize a damaged pattern or detect a fault. The computational system can also modify the engineered systems, for example, fault isolation and damage alert.

3. Agent-based network framework to enable immune functions in structural health monitoring networks

The presented AIS-based structural health monitoring network framework is shown in Fig. 3. A group of mobile software agents (mimicking immune cells, such as B-cells) equipped with damage pattern recognition algorithms monitors the health of a structure by patrolling over a sensor network deployed on the structure. The "antibody" of a mobile agent is a pattern recognition algorithm tuned to a certain type of structural damage pattern. A mobile monitoring agent reads real-time structural data from sensors and performs damage diagnosis using an equipped pattern recognition algorithm. Mobile agents can communicate with each other for group decision making or collaborate with other network components. A mobile agent system embedded in sensor nodes supports the generation, migration, communication, and management of mobile monitoring agents automatically, with no human interventions are involved. When the antibody of a mobile monitoring agent recognizes a damage pattern at a sensor node, the mobile agent communicates with a stationary agent in the knowledge base. If the damage is confirmed, the mobile agent is activated. Its antibody will be cloned and mutated through a clonal selection process. In addition, an alert agent is generated to inform remote operators. At the same time, mobile monitoring agents carrying cloned antibodies migrate to the locations close to the damage site, where they conduct a careful damage diagnosis. A diagnosis report is sent to the human operators.

In the following subsections, the components to establish an artificial immune system-based sensor network framework, adaptive agent population management, and agent interaction to mimic immune functions are introduced.

3.1. Basic components of AIS-based monitoring networks

To mimic immune functions, basic components of an AIS-based monitoring network and the mapping of these components with the natural immune system are listed in Table 1. Mobile monitoring agents mimic B-cells in the human body. Pattern recognition algorithms are antibodies of monitoring agents. Feature vectors are used to represent antigenic patterns. A monitoring agent activated by a specific damage pattern will be cloned to increase the number of special skilled monitoring agents in the network. The reproduction will be performed with mutation to increase the diversity of monitoring agents. The damage detection is performed through mutual and dynamic interaction among agents and the monitoring network. The building blocks to realize the AIS-based monitoring networks include pattern recognition algorithms and feature representations for each pattern, coordination protocols among monitoring agents and network components, and adaptive agent population management.

3.2. Adaptive agent population management

Adaptive agent population management is based on immune clonal selection principle for controlling the amount and type of mobile monitoring agents in a network. The clonal expansion, immune memory, and programmed cell death rates allow the immune system to dynamically allocate resources as needed in a distributed environment (Zhong et al., 2006). In the natural immune system, cells capable of recognizing an antigenic stimulus will proliferate and differentiate into effecter cells (de Castro and Timmis, 2002). This adaptive resource management mechanism is valuable for monitoring sensor networks. The selective generation of mobile monitoring agents is essential for producing large enough amount of specialized mobile monitoring agents in resource-constrained sensor networks (Negoita, 2005).

In the presented system, the agent population control mechanism is shown in Fig. 4. The adaptive management is accomplished by the collaboration of monitoring agents, knowledge base, and network components responsible for clonal selection. If a monitoring agent recognizes its corresponding damage pattern, coupled with the confirmation signal from the knowledge base, it is

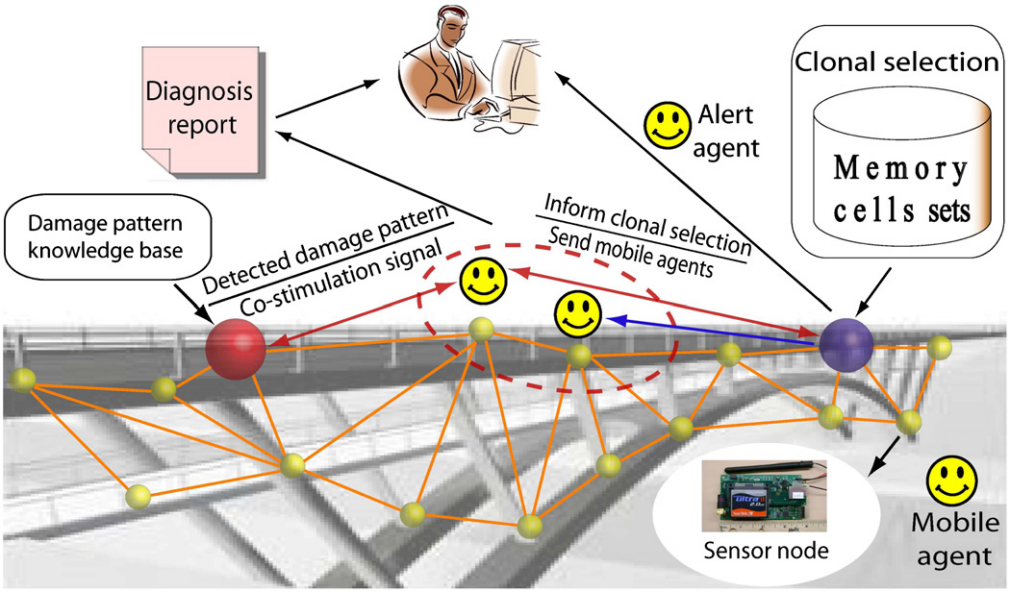

Fig. 3. AIS-based SHM sensor networks (Chen, 2009).

Table 1
Mapping between the immune system and the AIS-based monitoring networks.

Immune system	AIS-based monitoring networks
• B-cells	• Mobile monitoring agents
• T-cells	• Knowledge base
• Antibodies	• Pattern recognition algorithms
• Antigens	• Data pattern feature vectors
• Clonal selection	• Clonal selection algorithm
• Immune memory	• Memory cell sets
• Self/nonself discrimination	• Negative selection algorithm
• Immune network theory	• Immune network computational models

selected to be cloned. The cloned monitoring agents are divided into two groups. One group is effecter monitoring agents who are dispatched to the sensor nodes close to the location where the damage is detected for further diagnosis. The other group is memory cells. Agent life is defined by a parameter, lifetime. After creating a mobile monitoring agent, it is assigned to a specific lifetime value to control the death of the agent. For memory cells, it will be assigned a longer lifetime, so it can circulate over the network for a relatively longer time period. The monitoring agents that are not activated within its lifetime will die to allow monitoring networks generating other types of monitoring agents for exploring a diverse range of damages.

3.3. *Mimic immune functions through agent interaction*

Agent technology is promising in performing complex tasks since it facilitates collaboration among a group of agents and the service providers in the networks. Agent collaboration is achieved through agent communication. Certain message sequences involved in agent conversation often fall into typical patterns. These typical patterns of message exchange are called interaction protocols. To promote interoperability, agent interaction protocols in the presented AIS-based monitoring networks are designed based on the Foundation for Intelligent Physical Agents (FIPA) agent interaction protocols. This section illustrates several key scenarios involved in mimicking artificial immune functions in monitoring networks and the required interaction of the agents in each of these scenarios.

Fig. 5 shows the interaction protocol of activating a mobile monitoring agent. The mobile agent is an initiator; and the knowledge based agent and clonal selection agent are participants in this interaction protocol. A knowledge base is used to keep feature vectors of self and nonself. It also maintains historical data for regenerating pattern representatives. A stationary agent resides in the knowledge base to communicate with mobile monitoring agents in the network and handle requested services. The clonal selection agent performs selective generation of monitoring agents based on the clonal selection algorithm inspired by the immune system. When a monitoring agent detects a certain type of damage in a sensor node, following sequence of agent interaction occurs to activate the monitoring agent. (1) The monitoring agent requests the confirmation of damage by sending a "request" message to the knowledge based agent. The content of the message includes a description of the requested action and the damage feature vectors calculated by sensor data. (2) The knowledge base agent processes the request and makes a decision whether to accept or refuse the request. If an agree decision is made, the knowledge base agent sends an agree message to the monitoring agent, otherwise, a refuse message. (3) When the knowledge base agent successfully completes the request, it will notify the monitoring agent of the results. (4) The knowledge base agent will also send a "request when" message to the clonal selection agent to confirm the damage detected by the monitoring agent. (5) When the damage is confirmed by the knowledge base agent, the monitoring agent requests cloning it by sending a "request" message to the clonal selection agent. (6) If the clonal selection agent agrees to clone the monitoring agent, it will send an agree message to the monitoring agent (7) and the knowledge base agent. (8) When the clonal selection agent completes the agent clone, it will also send an inform message to the monitoring agent (9) and the knowledge base agent.

Fig. 6 illustrates the interaction protocol for a group of monitoring agents performing damage diagnosis. This interaction protocol is based on the FIPA recruiting interaction protocol. The initiator is the mobile monitoring agent who detects damage, while the knowledge base agent acts as the recruiter. When the monitoring agent is informed by the clonal selection agent about the completion of agent clone, it will organize a mediated group

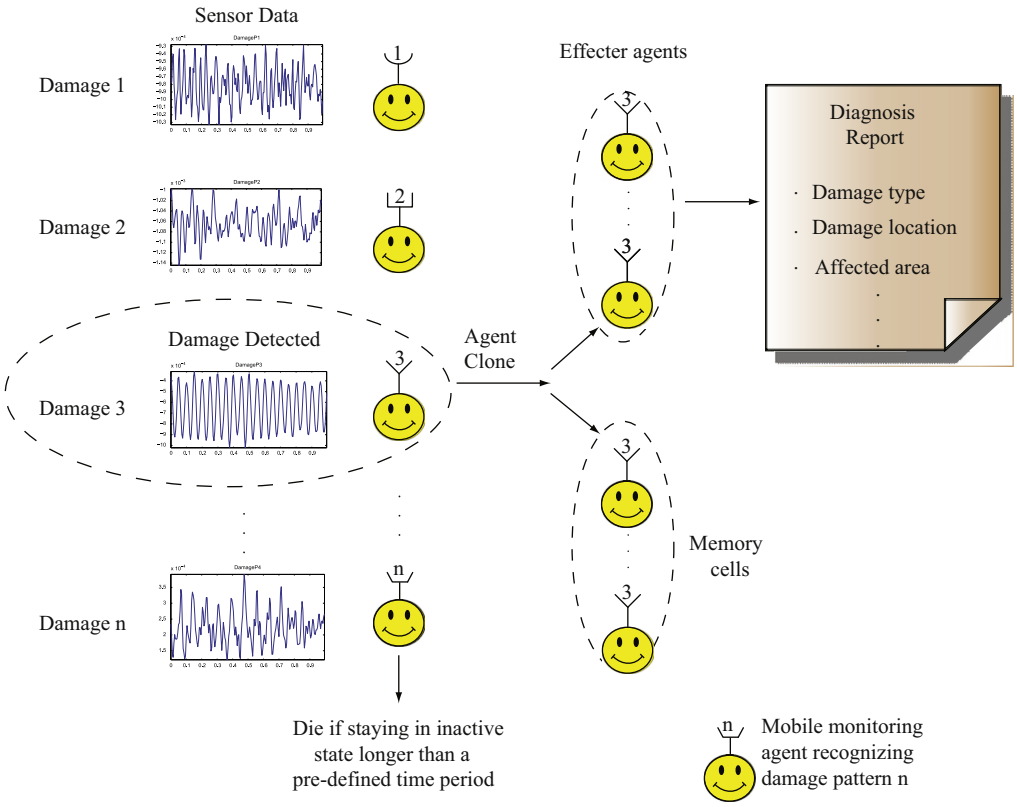

Fig. 4. Adaptive agent population management.

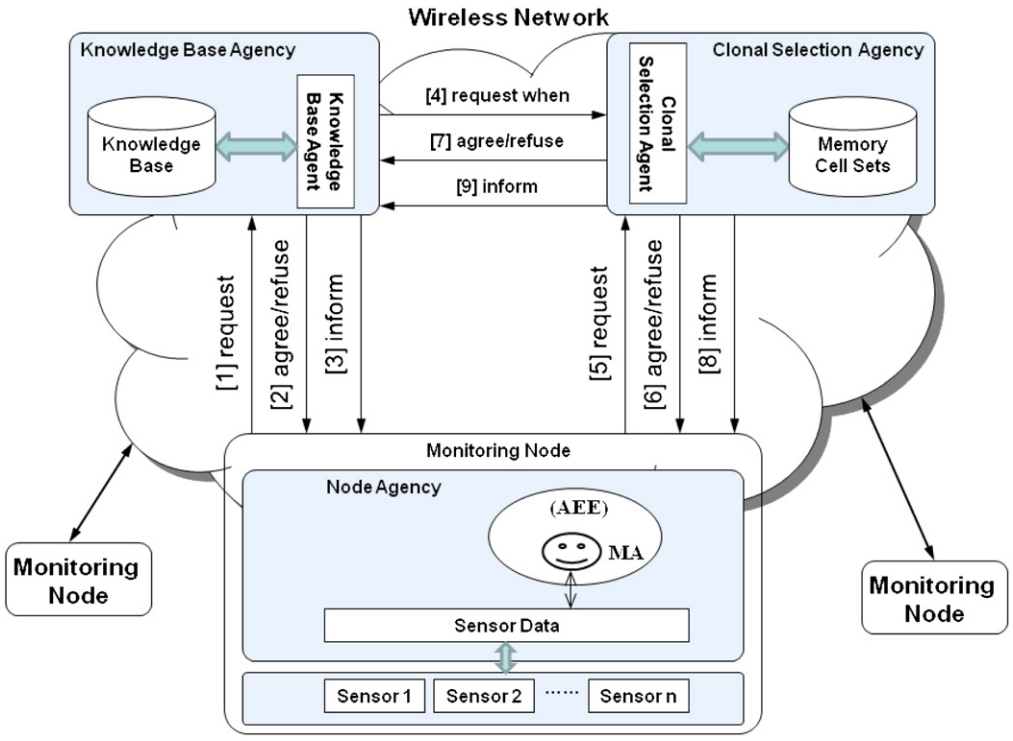

Fig. 5. The activation protocol of a mobile monitoring agent MA: mobile agent, AEE: agent execution engine.

damage diagnosis. The monitoring agent begins recruiting interaction by sending a proxy message to the knowledge based agent. The proxy message contains: a referential expression denoting the target agents to which the recruiter should forward the communicative act, the communicative act to forward and a set of proxy conditions such as the maximum number of agents to be forwarded. The knowledge base agent processes the request and makes a decision whether to agree to or refuse the request, and

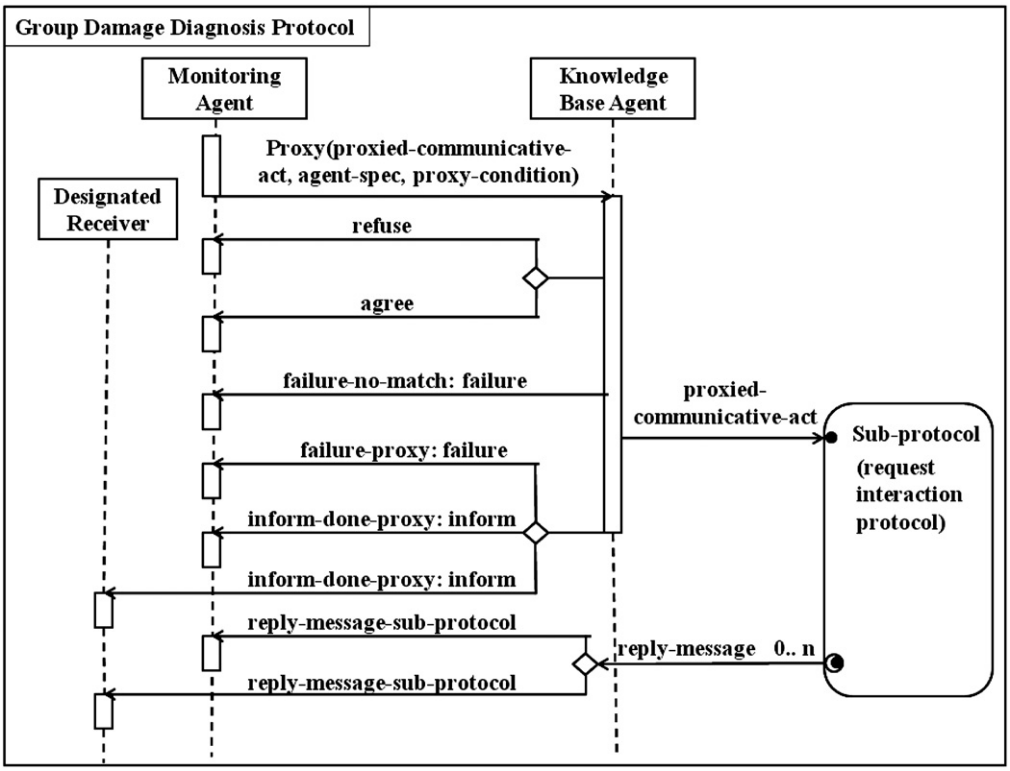

Fig. 6. The interaction protocol of group damage diagnosis.

communicates either agree or refuse communicative act accordingly. Once the recruiter has agreed to be a proxy, it locates agents as per the description from the proxy message. If no such agents can be found, the recruiter returns a failure-no-match and the interaction terminates. Otherwise, the recruiter interacts with the matching agents with sub-protocol embedded in the proxy message, request damage diagnosis. The diagnosis results from recruited agents can either be forwarded to the original requester or to a designated receiver. If the recruiter has been given a separate designated receiver from the initiator, the replies of the sub-protocol will be sent to the designated receiver, otherwise to the initiator.

4. Artificial immune pattern recognition

One of the most appealing characteristics of the immune system is its immune cells (such as B-cells) carrying surface receptors that are able to recognize and bind antigens. This section introduces Structural Damage Classification (SDC) based on Artificial Immune Pattern Recognition (AIPR) approach (Chen and Zang, 2009). The classification algorithm is designed using concepts derived from the natural immune system. The immune pattern recognition method achieves pattern recognition by establishing memory cell sets; each set is responsible for recognizing one type of pattern. Memory cells are feature vectors of recognizing data pattern. The AIPR-SDC algorithm consists of two major stages. The first stage is the feature extraction from structural dynamic response data. In this stage, all the training data (sensor data) need to be standardized and the feature vectors need to be generated. In addition, memory cell set and antibody set for all the patterns are initialized. In the second stage, training antigens stimulate the antibody set and thus causes some of antibodies to produce clones. The cloned antibodies are mutated to increase the diversity of the antibody set. The antibody having the highest affinity with the stimulating antigen is chosen as a candidate memory cell for updating memory cell set.

4.1. Feature extraction

The structural damage patterns are represented by feature vectors extracted from the dynamic response data of a structure. The feature vector of a time series is formed by coefficients of an autoregressive (AR) model of the time series. To reduce environmental effects, the measurement data Z are standardized by $y_{ij}=z_{ij}-\mu_i/\sigma_i \quad j=1,2,\ldots,n$, where μ_i and σ_i are the mean and standard deviation of the time series $\bar{z}_i$. To extract feature vectors for a local area, time series measurement data sets from multiple sensors are reduced to lower dimensions by the principal component analysis method. The compressed time series x is then fitted to an AR model of order p as shown in Eq. (1):

$$x_k=\sum_{i=1}^{p}\alpha_i x_{k-i}+r_k \quad k=p+1,\cdots\cdots,n \tag{1}$$

where α_i, $i=1,2,...,p$ are the coefficients of the AR model. The vector $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_p)^T$, a collection of the AR coefficients, is selected as the feature vector of the measurement data Z. The effectiveness of the AR-model-based feature vectors is tested using experimental data of a benchmark structure proposed by the American Society of Civil Engineers (ASCE). The acceleration signals of the five data patterns of the benchmark structure are shown in Fig. 7. The feature vectors of the corresponding data patterns projected to the first two principle components are shown in Fig. 8. The memory cells generated by the AIPR method for these five data patterns are shown in Fig. 9. The detailed introduction of the memory cell generation is given in the following section.

4.2. Memory cell generation

The process of the memory cell generation includes two sub-processes: the evolution of the antibody set and the update of the memory cell set. The flow chart of the memory cell generation is

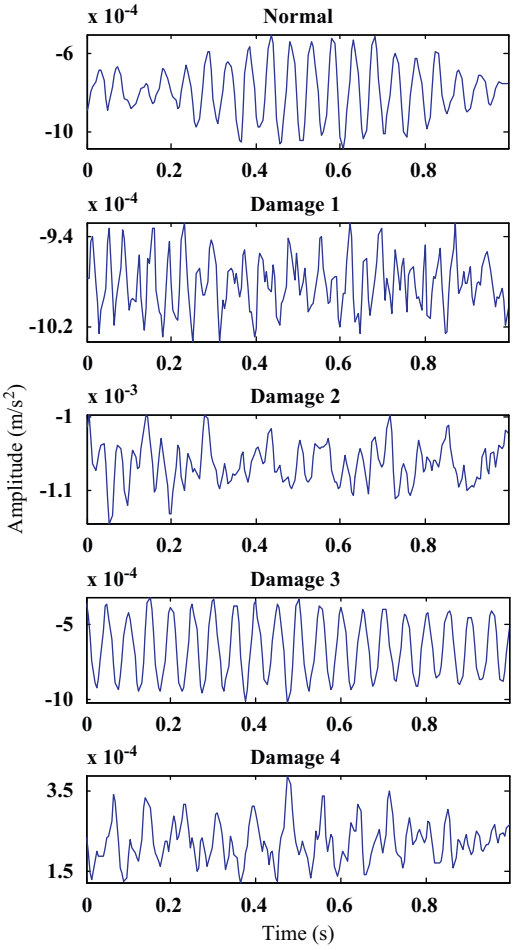

Fig. 7. Acceleration signals of the ASCE benchmark structure.

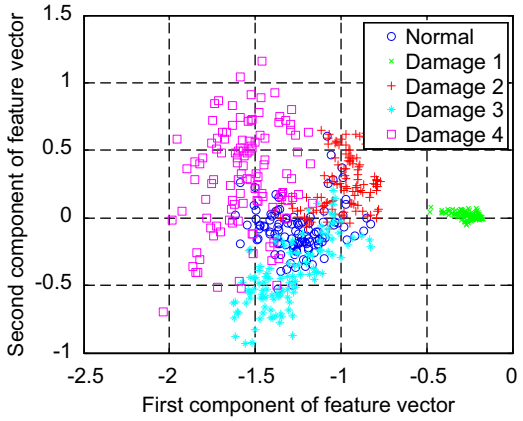

Fig. 8. Feature vectors of the experimental data of the ASCE structure.

shown in Fig. 10. The training feature vectors are used to stimulate this process.

4.2.1. Evolution of the antibody set using antigenic stimulation

The initial antibody set is generated by the random selection of antibodies from the training data. The stimulation of the antibody set by an invading antigen (a training data) will cause the evolution of the antibody set. The description of the antibody set evolution algorithm is given in Fig. 11. For a training antigen *ag*, the affinity between the antigen and each antibody *ab* that is in

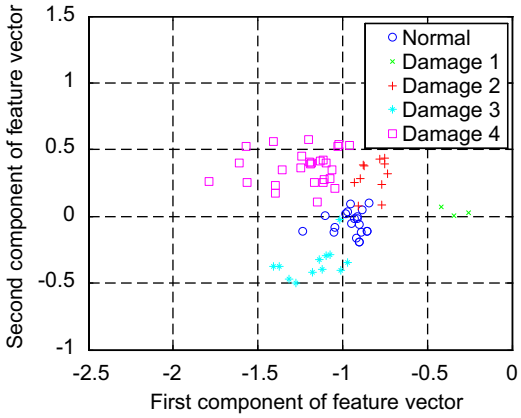

Fig. 9. Memory cells for the five data patterns of the ASCE structure.

the same class (pattern) as the antigen is calculated. Let $ab.f=\beta=(\beta_1,\beta_2,\cdots,\beta_p)^T$ and $ag.f=\gamma=(\gamma_1,\gamma_2,\cdots,\gamma_p)^T$ denote the feature vectors of an antibody *ab* and the antigen *ag*, respectively. The affinity between an antibody and the antigen is defined as

$$aff(ab,ag)=1-\frac{1}{2}dist(\beta,\gamma) \tag{2}$$

where $dist(\beta,\gamma)$ is the Euclidian distance between the feature vectors of β and γ. The probability that an antibody *ab* is cloned depends on its affinity with the antigen. The number of the cloned antibodies, *CloneNumber*, depends on the Clonal Rate (CR) and the Clonal Value (CV). The CR is an integer value used to control the number of antibody clones allowed for the activated B-cell. The CV is a value that measures the response of a B-cell to an antigen. According to the natural immune system, the higher the affinity, the larger the number of antibodies is cloned. We choose the clonal value being equal to the affinity value. The *CloneNumber* is then calculated using

$$CloneNumber=round(CR\times CV)=round(CR\times aff(ab,ag)) \tag{3}$$

where $round(\cdot)$ is an operator that rounds its value to the closest integer. The cloned antibodies undergo a maturation process that increases the diversity of the antibody set. The mutation is performed by mutating the feature vectors of the cloned antibodies as shown in Eq. (4)

$$ab_{mutated}\cdot f=ab\cdot f+MV\times\phi \tag{4}$$

where $ab_{mutated}$ is the mutated antibody and *MV* is the Mutation Value (MV). Typically, the higher the affinity is, the smaller the mutation value. In our design, the mutation value MV is defined as $MV=1-CV$. In Eq. (4), the vector $\phi=(\phi_1,\phi_2,\cdots,\phi_p)^T$ is a random vector whose dimension is the same as that of the feature vector. Each element ϕ_i is defined by $\phi_i\sim N(0,\sigma^2)$, where $N(0,\sigma^2)$ is a normal random variable with the standard deviation of σ.

The mutated antibodies are added into the antibody subset to which the *ag* belongs. The resulting antibody subset is sorted in a descending order according to the affinity values of the antibodies with the given antigen. The top *MaxABN* number of antibodies is selected to form the evolved antibody set. The rest of antibodies are discarded. The antibody with the highest affinity is chosen as the candidate memory cell $MC_{candidate}$ for the updating of memory cell set.

4.2.2. Update memory cell set

The candidate memory cell generated in the antibody set evolution process is used to update the memory cell set to enhance the representative quality of memory cells for each pattern. The description of the memory cell set update algorithm is given in Fig. 12. The memory cell update occurs in the following

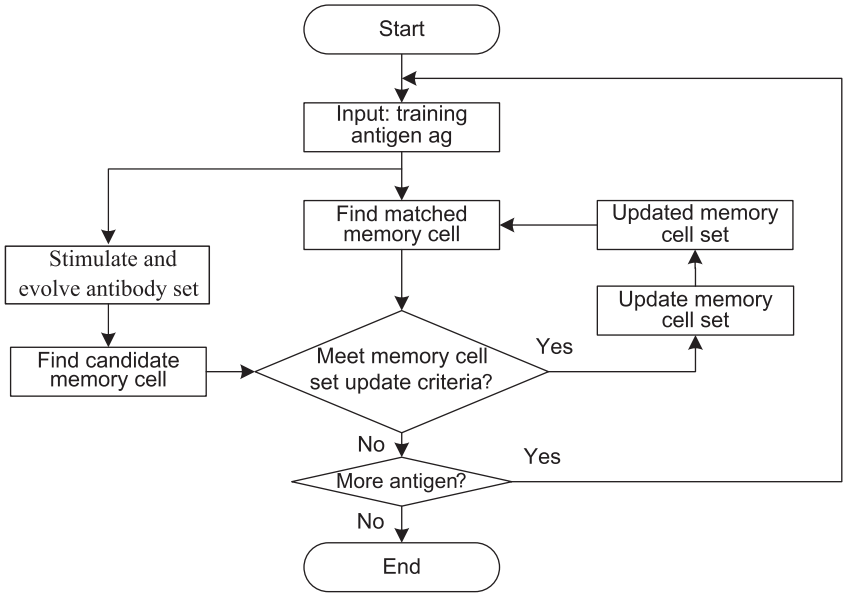

Fig. 10. Memory cell generation.

Begin
Input an antigen *ag*;
While (there is more antibody *ab* which is in the same class as *ag*) do
Clone antibody *ab* based on the affinity with the *ag*;
Mutate the cloned antibodies;
Keep the mutated antibodies staying within the unit hyper-sphere;
Form a new antibody set using top *MaxABN* number of antibodies;
End while
Select the highest affinity antibody as the candidate memory cell;
End

Fig. 11. The description of the antibody set evolution algorithm.

scenarios. First, when the root mean square distance, rms, between the candidate memory cell and the memory cells in the same class is greater than a specified threshold value Memory Cell Injection Threshold (MCIT), the candidate memory cell is injected into this class of memory cells. Let ag.c denote the class label of the antigen ag, let $MCS_{ag.c}$ denote the memory cell subset with the same class as the given antigen ag, and let $|MCS_{ag.c}|$ denote the total number of the memory cells in the subset $MCS_{ag.c}$. The rms is defined by

$$rms = RMS(dist_1, dist_2, \ldots dist_{|MCS_{ag.c}|}) = \frac{1}{\sqrt{|MCS_{ag.c}|}} \sqrt{\sum_{i=1}^{|MCS_{ag.c}|} dist_i^2} \quad (5)$$

where $dist_i = dist(mc_i, MC_{candidate})$, $mc_i \in MCS_{ag.c}$, and $i = 1,2,\ldots, |MCS_{ag.c}|$. If the *rms* is greater than the MCIT, the candidate memory cell is added into the memory cell subset $MCS_{ag.c}$.

In the second case (the *rms* is less than or equal to MCIT), the candidate memory cell compares with the matched memory cell. The matched memory cell is the memory cell that has the highest affinity with the given antigen in the same class, which is denoted by $MC_{matched}$. When the affinity between $MC_{candidate}$ and the given antigen *ag* is greater than the affinity between $MC_{matched}$ and antigen *ag*, the candidate memory cell replaces the matched memory cell if the affinity between $MC_{candidate}$ and $MC_{matched}$ is greater than the Memory Cell Replacement Threshold (MCRT), otherwise the candidate memory cell is added into the memory cell subset $MCS_{ag.c}$. The memory cells for the five data patterns of the benchmark structure are shown in Fig. 9. For classifying a

Begin
Input antigen *ag*;
Find the matched memory cell;
Calculate the root mean square *rms* for the candidate memory cell;
If *rms* > MCIT
Add the candidate memory cell into the memory cell set;
Else if ((*aff(MCcandidate, ag)*>*aff(MCmatched, ag*)) and
(*aff(MCcandidate, MCmatched)*> MCRT)
Replace the matched memory cell by the candidate memory cell;
Else if (*aff(MCcandidate, ag)*>*aff(MCmatched, ag*)
Add the candidate memory cell into the memory cell set
End if
End

Fig. 12. The description of the memory cell set update algorithm.

damage-pattern-unknown time series, the affinities between the feature vector of the time series and memory cells are calculated. The pattern of the time series is classified to the same pattern as the memory cell with which the time series has highest affinity.

5. Structural damage detection using AIPR method and mobile monitoring agents

To validate the presented AIPR approach using mobile agents, a mobile-agent-based middleware has been embedded into a high computational power sensor unit as discussed below.

5.1. High computational power sensor nodes

To achieve distributed damage diagnosis, sensor nodes are designed to possess high computational capabilities (Chen and Tomizuka, 2008). The high computational power sensor node consists of three boards as shown in Fig. 13. A sensor board sits at the bottom, a finger size embedded computer called Gumstix in the middle, and a WiFi transmission board at the top. The volume of the sensor node is about 4 in × 2.4 in × 0.65 in. The sensor board was designed and fabricated by our research group to meet the structural health monitoring sensing requirement. We employ multi-modal sensing approach and incorporate active sensing

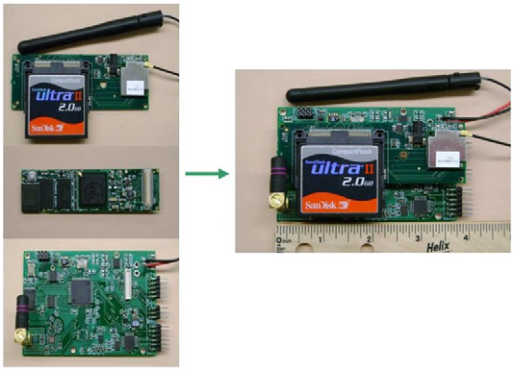

Fig. 13. A high computational power sensor node.

with passive sensing to achieve a better monitoring result. The sensor board connects to a number of sensors, including accelerometers, strain gauges, humidity and temperature sensors, and piezoelectric transducers.

A numerical library module is implemented in the Gumstix-based sensor nodes. The numerical library module provides computational building blocks to construct SHM analysis methods. It contains a C version of Linear Algebra PACKage (LAPACK) library and a utility function library. The C version of LAPACK library provides routines for solving systems of linear equations, linear least-squares problems, eigenvalue problems, and singular value problems (Anderson et al., 1999). It also handles matrix factorizations and estimates condition numbers, such as Cholesky decomposition, singular value decomposition, and Schur decomposition. The utility functions in the utility function library are designed to perform a certain subtask of SHM analysis or common computation that is not available in LAPACK library, for example, Fast Fourier Transform. The existing open source numeric libraries such as Numerical Recipes in C (Press et al., 1992) and the GNU Scientific Library can be used to implement these utility functions. The Numerical Recipes in C and the GNU Scientific Library provide a wide range of mathematical routines, such as random number generators, special functions and least-squares fitting, Eigensystems, Fast Fourier Transforms, and Statistics. Using these legacy programs promotes software reuse and will significantly reduce the development effort for SHM. The numerical accuracy of these open source libraries is comparable with commercial software packages, such as Matlab.

5.2. Mobile-agent-based sensor network middleware

A mobile agent is a software agent that is capable of migrating from one host to another in a network and resumes the execution in the new host. The migration and execution of mobile agents are supported by a mobile agent system. In previous studies, Chen et al. (2006) have developed a mobile agent system called Mobile-C. To build an AIS-based sensor network, Mobile-C has been integrated with developed sensor nodes as shown in Fig. 14. Mobile-C in sensor nodes can host both stationary agents and mobile agents. Stationary agents are those staying in the sensor nodes where they are created, such as data acquisition agent, knowledge base agent, and clonal selection agent. Mobile agents are those created during the system operation for monitoring purpose.

In a mobile-agent-based sensor network, a remote user can dispatch mobile agents to sensor nodes in the network. Different types of mobile agents could be created and dispatched to sensor nodes as needed. For example, the remote operators could dispatch mobile alert agents to sensor nodes for monitoring

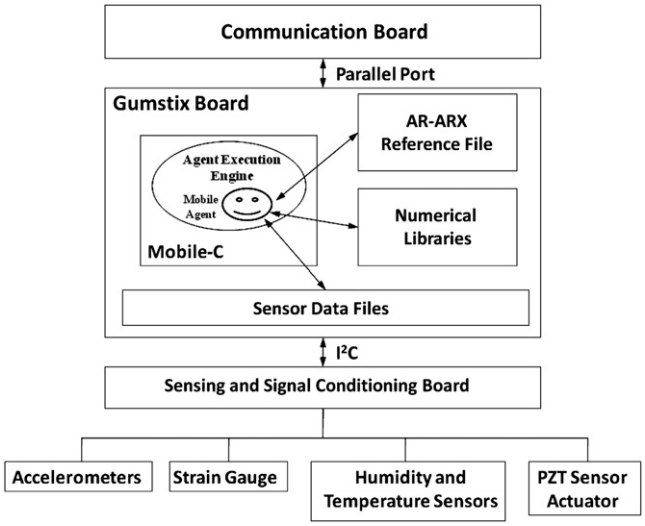

Fig. 14. Mobile-C integrated into sensor nodes.

specified events. Data analysis and damage diagnosis mobile agents with certain expertise (equipped with different data analysis and damage diagnosis algorithms) can roam over the network to perform distributed monitoring tasks. Mobile agents carrying code and execution states move from one sensor node to another, read sensing data from sensors, perform damage diagnosis on the sensor nodes where they reside, and send diagnosis results back to the remote operators. Each agent has its own identification number that is assigned to the agent when it is created. This number will accompany with the agent for the entire life of the agent. Agent migration is achieved through message passing. When a mobile agent is dispatched, information related to the agent such as agent ID, agent itinerary, tasks to be performed, and agent code for each task, is encapsulated into the mobile agent message. The intermediate results from each task will be added into the mobile agent message when the agent travels. Finally, the mobile agent will send all the results back to the dispatcher.

5.3. Scaled steel bridge test

AIPR-based damage pattern recognition was tested on a scaled steel bridge as shown in Fig. 15. During test, the bridge was excited by a shake in vertical direction. The excitation signals of the shaker were generated by Siglab and virtual instruments. Siglab system is integrated with Matlab. Virtual instruments running in the Matlab include classes of network analyzer, function generator, spectrum analyzer, and oscilloscope. For the bridge test, we used the function generator to generate excitation signals for the shaker and network analyzer to measure the signals from a force sensor that was attached to the shaker. The generated shaker excitation signals were amplified by a power amplifier.

The accelerometers were mounted on the beams along the bridge. The excitation signals applied to the shaker were sine waves with a frequency of 10 Hz. Two peak-to-peak voltages of 0.275 and 0.295 V were applied to the shaker to simulate different levels of acceleration. Structural damage was simulated by removing one cross member at the center of the bridge. The snapshots of normal and damaged acceleration signals are illustrated in Fig. 16. The amplitude of the acceleration signals of the damaged pattern is larger comparing to the normal acceleration signals. To generate memory cells for the normal and damaged pattern, acceleration signals for both patterns were

recorded. The sampling rate of accelerometers was 125 sps. Each acceleration data file contains 7500 data points. Twelve data files for each pattern at different excitation levels were used to generate training feature vectors. In each data file, five 2000-point time series were formed starting from the 1000th data point and advancing 1000 points each time. These training feature vectors were used to generate the memory cells for normal and damaged patterns using the AIPR method described in Section 4. The generated memory cells are shown in Fig. 17.

For detecting structural damage in remote sensor nodes, a user sends mobile monitoring agent 1 from a laptop to sensor node 1. The task of the mobile monitoring agent is to diagnose structural damage based on the acceleration data collected by the sensor node. The mobile agent message represented in Extensible Markup Language (XML) format is shown in Fig. 18. A mobile agent message contains general information of a mobile agent and tasks that the mobile agent is going to perform on destination hosts. The general information of a mobile agent includes agent name, agent owner, and the home of the agent. Task information includes number of tasks, a task progress pointer, and the definition for each task such as the hosts to perform tasks, return variables, and the agent code for each task.

Mobile agent code is a regular C program. In this example, mobile monitoring agent 1 reads the acceleration data on the sensor node 1 and builds an AR model for the sensor data. Based on the AR coefficients, the feature vector of the sensor data is formed, and the Euclidean distances from this feature vector to the memory cells are calculated. The k-Nearest Neighbor (kNN) algorithm is used to detect if damage is presented in the structure. For a given feature vector x, the nearest neighbor rule is summarized as follows (Theodoridis and Koutroumbas, 2008). (1) Out of the N training vectors, identify the k nearest neighbors to the vector x. The number of k is general not to be a multiple of the number of classes M. (2) Out of these k samples,

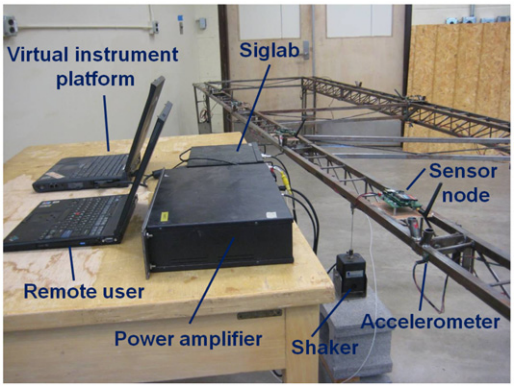

Fig. 15. A scaled steel bridge structure.

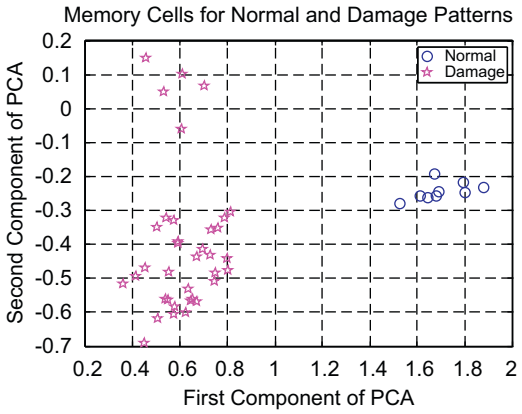

Fig. 17. Memory cells for normal and damaged patterns at 0.295 V excitation level.

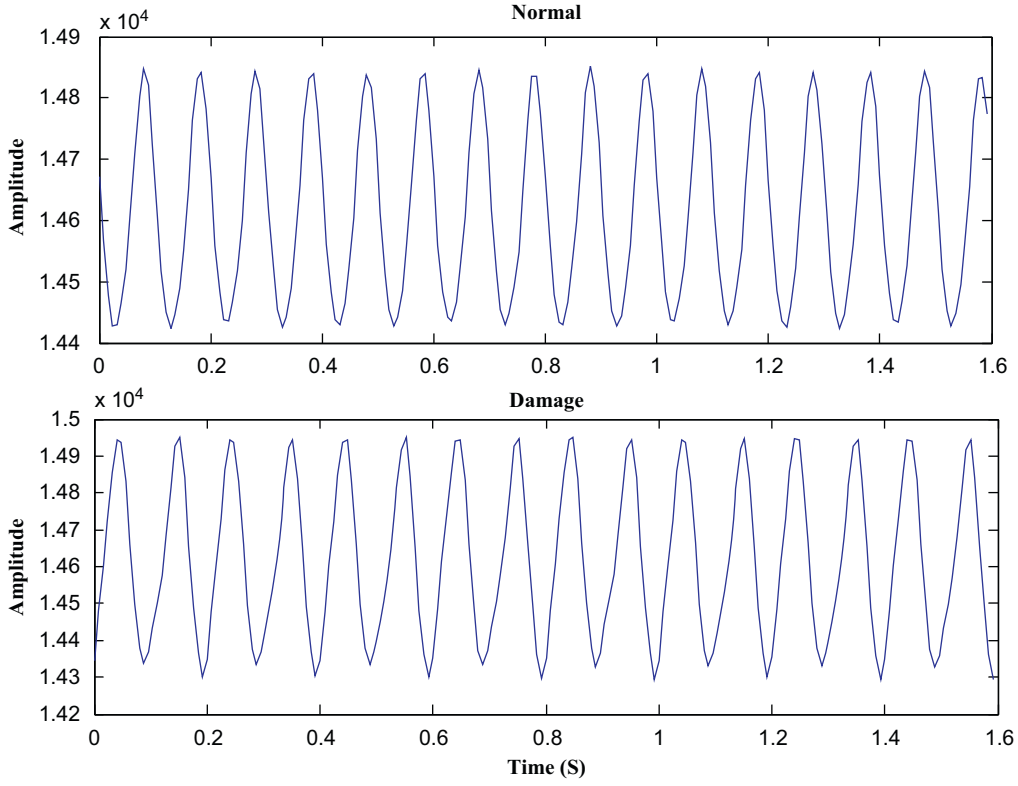

Fig. 16. Acceleration signals for normal and damaged patterns.

```
<!DOCTYPE myMessage SYSTEM "mobilec.dtd">

<MOBILEC_MESSAGE>
  <MESSAGE message="MOBILE_AGENT">
    <MOBILE_AGENT>
      <AGENT_DATA>
        <NAME>mobagent1</NAME>
        <OWNER>IMES</OWNER>
        <HOME>192.168.1.100:5050</HOME>
        <TASKS task="1" num="0">
          <TASK num="0" return="damaged" complete="0" server="gumstix2:5050"
           code_id="1" />
          <AGENT_CODE id="1">
            <![CDATA[
              // mobile monitoring agent code (regular C program)
              #include <stdio.h>
              ……
              int main() {
                // Mobile monitoring agents carry memory cells in arrays
                // Read sensor data from data files
                // Generate an AR model from the sensor data time series
                // Form sensor feature vector using coefficients of the AR model
                // Find Euclidean distances from the sensor feature vector to memory cells
                // Detect damage using k nearest neighbor algorithm
              }
            ]]>
          </AGENT_CODE>
        </TASKS>
      </AGENT_DATA>
    </MOBILE_AGENT>
  </MESSAGE>
</MOBILEC_MESSAGE>
```

Fig. 18. Mobile monitoring agent message represented in XML format.

identify the number of vectors, k_i, that belong to class ω_i, $i = 1,2,\cdots\cdots,M, \sum_i k_i = k$. (3) Assign x to the class ω_i with the maximum number k_i of samples. In our two classes example, the value of M is 2 and the number of k is chosen to be 9. Among 9 smallest Euclidean distances, if the majority (>4) of them are distances from the sensor feature vector to the damaged memory cells, the sensor feature vector is classified to the damaged pattern. As a result, structural damage is detected. Otherwise, the structure is in the normal pattern.

Fig. 19 is the screenshot of the output of the mobile agent 1 at the sensor node 1 in the simulated damage pattern. The name of the embedded computer at sensor node 1 is gumstix 2. Each Gumstix board runs mobile agent server program waiting for mobile monitoring agents. When the mobile monitoring agent 1 arrives at the sensor node 1, it calculates the AR coefficients for the acceleration data. The calculated AR coefficients are displayed on a secure shell connected to the embedded computer, gumstix 2, on the sensor node 1 as shown in Fig. 19. The sensor data feature vector is formed using these coefficients, and the Euclidean distances from this feature vector to the memory cells are computed. The 9 smallest Euclidean distances are displayed on the gumstix 2 terminal in Fig. 19. All the 9 distances are from the sensor data feature vector to the memory cells in the damaged memory cell group. As a result, the structure pattern is classified as damaged pattern.

To test the performance of the presented mobile monitoring agent-based AIPR method, a number of tests at different excitation voltage levels and different patterns are conducted. The test results are summarized in Table 2. In the normal pattern, 128 tests and 119 tests are performed at the 0.275 and 0.295 V excitation voltage, respectively. All the test results show the normal pattern of the structure. The success rates are 100% in the normal pattern. In the damaged pattern, 143 tests and 140 tests are conducted at the 0.275 and 0.295 V excitation voltage, respectively. At the 0.275 V excitation level, 119 tests out of 143 tests classify the structure to the damaged pattern. At the 0.295 V excitation level, 133 tests out of 140 tests classify the structure to the damaged pattern. The success rates in the damaged pattern are 83.2% and 95% for the 0.275 and 0.295 V excitation voltage, respectively.

6. Conclusions

An artificial-immune-system-based network framework for adaptive monitoring is presented in this paper. The presented network framework transforms immune concepts and operating principles into computational models and embeds these computational models into engineered monitoring networks. The immune cells are mimicked by the mobile monitoring agents and the adaptive immune responses are simulated by the agent interaction and collaboration. The presented immune-inspired monitoring paradigm has been applied to the structural health monitoring networks. The adaptive structural health monitoring is achieved through the integration of high computational power sensor nodes, a mobile-agent-based sensor network middleware, and the AIPR-based structural damage pattern recognition. An embeddable mobile-agent-based sensor network middleware manages the generation, maturation, and distribution of mobile monitoring agents automatically. The mobile monitoring agents

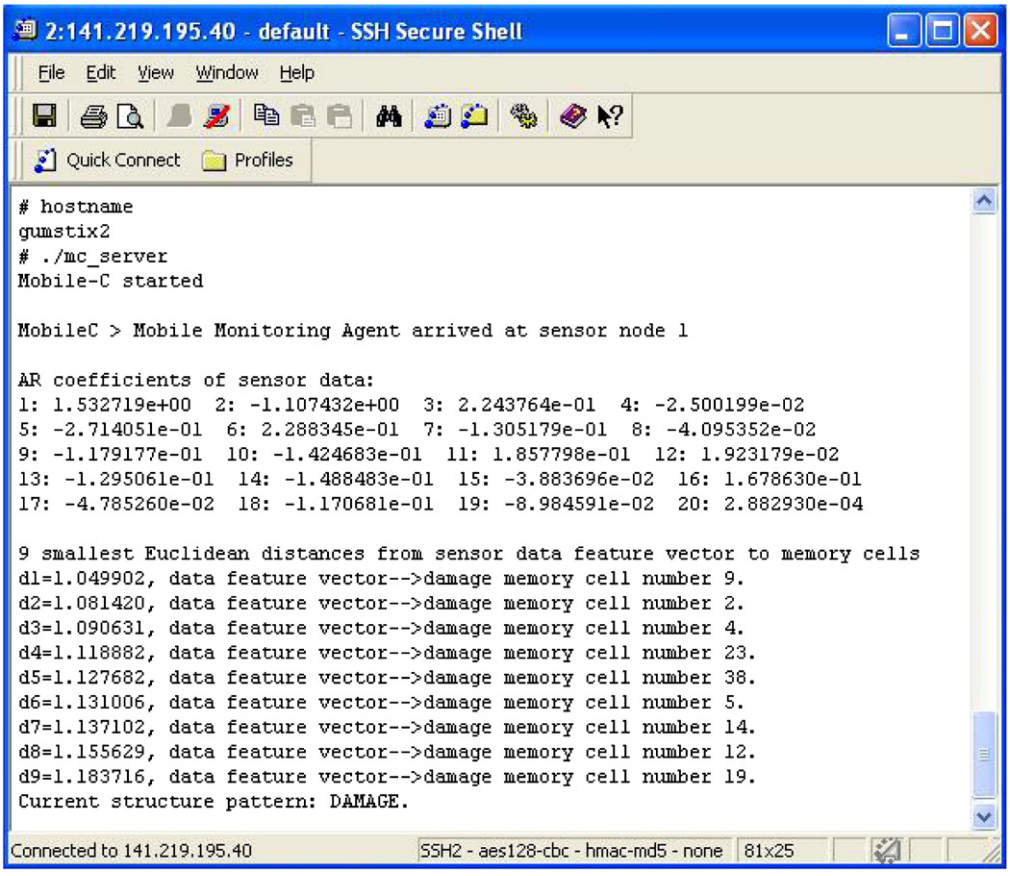

Fig. 19. The output of the mobile agent 1 on sensor node 1 in the simulated damage case.

Table 2
Mobile monitoring agent-based AIPR test results.

Excitation Voltage (V)	Normal Pattern			Damaged Pattern		
	Number of tests	Number of Normal Patterns	Success rate (%)	Number of tests	Damage Detected	Success rate (%)
0.275	128	128	100	143	119	83.2
0.295	119	119	100	140	133	95

and the AIPR-based structural damage diagnosis have been tested in a scaled bridge structure. The mobile monitoring agent was able to recognize structural damage pattern using real-time structural acceleration signals.

References

Al-Kasassbeh M, Adda M. Network fault detection with Wiener filter-based agent. Journal of Network and Computer Applications 2009;32(4):824–33.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. LAPACK users' guide. Society for Industry and Applied Mathematics 1999.

Boukerche A, Machado RB, Juca KRL, Sobral JBM, Notare MSMA. An agent based and biological inspired real-time intrusion detection and security model for computer network operations. Computer Communications 2007;30:2649–60.

Boulis, A., Han, C.-C., Srivastava, M.B. Design and implementation of a framework for efficient and programmable sensor networks. In: Proceedings of the first international conference on Mobile systems, applications and services MobiSys '03 San Francisco, California; 2003. pp. 187–200.

Carter JH. The immune system as a model for pattern recognition and classification. Journal of the American Medical Informatics Association 2000;7(1):28–41.

Castiglione F, Motta S, Nicosia G. Pattern recognition by primary and secondary response of an artificial immune system. Theory in Biosciences 2001;120(2):93–106.

Chen, B. Monitoring structure's health: applying lessons learned from the immune system, SPIE Newsroom; 2009.

Chen B, Cheng HH, Palen J. Mobile-C: a mobile agent platform for mobile C/C++ agents. Software-Practice & Experience 2006;36(15):1711–33.

Chen, B., Tomizuka, M. OpenSHM: open software architecture design of structural health monitoring algorithms in wireless sensor nodes. In: Proceedings of the 2008 IEEE/ASME international conference on mechatronic and embedded systems and applications, Beijing, China; 2008.

Chen B, Zang C. Artificial immune pattern recognition for structure damage classification. Computers & Structures 2009;87(21–22):1394–407.

Dasgupta D, Yu S, Majumdar NS. MILA–multilevel immune learning algorithm and its application to anomaly detection. Soft Computing 2005;9(3):172–84.

de Castro LN. Fundamentals of natural computing: basic concepts, algorithms, and applications. Chapman & Hall/CRC; 2006.

de Castro LN, Timmis J. Artificial immune systems: a new computational intelligence approach. Springer; 2002.

Delves PJ, Martin SJ, Burton DR, Roitt IM. Roitt's essential immunology. Blackwell Publishing; 2006.

Harmer PK, Williams PD, Gunsch GH, Lamont GB. An artificial immune system architecture for computer security applications. IEEE Transactions on Evolutionary Computation 2002;6(3):252–80.

Ilarri S, Mena E, Illarramendi A. A system based on mobile agents to test mobile computing applications. Journal of Network and Computer Applications 2009; 32(4):846–65.

Jabeur N, McCarthy JD, Xitao X, Graniero PA. A knowledge-oriented meta-framework for integrating sensor network infrastructures. Computers & Geosciences 2009:809–19.
Lanaridis, A., Karakasis, V., Stafylopatis, A. Clonal selection-based neural classifier. In: Proceedings of the 2008 eighth international conference on hybrid intelligent systems (HIS); 2008, pp. 655–660.
Levis P, Culler D. Mate: A tiny virtual machine for sensor networks. ACM Sigplan Notices 2002;37(10):85–95.
Negoita, M. Artificial immune systems–an emergent technology for autonomous intelligent systems and data mining. Autonomous Intelligent Systems: agents and data mining. International Workshop, AIS-ADM 2005. In: Proceedings (Lecture Notes in Artificial Intelligence. vol. 3505) 2005, 19-36|xiii+301.
Press WH, Flannery BP, Teukolsky SA, Vetterling WT. Numerical recipes in C: the art of scientific computing. Cambridge University Press; 1992.
Sarafijanovic S, Le Boudec JY. An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks. IEEE Transactions on Neural Networks 2005;16(5):1076–87.
Shi Z, Huang H, Luo J, Lin F, Zhang H. Agent-based grid computing. Applied Mathematical Modelling 2006;30(7):629–40.
Stepney S. Embodiment. In: Flower D, Timmis J, editors. Silicon Immunology. Springer; 2007.
Szumel, L., LeBrun, J., Owens, J.D. Towards a mobile agent framework for sensor networks. In: Proceedings of the second IEEE workshop on embedded networked sensors (IEEE Cat. No. 05EX1105): 79–87|x+168, 2005.
Tarakanov AO. Immunocomputing for intelligent intrusion detection. IEEE Computational Intelligence Magazine 2008;3(2):22–30.
Theodoridis S, Koutroumbas K. Pattern recognition. Academic Press; 2008.
Watkins A, Timmis J, Boggess L. Artificial immune recognition system (AIRS): an immune-inspired supervised learning algorithm. Genetic Programming and Evolvable Machines 2004;5(3):291–317.
Zhong YF, Zhang LP, Gong JY, Li PX. A supervised artificial immune classifier for remote-sensing imagery. IEEE Transactions on Geoscience and Remote Sensing 2007;45:3957–66.
Zhong YF, Zhang LP, Huang B, Li PX. An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery. IEEE Transactions on Geoscience and Remote Sensing 2006;44(2):420–31.
Zhu Q, Aldridge SL, Resha TN. Hierarchical Collective Agent Network (HCAN) for efficient fusion and management of multiple networked sensors. Information Fusion 2007:266–80.