IOP Conf. Series: Materials Science and Engineering **419** (2018) 012014 doi:10.1088/1757-899X/419/1/012014

Structural health monitoring for combined damage states

Martin A Butler[1], James A Swanson[1] and Gian A Rassati[1]

[1]University of Cincinnati, 2850 Campus Way, Cincinnati OH, United States

E-mail: butlerm3@mail.uc.edu

Abstract. The current state of the art in automated structural health monitoring (SHM) is to compare the results of a set of sensors to a set of previously established values, picking the most similar set. To identify a damage state, it must be envisioned upon the creation of the SHM system and analysed to determine the output. Following catastrophic events, bridges may be damaged in several ways in several locations, causing problems for the automated system in identifying the exact repairs that are necessary or whether sufficient strength remains to sustain reduced loading. An alternative method to modelling a massive number of damage states and memorizing them all is presented based on subdivided attractor networks. These artificial neural networks are similar to the feedforward networks that have found much use in industry. Instead of acting as an approximator for some function, these networks work as content addressable memory. Partitioning these networks allows different portions of the network to represent different portions of the structure, and to mix and match states that may be associated with each other. In this way a SHM regimen is produced that needs far fewer memorized states to recall all of the possible damage states.

1. Structural Health Monitoring System

Structural Health Monitoring (SHM) is used to maintain constant knowledge of a structures state, this may be necessary for particularly important pieces of infrastructure, or a bridge that is approaching the end of its design life that it may not be economical to replace. Current state of the art is to compare the results of a set of sensors to known sets of data that have been already established using some algorithm to determine the most similar damage state. Many different approaches have found success including effective independence method [1], Fourier component analysis [2], and time series analysis [3]. In order to do this, damage states must be thought of beforehand so that the structures state may be compared. Combinations of damage are difficult to discern, as they will resemble both of the states in some ways, and neither in some ways. Damage states in one locale may, through an alternative load path, result in a strain state that somewhat resembles damage to the subsystem that provides this alternative load path. The proposed SHM system is meant to avoid these problems, by producing a network of subsystems that communicate with each other regarding the state of each subsystem.

The proposed SHM system uses subdivided attractor artificial neural networks to identify the state of the structure. Specifically, strains are collected describing the bending and axial stresses at multiple members within the subsystem, these strains are fed through a feedforward artificial neural network (FF-ANN). The FF-ANN outputs a bottleneck image for the SA-ANN, a set of interface synaptic weights processes the bottleneck image to the full sized initial image. Finally, the SA-ANN converges to one of its stable neural states, describing the structure.

For this work strains are the inputs considered. Static behaviour is more sensitive to localized damage which is what is most important when identifying multiple damages in one structure [4]. Use of static behaviour has the added benefit of avoiding interference from local modes of vibration overlaid with global ones [2].

1.1. Pony Truss Bridge

The example bridge is a small pony truss bridge. A five panel bridge with a corrugated steel deck, seven rows of stringers, and skew of 20 degrees. The nonlinear behaviour of the connections was modelled using nonlinear links. The nonlinear behaviour is defined in the way suggested in [5]. The only loads applied to the structure were dead loads. Two kinds of damage were introduced to this structure: release and stiffness reduction. Release type damages include the introduction of a release on one of the ends of the frames. What releases and where they were applied depends on the type of member, truss members only experience axial releases, except for vertical members which are zero force members and are not considered for monitoring. Floor beams and stringers release moments, shears, and axial constraints at either end. The second damage type is property modifiers to the member, factoring by 0.3 the area or moment of inertia of the section. The releases and section reductions are shown in Table 1.

Table 1. Breaks and Softening states of the Pony Truss Example

Break Classification

	P_i	V_{2i}	V_{3i}	T_i	M_{21}	M_{3i}	P_j	V_{2j}	V_{3j}	T_j	M_{2j}	M_{3j}
Bottom Chord	1	0	0	0	0	0	0	0	0	0	0	0
Top Chord	1	0	0	0	0	0	0	0	0	0	0	0
Diagonals	1	0	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0	0	1	1
Verticals	0	0	0	0	0	0	0	0	0	0	0	0
Floor Beams	1	1	1	0	1	1	0	0	0	0	0	0
	0	0	0	0	0	0	1	1	1	0	1	1
Stringers	0	1	1	0	1	1	0	0	0	0	0	0
	0	0	0	0	0	0	0	1	1	0	1	1

Softening Classification

	A	As2	As3	J	I22	I33	Mass	Weight
Bottom Chord	0.3	1	1	1	1	1	1	1
Top Chord	0.3	1	1	1	1	1	1	1
	1	0.3	0.3	1	0.3	0.3	1	1
Diagonals	0.3	1	1	1	1	1	1	1
Verticals	1	1	1	1	1	1	1	1
Floor Beams	1	1	1	1	1	1	1	1
Stringers	0.3	1	1	1	1	1	1	1
	1	0.3	0.3	1	1	0.3	0.3	1

The subsystems considered were the two trusses and the floor system. The two trusses each have 53 damage states, they also have another 6 and 7 states that indicate that the substructure is not damaged, these either correlate to the whole structure being intact or other damage states existing in one of the other systems. The floor system is larger, with 82 states indicating the damage of the floor system, and a further 18 states indicating an intact structure or damage to other subsystems. The two trusses each had 11 states which induced collapse under dead load. These included eight of the top chord elements the diagonals at both ends, and one chord member. They were not included in training data.

2. Artificial neural networks as content addressable memory

SHM requires some degree of signal processing to discern what state the structure is in. For this work subdivided attractor networks are used, these are Hopfield networks of McCulloch-Pitts neurons that have been cleaved into several subnetworks [6, 7]. When not divided, attractor networks act as content addressable memory, converging to the most similar state to what has been initiated on the network [8]. This is done by connecting each neuron to each other neuron with a synaptic weight, the more often the neurons activate together (or are inactive together), the larger the weight, the more often one activates and the other does not the smaller the weight. McCulloch-Pitts neurons have values of +/-1, whether the value changes in an iteration is based on the signal given by the rest of the network; this signal is:

$$h_i(t) = \left(\sum_j^N J_{ij} s_j(t-1)\right) \quad (1)$$

Where: $s_i(t)$ is the state of the ith neuron at iteration t, J_{ij} is the synaptic weight between neuron i and j, and N is the number of neurons.

Other things may add to this signal field such as biases, new information, or noise meant to draw the network out of local minima. It is helpful to consider the negative of the signal field analogous to energy, with the attractor network falling into the lowest energy state. Simulated annealing is used, in this case a constant heat bath is used to allow the network to raise out of local minima.

2.1 Subdivided attractor networks

Subdivisions produce several smaller subnetworks, each with their own output called subdivided attractor artificial neural networks (SA-ANN) [7]. The synaptic weights between the nodes of the local subnetwork are treated normally, but the weights between the different subnetworks are reduced. The multiple networks identify different damage states that are non-orthogonal. Because the networks states are related to one another, an exchange of signals occurs. As an example, in identifying Superman as he flies overhead, he is typically misidentified as a bird or a plane. The movement type (flying) and the shape (man) are not typically combined, the movement type strongly suggest that what is seen is either a bird or a plane. The other subnetworks add a term to the signal, making it:

$$h_i(t) = \left(\sum_j^{N/q} J_{ij} s_j(t-1)\right) + \left(\sum_j^{N(1-q)/q} J_{ij} s_j(t-1)\right) \quad (2)$$

Where q is the number of subnetworks

For this work, we divide the structure into subsystems each with their own attractor network monitoring the state of the strains in members due to moment and axial loads of members. Damage to one substructure will affect the stress state of the rest of the structure, though not so much as the local damages will. The signal between subnetworks is of less import than internal signals, and in the case of a bridge with several spans, the import of the signal is a function of how close the subsystems are to each other. If the states are related, the subnetworks work together to determine the state of each network, in the case of a superman, this is the case of an odd looking bird being positively identified as a bird thanks to its movement, not merely its shape. However, because of the reduction in signal from outside the subnetwork, it is possible to remix the states of different subnetworks to those which are not usually associated.

Such an event would be damage present in multiple subsystems at once. If the full system is monitored this may produce states that are not recognizable as one pre-modelled damage state or another. Conversely, a regimen of independently monitoring subsystems may be able to identify combinations of damage states, but they will be more easily confounded by damages outside of the subsystem, likely requiring more inputs. The SA-ANN combines the benefits of both these methods while mitigating the drawbacks.

2.1. Feedforward Artificial Neural Networks

Feedforward artificial neural networks are a flexible way to categorize inputs or interpolate and extrapolate a function that is difficult to model directly [9]. They are by far the form of artificial neural networks to see the most application [7, 10]. In the proposed work, FF-ANN are used to categorize the state of a substructure based on a set of strain data gathered from that substructure. While in this work FF-ANN are used as an intermediate step in identifying the damage state of a structure, they have been used as SHM systems in their own right [11].

2.2. Bottleneck image
SA-ANN require a large number of neurons to function, far more than is necessary to describe as many states it memorizes in more conventional computing. The FF-ANN require a more complex training regimen than SA-ANN, so it is desirable to have fewer synaptic weights as long as there are enough to sufficiently approximate the output. The FF-ANN is set to output a bottleneck image of logistic function defined nodes that is smaller than the full subnetwork. This bottleneck image is multiplied by a set of one way synaptic weights created by Hebbian learning, in the same way as synaptic weights are defined for the SA-ANN. The reduction in size will necessarily cause some loss of fidelity relative to routing the FF-ANN directly onto the initial state of the attractor network. However the reduction in training time for the FF-ANN is worth it as the SA-ANN subnetworks are quite large. The damage to the initial network state caused by the bottleneck was examined as a function of the size of the bottleneck at several levels of damage to the bottleneck image. The results are shown in Figure 1. They indicate that even for a perfect image in the bottleneck, there is substantial damage to the initial network image. As the bottleneck approaches the full size of the network however, the interface between the two begins to act as an iteration of the SA-ANN and it begins to have lower damage to the initial subnetwork state than to the bottleneck state. The 50% size of bottleneck was selected for this work.

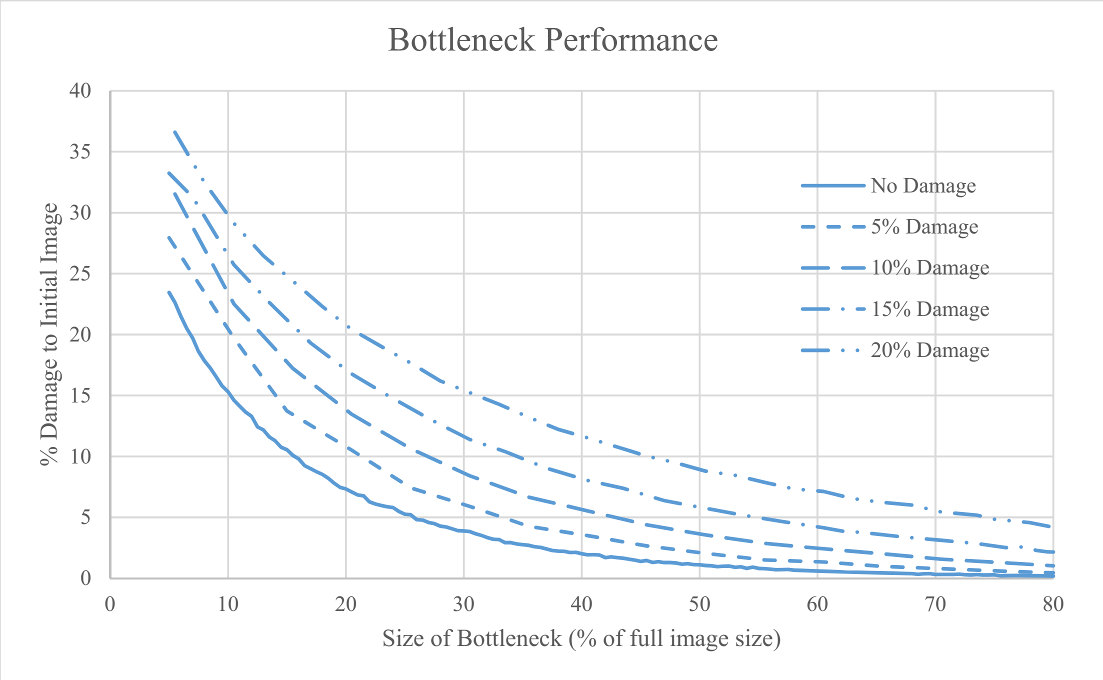

Figure 1. Averages of the effects of bottlenecks on 100 networks with 20 memorized states

2.3. Grouping
While each damage state considered requires its own memory state in its associated subnetwork, this is not the case for other subsystem damages. Each damage state is associated with a network state in each subnetwork. If the damage is not located in the subnetwork's subsystem, the associated subnetwork state will be shared with other less pertinent damage states. For example, in a long span bridge with many supports damage near one end of the bridge will have little effect on the other end. Part of this is covered

by reducing the magnitude of the synaptic weights, but the physical state of the undamaged bridge end will be approximately the same, thus only a couple subnetwork states are memorized to cover all the damages to the other subsystems. This reduces the signal to the subnetwork, specifically the second term of equation 2. This reduction may not appear to necessarily happen when one looks at equation 2, however if the same subnetwork state in one subnetwork is associated with every single network state in another subnetwork the signal must necessarily go to zero from this term, just as if the whole world were painted green, the color green would not suggest that an object is a leaf rather than a piece of paper.

A method of grouping was developed based on the distance between states. The input vectors were standardized and principal component analysis was performed. The square distance between states was taken as:

$$D_{ij} = \sum_{k}^{N} (S_{i\,k} - S_{j\,k})^2 \qquad (3)$$

Where D_{ij} is the standardized square distance between the ith and jth input vector and S_{ik} is the k^{th} value for the i^{th} input vector. The values in D_{ij} are then compared to each other; if they are less than some previously determined distance from each other they are placed into the same group. This does allow for a group "drift" as each new member slightly changes the bounds of what is considered similar to the group for the next vector.

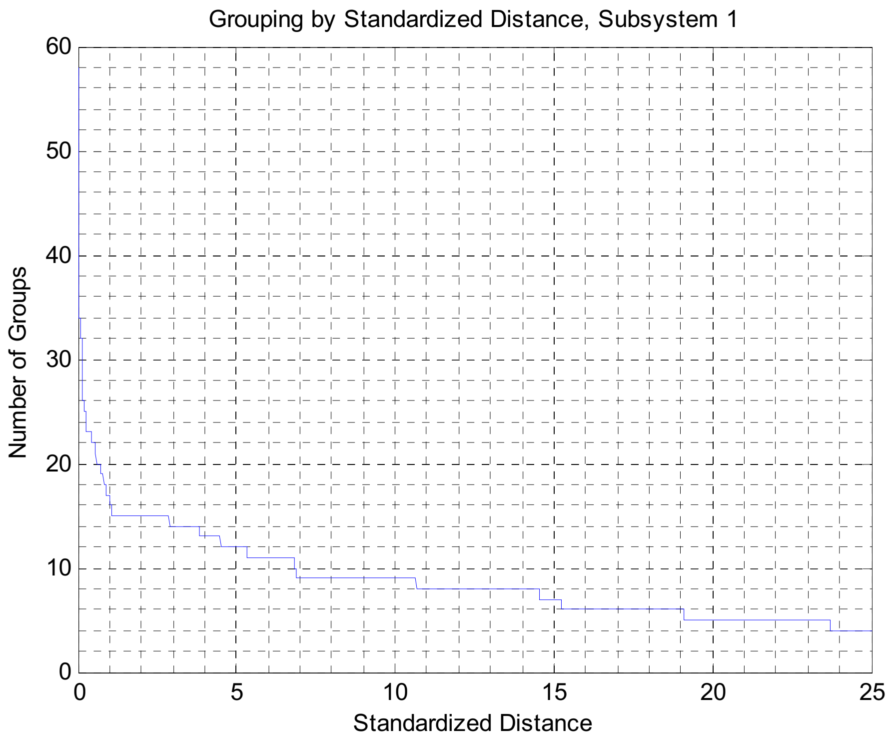

Figure 2. Number of groups of non-pertinent damage states based on normalized square distance

For the example pony truss bridge, subsystem 1 is a truss, it has 96 damage states to remember, there are 152 damage states for system 3 and another 96 damage states for the other truss, for a total of 248 non-pertinent states for this subnetwork. Initial increases in the acceptable square distance give precipitous decrease in the number of groups, but the decrease slows as the acceptable distance increases as shown in figure 2. It appears that grouping the non-pertinent input states into 5 to 9 states will be stable with substantial similarities between all group members. For the two trusses, (subsystems 1 and 2) we sort the non-pertinent groups into 7 and 6 groups respectively, for the floor system (subsystem 3) the non-pertinent states are grouped into 18 states. The floor system is closer to both the trusses and has substantially more sensors, allowing it greater granularity.

3. Methodology

The proposed SHM scheme was produced in Matlab and trained on strains from a nonlinear model produced in SAP2000. The nonlinear properties of the connections of the pony truss bridge were considered. Following this, some strain states were produced that the system was not trained to identify, including damage to multiple systems at once. These combined damage states were made up of damage states that the system was trained to recognize

The FF-ANN were defined using a standard backpropagation algorithm with momentum and regulation, as well as normally distributed noise on the inputs to prevent overfitting and ensure generality. Principal component analysis was used to reduce the inputs to the FF-ANN from 23 inputs for the trusses and 242 for the deck system to 13 and 65, respectively. The trusses each had 90 nodes in one processing layer, while the floor system had 250 nodes in its processing layer.

Results

The SHM system is able to identify the states it is trained for, but this is not all that is expected of this system. It was also subject to variety of damage states that it was not trained to recognize. The success of the network at identifying these untrained combinations is shown below in Table 2. All of the tested states are shown, failures of a subnetwork to converge to the correct neural state are shaded in. When the minor damage types such as merely reducing the moment of inertia or cross sectional area to 30% of what it is when intact, are paired with more substantial damage types such as truss member fracture, the system often fails to discern the more minor damage, either failing to converge to a state at all or converging to a state indicating that the subsystem is undamaged but the other subsystem is damaged. This second state at least enjoys some of the benefits of the subdivisions.

When significant damage is experienced in locations that are fully distinct in the geometry the system works at its best, such as when a stringer breaks off in the middle of the deck and a chord member completely fails. The physical system being a single span non-redundant structure may have reduced the success rate of this initial system, the systems are tightly associated with each other, and attempts to introduce multiple instances of fractures often result in collapse.

Table 2. Performance of SA-ANN on Combinations of Damage

TRUSS 1	TRUSS 2	FLOOR
bottom chord 1 axial loss	diagonal 16 RBS	intact
bottom chord 2 axial loss	diagonal 16 RAS	intact/damage elsewhere
bottom chord 3 axial loss	diagonal 15 RBS	intact/damage elsewhere
bottom chord 4 axial loss	diagonal 15 RAS	intact/damage elsewhere
top chord 6 axial loss	diagonal 14 RBS	intact/damage elsewhere
top chord 7 axial loss	diagonal 14 RAS	intact/damage elsewhere
diagonal 2 axial loss	diagonal 13 RBS	intact/damage elsewhere
diagonal 3 axial loss	diagonal 13 RAS	intact/damage elsewhere
diagonal 4 axial loss	diagonal 12 RBS	intact
diagonal 5 axial loss	diagonal 12 RAS	intact
diagonal 6 axial loss	diagonal 11 RBS	intact/damage elsewhere
diagonal 7 axial loss	diagonal 11 RAS	intact/damage elsewhere
bottom chord 1 RAS	diagonal 10 RBS	intact
intact/damage elsewhere	bottom chord 1 RAS	intact/damage elsewhere
intact/damage elsewhere	bottom chord 2 axial loss	intact/damage elsewhere
diagonal 8 RBS	bottom chord 3 axial loss	intact/damage elsewhere
diagonal 8 RAS	bottom chord 4 axial loss	intact/damage elsewhere
diagonal 7 RBS	top chord 6 axial loss	intact/damage elsewhere
diagonal 7 RAS	top chord 7 axial loss	intact/damage elsewhere
diagonal 6 RBS	diagonal 2 axial loss	intact/damage elsewhere
diagonal 6 RAS	diagonal 3 axial loss	intact/damage elsewhere
diagonal 5 RBS	diagonal 4 axial loss	intact
diagonal 5 RAS	diagonal 5 axial loss	intact
diagonal 4 RBS	diagonal 6 axial loss	intact/damage elsewhere
diagonal 4 RAS	diagonal 7 axial loss	intact/damage elsewhere
diagonal 3 RBS	bottom chord 1 RAS	intact
bottom chord 2 RAS	intact/damage elsewhere	FB 1 end 1 break
bottom chord 3 RAS	intact/damage elsewhere	FB 1 end 2 break
intact/damage elsewhere	diagonal 15 axial release	FB 2 end 1 break
intact/damage elsewhere	bottom chord 4 RAS	FB 2 end 2 break
top chord 1 RAS	top chord 5 axial release	intact
bottom chord 4 RAS	bottom chord 5 RAS	intact
bottom chord 5 RAS	intact/damage elsewhere	floor beam 1 release end 1
top chord 2 RAS	intact/damage elsewhere	floor beam 1 release end 2
bottom chord 2 axial release	intact/damage elsewhere	floor beam 2 release end 1
bottom chord 4 axial release	intact/damage elsewhere	floor beam 2 release end 2
top chord 7 axial release	intact/damage elsewhere	floor beam 3 release end 1
intact/damage elsewhere	bottom chord 1 axial release	floor beam 3 release end 2
intact/damage elsewhere	bottom chord 2 axial release	floor beam 4 release end 1
intact/damage elsewhere	bottom chord 3 axial release	floor beam 4 release end 2
intact/damage elsewhere	bottom chord 4 axial release	stringer 1 release end 1
intact/damage elsewhere	top chord 4 axial release	stringer 1 release end 2
intact/damage elsewhere	top chord 5 axial release	stringer 2 release end 1
intact/damage elsewhere	diagonal 3 axial release	stringer 2 release end 2
RAS = Reduced Axial Stiffness		
RBS = Reduced Bending Stiffness		

Conclusion

We propose a new method of SHM using SA-ANN, it allows for multiple damage states to be identified at once. The SA-ANN's ability to discern small damages alongside significant damages is marginal, showing the same weakness in discerning minor damages alongside large damages that is present in other approaches. In cases of damages of similar significance the combinations are determined more easily. The relationship between either of the trusses in the example problems and the floor system is closer than the relationship between the two trusses. The minor damages considered in the reduced area and bending stiffnesses, particularly in the bending stiffness of truss members are too small for this SHM system to identify in any but optimum conditions. This approach may have greater success with systems with more alternative load paths than a pony truss bridge, where the subsystems are less closely related.

References

[1] Yi, T.H., Li, H.N. and Gu, M., 2011. Optimal sensor placement for structural health monitoring based on multiple optimization strategies. The Structural Design of Tall and Special Buildings, 20(7), pp.881-900.
[2] Antonacci, E., De Stefano, A., Gattulli, V., Lepidi, M. and Matta, E., 2012. Comparative study of vibration-based parametric identification techniques for a three-dimensional frame structure. Structural Control and Health Monitoring, 19(5), pp.579-608.
[3] Sohn, H. and Farrar, C.R., 2001. Damage diagnosis using time series analysis of vibration signals. Smart materials and structures, 10 (3), p.446.
[4] Jenkins, C.H., Kjerengtroen, L. and Oestensen, H., 1997. Sensitivity of parameter changes in structural damage detection. Shock and Vibration, 4(1), pp.27-37.
[5] Butler, M.A., Swanson, J.A., Rassati, G.A. and Dues, E.F., 2018. High Resolution Modeling and Modeling of Connections in Pony Truss Bridges (No. 18-03496).
[6] Coughlin, J.P. and Baran, R.H., 1995. Neural computation in hopfield networks and boltzmann machines. (University of Delaware Press)
[7] Bar-Yam, Y., 1997. Dynamics of complex systems. (Reading, MA: Addison-Wesley)
[8] Amit, D.J., 1992. Modeling brain function: The world of attractor neural networks. Cambridge university press.
[9] Samarasinghe, S., 2016. Neural networks for applied sciences and engineering: from fundamentals to complex pattern recognition. CRC Press.
[10] Muttil, N. and Chau, K.W., 2007. Machine-learning paradigms for selecting ecologically significant input variables. Engineering Applications of Artificial Intelligence, 20 (6), pp.735-744.
[11] Bakhary, N., Hao, H. and Deeks, A.J., 2007. Damage detection using artificial neural network with consideration of uncertainties. Engineering Structures, 29 (11), pp.2806-2815.