3. Technique Evaluation - PSOM

This chapter deals with the testing, comparison and evaluation of the PSOM algorithm. Both the PSOM and EPSOM network were run through identical tests but the PSOM was chosen for detailed analysis. This was because the PSOM yielded consistently better results. Further comparative analysis can be found in Chapter 6. The aim of presenting these results is to demonstrate the operation of the networks, the effects of the parameters on the operation of the algorithms and to compare the PSOM with GNG-U (Fritzke, 1997). In the tests that follow, three types of data set were used.

1. An artificial data set was generated to allow comparison between the different techniques. This data set was designed to show the capabilities of the different networks. This set was generated with 5 classes.
2. A real-world data set was taken from Thales Research and Technology (UK) and consisted of radar pulse data.
3. A two dimensional dataset with two classes as specified in Fritzke's seminal work on GNG-U (1997).

Testing dynamic Neural Networks (as defined in Chapter 1) is not a trivial matter since the dynamic type of data required to test the unique dynamic nature of the networks is not readily available. With their static counterparts, it is possible to have a final gauge of the quality of the network's operation. There is a final state and a final error or at the very least a stopping criterion. Also, there are a number of repositories of static data sets for comparison purposes (e.g. Prechelt 1994). However, with dynamic Neural Networks, there is no final state. One might halt the process temporarily but it should be noted that the system driving the input may continue to change after that point. For a dynamic network, it is important that any halted state should be considered temporary and that the network may be required to continue operating after that. Each test must be created such that the network's behaviour is observed over time. The outputs for the networks include:

- The Class Output, this is the arbitrary number assigned to each class within the PSOM. This arbitrary class number can then be matched with known classes during training if required.
- The Recognition Error which is the Euclidean distance between the input and the focus (best match).
- The number of links to gauge the size of the network.
- A class association matrix that compares the labels of the input class to that of the output class.
- The topology of the network. The interpretation of structure can be used to infer more information about the input space – particularly in the case of GNG-U.

All of these outputs are dynamic and are studied over time.

Comparisons with static Neural Networks are difficult as the construction of fair tests is not possible. If one were to show the entire data set to a static network, then it is likely to learn the patterns with more success than a dynamic network. However, one can assume that not all the patterns are present at the time of training static network; how many patterns should therefore be included in the training process? It is clear that any patterns not included in the static network training process will not be classified correctly. Thus, setting comparison tests between static and dynamic networks will mainly show differences in the manner in which the data is presented to the networks.

All tests were created using custom written software written in Java and available for public use[1]. This software allows many tests to be run such that the data can be collated at the end of all testing (Figure 3.1). The tests are numbered to give a logical flow although they may not have been performed in order.

[1] The RAIH class files can be found at – http://www.maison-de-stuff.net/rob/ai.html, version 2.x was used for all tests.

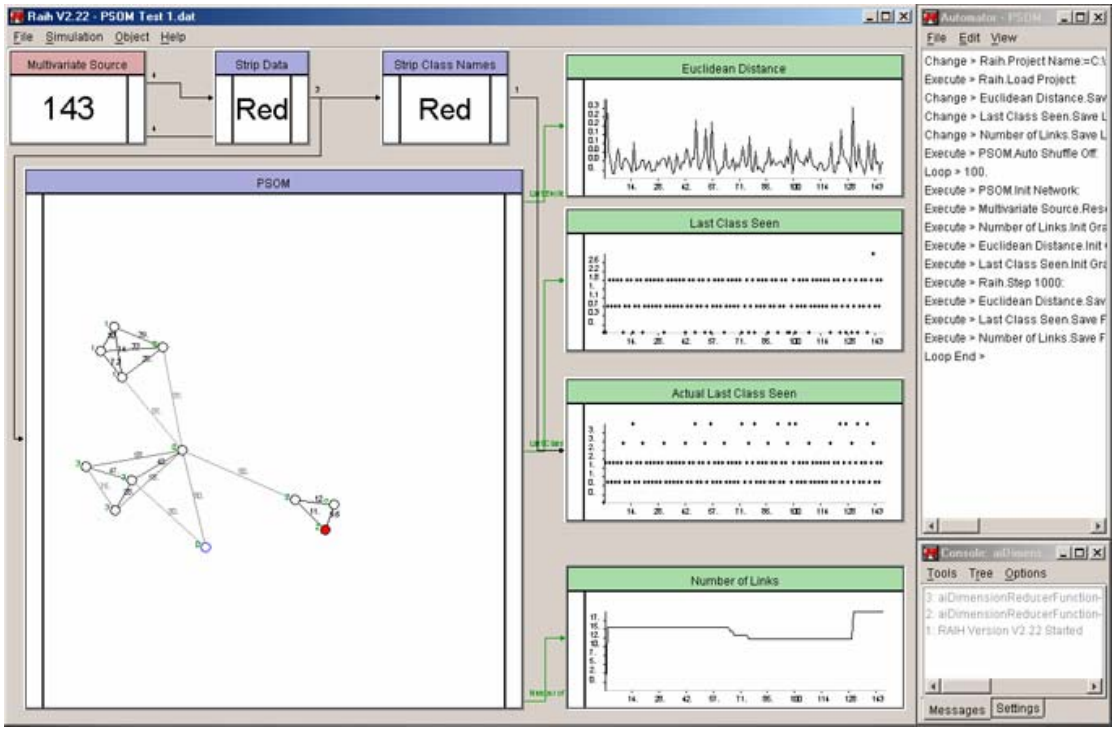

Figure 3.1 Rob's Artificial Intelligence Harness (RAIH)

3.1 Artificial Data Sets

These artificial datasets were constructed to test the temporal properties of the neural networks and to facilitate this, the classes were created to be linearly separable. Once the network's operation on linearly separable problem was understood, more complex problems could be posed. Linearly separable problems are easily solved with static networks but the aim here was to demonstrate the dynamic properties. In the list below, each numbered item represents the number of the class assigned to that particular type of pattern.

1. **Regular Patterns.** If a pattern is regularly occurring and persistent, then each network should continue to recognise the pattern.
2. **Stopping Pattern.** A pattern that disappears from the data set. Each technique should eventually forget the pattern that has disappeared.
3. **Infrequent Patterns.** A pattern is infrequent if it appears in a regular way but with a large time difference between appearances.

4. **Irregular Patterns.** An irregular pattern appears at random intervals. This type of pattern is used to show that the network is not responding to the frequency of pattern occurrence.
5. **Late Starting Pattern.** A pattern that appears late in the operation. Each technique should accept the new pattern.

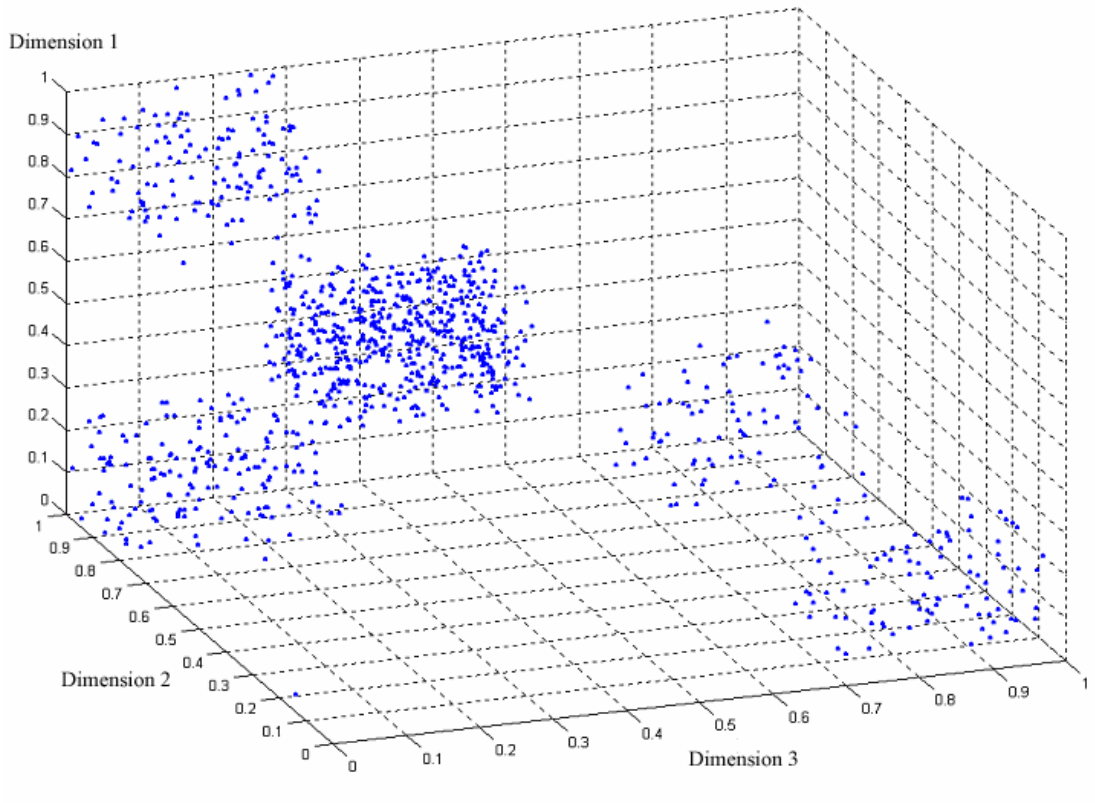

Figure 3.2 The Artificial Multivariate Data Set.

The first version of the artificial data set contains five classes from three arbitrary dimensions (Figure 3.2), each class testing one of the above criteria. A pattern is a binary pattern that is corrupted by 20% noise, between –0.2 and 0.2. The algorithm for this process is included in Appendix A.1. A useful measure, used later, is the number of times a class is presented to the network.

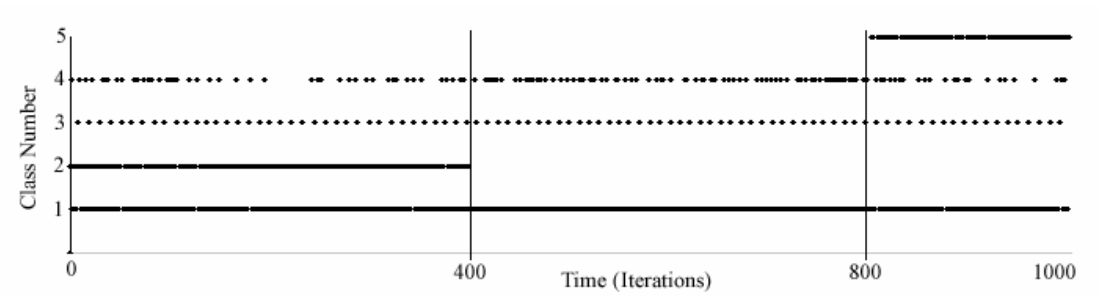

Figure 3.3 A graph depicting when the classes are present in the network.

Figure 3.3 shows when the classes appear in the network. Each dot on the graph represents one pattern or instance of the class. Table 3.1 shows the number of occurrences of each class in the Artificial Data Set.

Table 3.1. The number of occurrences for each class in the Artificial Data Set.

Class	Number of Occurrences
1	527
2	160
3	96
4	136
5	81
Total	1000

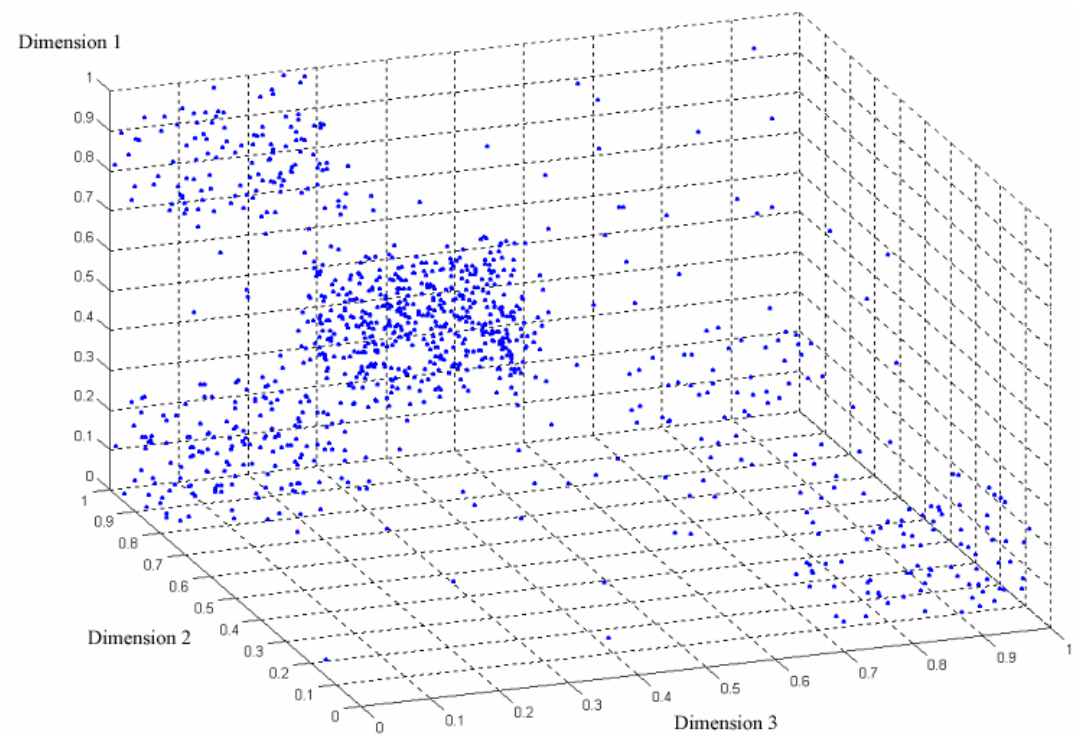

Figure 3.4 Artificial Multivariate Dataset with 20% noise and 100 pure noise patterns.

Although the first artificial data set contained patterns that were corrupted by random noise (20%), it was assumed that all the patterns belonged to a class. This form of noise does not allow for inputs which do not belong to any class and are purely noise. The noise rejection properties of the PSOM were tested by randomly inserting noise patterns into the data set (Figure 3.4).

3.2 Radar Data Set

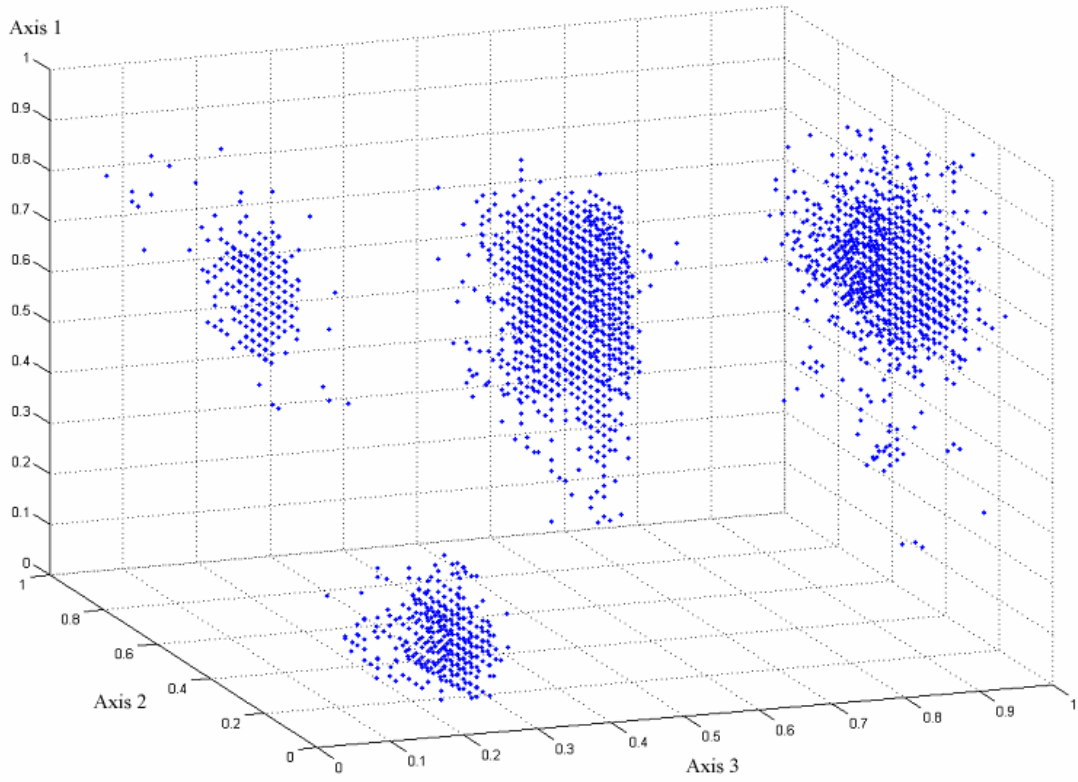

Figure 3.5 Radar Data Set showing four classes.

The real data set used was provided by Thales Research Technology[2] (TRT), Sensors (Figure 3.5). The data consists of four different sources, called emitters. Each emitter releases pulses of electromagnetic radiation which have three properties: pulse width, bearing and amplitude. The class number was arbitrarily set to be the type of source. This data set tests two facets of the network. Firstly, it allows us to test whether the network can consistently classify an input class as an arbitrarily set network class. This means that given all the patterns belonging to an input class will be classified as

[2] Due to the sensitivity of the data, it may not be published in a digital format, nor the axis labelled or true scales published.

members of the same class in the network. Secondly, the network can be tested for its ability to detect novelty. In many applications, the arrival of a type of radar pulse that has not previously been detected is particularly useful, especially when flagged to the user as a new occurrence. The semi-regular spacing of points is a by-product of quantisation in the digitisation process.

3.3 2 Dimensional Data Set

This artificial data set was recreated from the data set specified in the GNG-U work (Fritzke 1997). It was chosen as it shows GNG-U in a very positive light as network parameters have already been selected to give the desired behaviour.

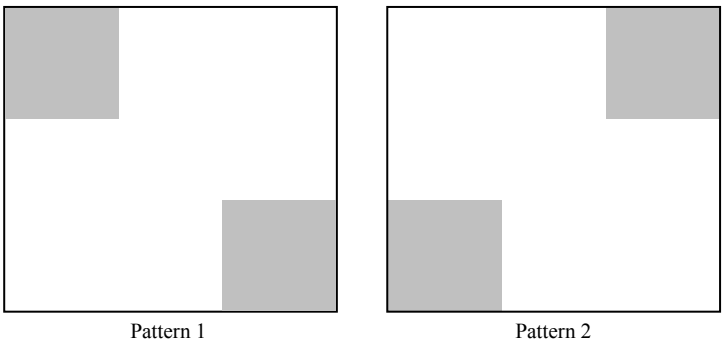

Figure 3.6 The GNG-U preferred dataset is derived from two 2D distributions.

The data is derived from two separate 2 dimensional distributions as shown in Figure 3.6. Pattern 1 is comprised of two distributions and signals are generated randomly from it. At some juncture (depending on the network), the input to the network changes to Pattern 2. This switch provides the non-stationary element of the input space. For the tests given here, the dimensions of each square box were 70 pixels, the height and width of the input space was 230 and only integers were generated. This is in keeping with the research performed by Fritzke (1997b).

The GNG-U algorithm accepted the input signals without transformation. The PSOM required data scaled between 0 and 1. To achieve this, each input signal generated from the above distribution was divided by 230 (the upper limit of the input space).

3.4 Plastic Self Organising Map

3.4.1 Artificial Data Set

The PSOM was tested with the first Artificial Data Set (Figure 3.2) first, to show the network working under normal conditions and then to see how network parameters affect the overall operation. This provides a broad basis for comparisons with other networks and gives some indication of how alterations in network parameters affect the operation of the network.

3.4.2 Test 1: Normal Operation

To show the PSOM working with a variety of initial starting values (both links and reference vectors), 100 tests were run each with random initial weights. The PSOM's parameters (Table 3.2) for the test were found empirically.

Table 3.2: PSOM Test 1 Parameters

Parameter	Value
Node Building Parameter (a_n)	0.29
Clustering Threshold (a_{cl})	0.23
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0.3

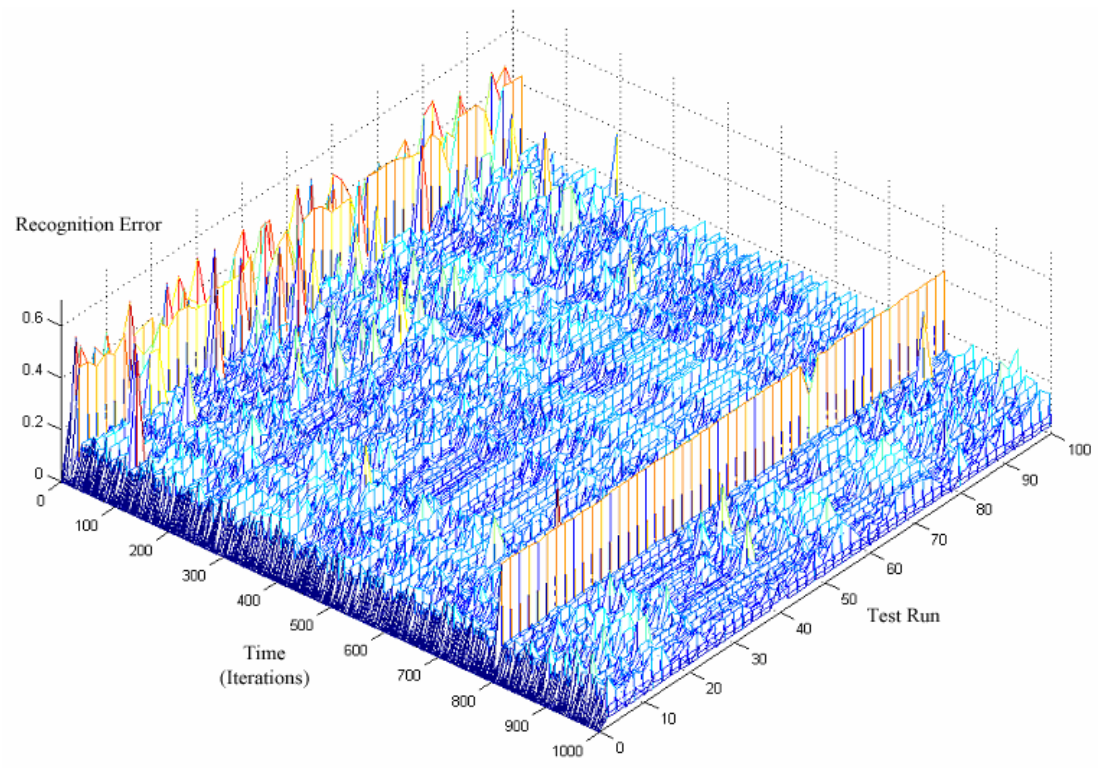

Figure 3.7 PSOM Test 1: The recognition error over time for 100 test runs.

The collated data (Figure 3.7) shows the network operating over the 1000 iterations in 100 tests. Figure 3.7 shows the Recognition error of the network. At iteration 0 and 800 there are large errors. This was expected as there were new patterns shown to the network at iteration 0 (the start of the network's operation) and Class 5 begins at iteration 800. The network behaved in a similar manner through all of the tests, signifying that the error of the network does not depend on the values chosen for the initial weights. However, the network error is strongly affected by the introduction of new pattern classes.

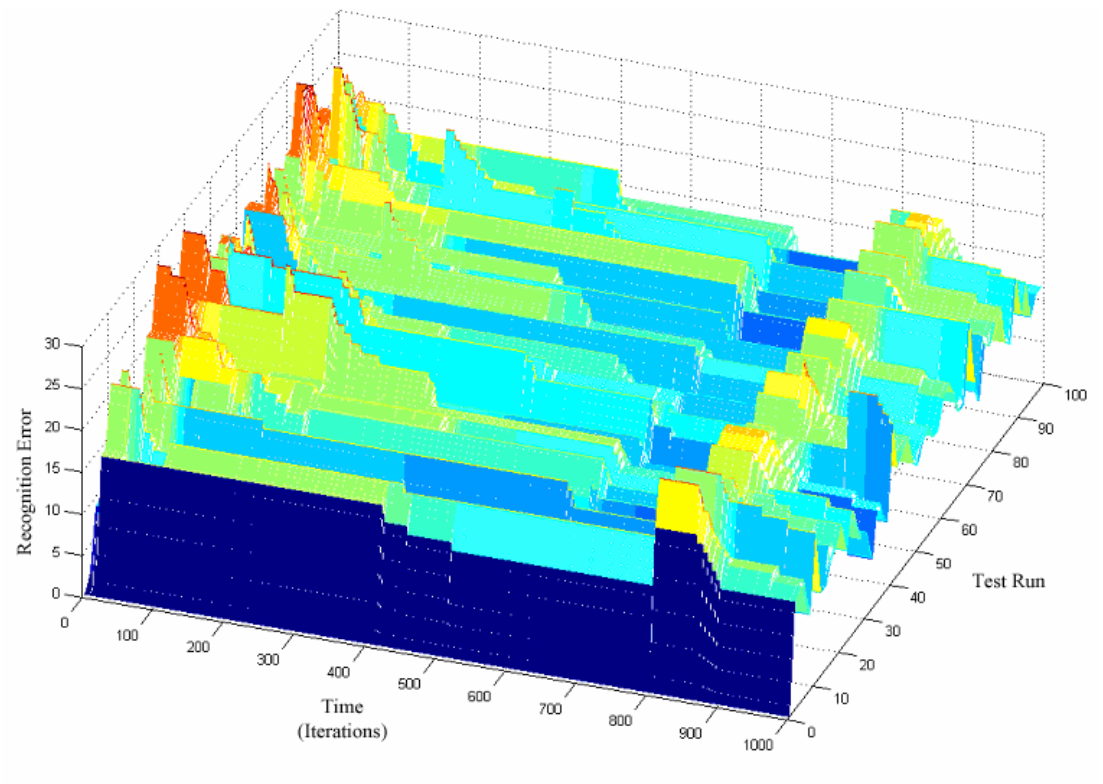

Figure 3.8 PSOM Test 1: The number of links over time for 100 test runs.

Figure 3.8 shows the number of links over time. The number of links is an important parameter as it indicates the size of the network and thus its efficiency. The larger the network is compared to the number of output classes, the less efficient it is. At the start of the network's operation, the number of links is large, as a result of the network growing to accept and learn new input data. Gradually over time, the number of links drops. This is the effect of the ageing process as the network is becoming more efficient by cutting links between neurons associated with dissimilar classes. At iteration 700, the number of links has reached its minimum of 12, for the test run. Even though the network has removed many links, the recognition error of the network (as seen in Figure 3.7 at iteration 700) has not been affected. At iteration 800, Pattern 5 is presented for the first time and the network grows in size to accept the new information. The ageing process quickly removes links which are not relevant for the structure of the data and thus increases efficiency.

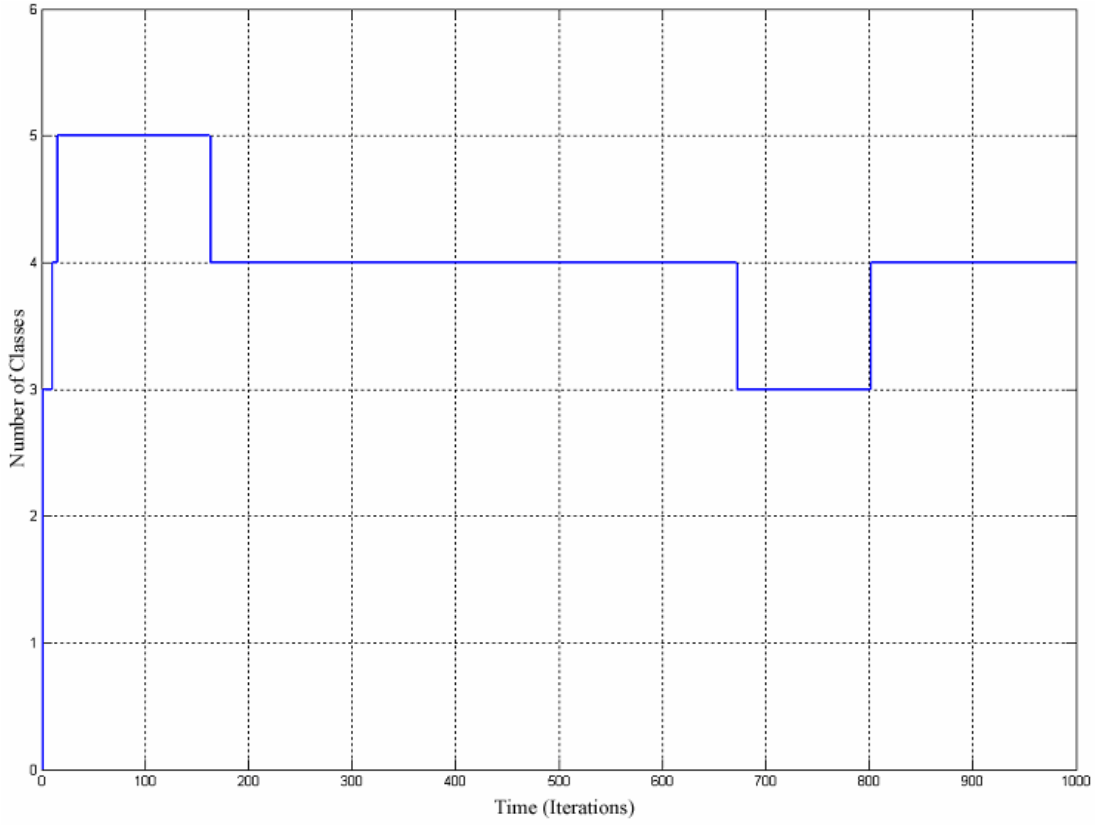

Figure 3.9 PSOM Test 1: Number of Classes over time.

Figure 3.9 shows the number of classes held over time by the PSOM. Initially, the PSOM has five classes, this includes class '0' which is the number associated with the initial neurons. After 180 iterations, this class is removed and the PSOM has the correct number of 4 classes. At 400 the stopping class is removed from the input set; however, this class still exists within the PSOM until time 670. This persistence is desirable for the continued classification of less frequently occurring classes. At time 800, the PSOM creates another class relating to the newly arriving pattern.

Although this graph shows the number of classes over time, it does not show whether or not the PSOM's labels consistently match the labels of the input set. If the PSOM classifies correctly, then one can expect each arrival of a class to match a single PSOM label. This can be displayed in a table of values called a class association table (Table 3.3). The horizontal dimension (row) of the table is the class numbers of the input set and the vertical dimension (column) represents the PSOM class number. The number within the grid is the number of times that this association has been made. For example, input class 2 (vertical column beginning with 2) has been shown

to the network 160 times. Once it was classed by the PSOM as class 0 and the other 159 times it was classed as class 2. This one error occurred when the pattern was seen for the first time and usually occurs because the PSOM class number is derived from the focus that is first. This focus is found before the new neuron group is formed, so the focus is likely to be part of another class. This only occurs once and thus is not seen as a problem. It could be rectified by taking the number of the new group when it is created, instead of the focus. This extra complexity does not add much value to the algorithm as a single incorrect classification is not statistically significant and therefore, the algorithm is left unmodified.

The manner in which the PSOM has classed Input Class Number 2 with PSOM Class 2 is pure coincidence stemming from the order in which the patterns are shown to the network. Table 3.3 shows the PSOM to be classifying correctly as for each column (input class) there is one row (PSOM class) that is predominant. A system that is not classifying correctly would have an even spread of numbers in all rows for each column. This type of behaviour will be shown in later tests.

Table 3.3 PSOM Test 1: The relationship between the input class number and the PSOM class number.

		Input Set Class Number					
		0	1	2	3	4	5
PSOM Class Number	0	1	0	1	0	0	0
	1	0	527	0	1	1	0
	2	0	0	159	0	0	0
	3	0	0	0	95	0	1
	4	0	0	0	0	135	0
	5	0	0	0	0	0	80

3.4.2.1 Network Topologies

The following diagrams show the PSOM at times when it is either in a state of flux (at the start or after the addition of a pattern) or in a settled state. In all of these diagrams, a node is represented by a circle and a link by a line between them. As links get longer, they get lighter. Attached to the centre of each link is a number which is the link length. A red circle indicates the focus. The green number by each node is the class number. This is best illustrated in Figure 3.10, which shows the

initial state for this test. Unlike other SOM algorithms that do show the topology of the input space, the precise 2D locations of the neurons hold little significance. The location of a neuron is only of relevance when compared with other neurons to which it is linked.

Figure 3.10 PSOM Test 1: PSOM initial state showing random links and class numbers.

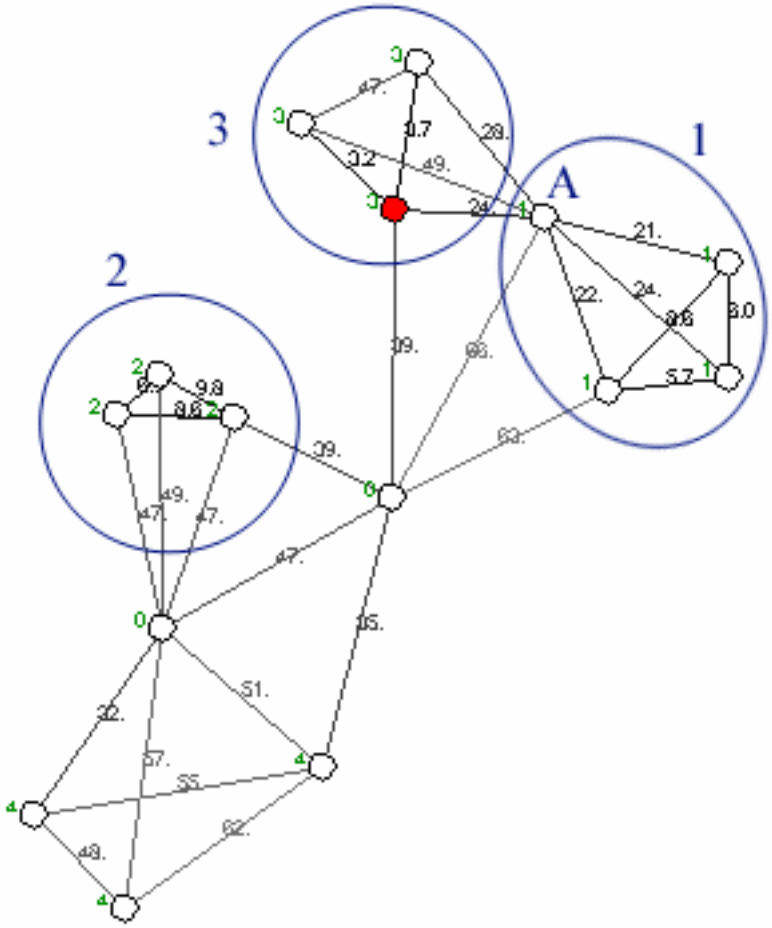

Figure 3.11 PSOM Test 1: PSOM after 21 iterations. The network has grown to learn the new patterns. Easily discernable classes are clearly defined using numbered blue circles and node 'A' represents an initial node of class 0 that was re-classed as class 1.

Figure 3.11 shows the network after 21 iterations. This point was chosen to show the network as its largest. The ageing and neighbourhood functions are in the process of lengthening unwanted links. Even at this early stage, one can discern classes 1, 2 and

3. Class 4 (represented at the bottom of the image) is not yet clearly defined. Node A, part of class 1, was originally an initial node but was then re-classed as class 1 due to its proximity to class 1 in the input space.

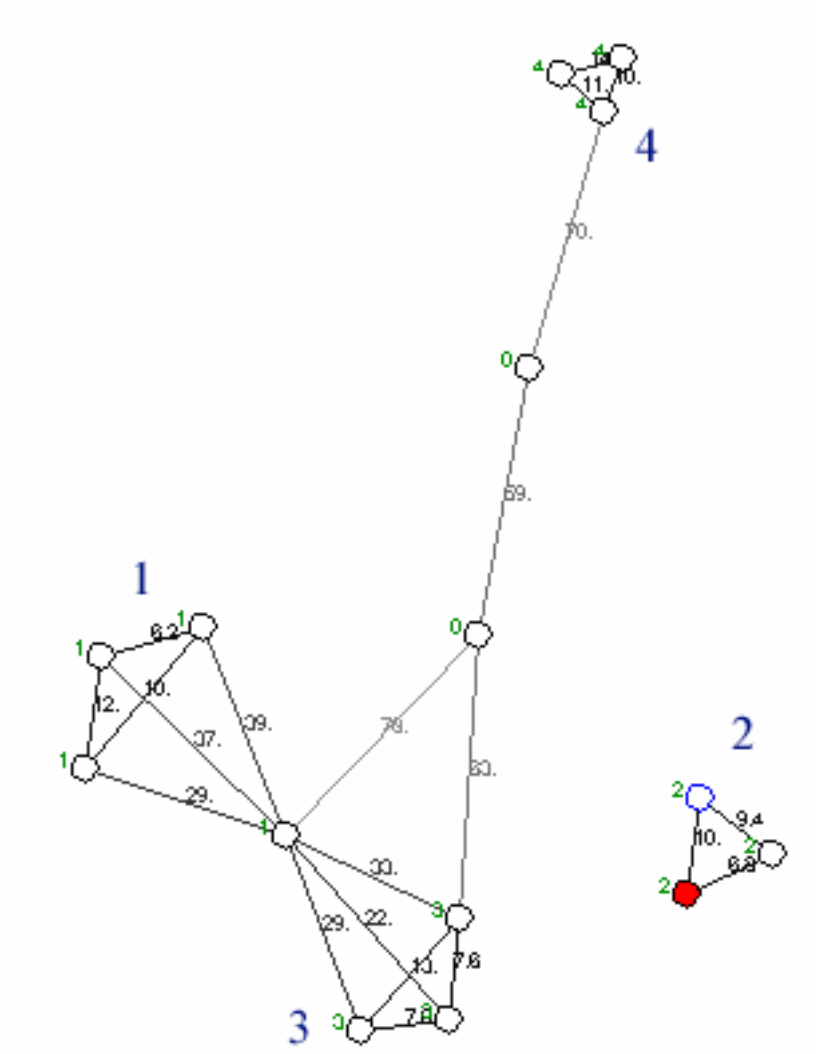

Figure 3.12 PSOM Test 1: PSOM after 62 iterations, the PSOM is in the process of splitting classes apart. The class numbers are repeated in blue for ease of reference.

Figure 3.12 shows the PSOM after 62 iterations. This diagram neatly shows the 3 different stages that the network goes through when splitting up classes. Firstly, class 2 has been completely split from the rest of the network. The small (with short links), separated group indicates that the class it represents is persistent and not near any other class in the input space because no links connect it with the other classes. Secondly, class 4 is in the process of being split off from the main group. The links connected it and the other classes are long and will soon be cut, leaving class 4 in a similar state to class 2. Finally, classes 1 and 3 show classes similar to each other in

the input space. They will eventually break apart but the process is prolonged due to their similarity.

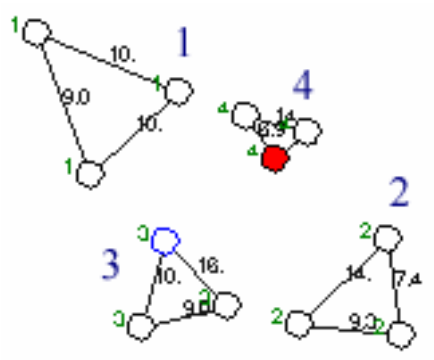

Figure 3.13 PSOM Test 1: The network after 178 iterations. The PSOM has reached something of a stable state where 4 classes exist in the network.

Figure 3.13 shows the PSOM after 178 iterations. At this point, it has seen enough examples of the four input classes to separate the neurons that classify them into distinct groups. The four groups can be clearly seen and this arrangement will persist until either novelty is detected or a class disappears from the input set.

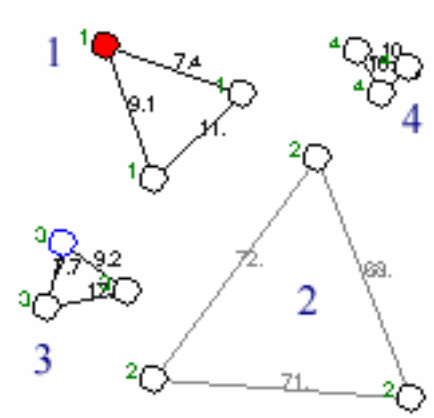

Figure 3.14 PSOM Test 1: The PSOM after 600 iterations showing three persistent classes and one class (2) which has disappeared from the input set.

Figure 3.14 shows the network 600 iterations after class 2 was last shown to the network. As class 2 no longer has the focus, the links are under the sole control of the link ageing parameter and are only going to lengthen until they reach the link removal threshold (90 in this case) and then split. Diagrammatically, links that are close to being cut and removed are shown as being long and light grey; thus group 2

is a large triangle with grey edges, indicating that the group is likely to be soon lost by the network.

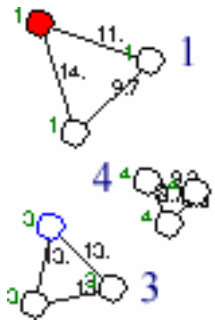

Figure 3.15 PSOM Test 1: The PSOM at 700 iterations, class 2 has been forgotten by the network.

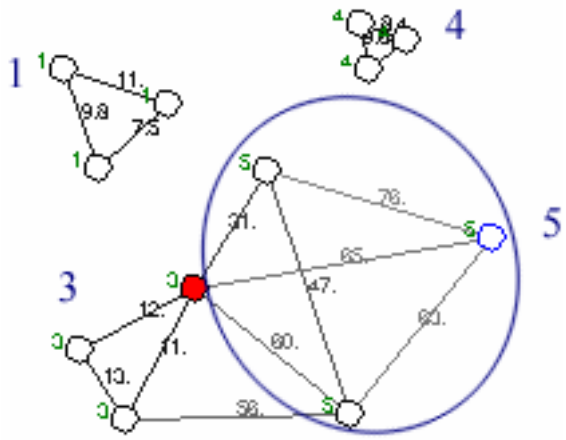

Figure 3.16 PSOM Test 1: The PSOM after 802 iterations. Class 5 has just been shown to the network and the new neurons are attached to class 3.

By 700 iterations, the PSOM has completely removed class 2 (Figure 3.15) and so the remaining 3 classes are held by the network. The next instance of novelty is at 802 iterations, when class 5 is shown to the network for the first time. Figure 3.16 shows the network just after the neurons have been added. The focus (indicated by the red circle) must be an existing neuron because the algorithm needs to know what the best match is before deciding whether any new neurons should be added. Here, class 3 is the focus and the three new neurons have been linked to it and one of its neighbours. The new node group is relatively sparse because it has just been added and the initial link lengths are random.

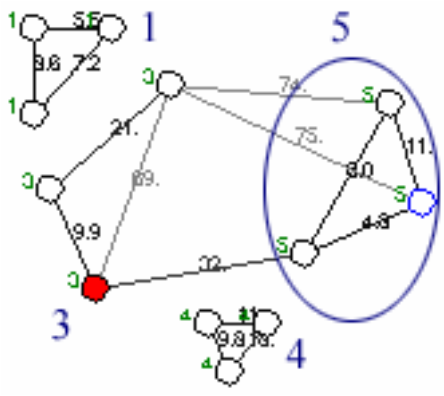

Figure 3.17 PSOM Test 1: The PSOM after 863 iterations. This shows two groups in the process of breaking apart from each other.

Figure 3.17 shows the network 61 iterations after the first class 5 example input has been shown to the network. Classes 3 and 5, although still joined, are in the process of being pushed apart. This would be partly down to the neighbourhood function of the neurons being pushed and partly due to link ageing. Classes 1 and 4 remain unaffected.

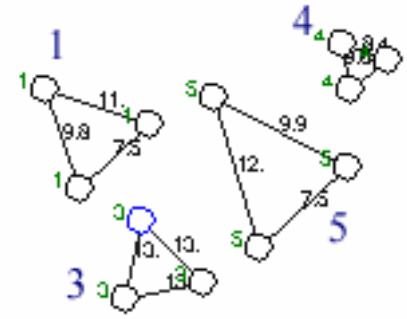

Figure 3.18 PSOM Test 1: The PSOM after 952 iterations. The PSOM in the final state before the test was terminated.

Figure 3.18 shows the PSOM in the last state before the test was terminated. Once again, the PSOM has organised itself into 4 small, stable classes, having forgotten the absent class 2 and accepted the new class 5.

3.4.2.2 Test Summary

Test 1 has shown that the PSOM can alter its topology to accommodate new classes, shrink to remove old classes and classify correctly by assigning arbitrary labels independent of the initial weights. To do this, the PSOM requires three parameters to be set (Table 3.2). Although the input patterns are corrupted by noise, all the input patterns are valid classes in the input space. When a new class is presented to the

network, the error peaks for a short time (1 or 2 iterations). This is expected and desirable as this peaking of error is the main mechanism for the detection of novelty.

3.4.3 Test 2: The Effect of the Node Building Parameter

Intuitively, the Node Building Parameter (a_n) controls the manner in which the PSOM adds neurons. A small value would lead to more neurons and a smaller error between input and focus, whilst a large value would lead to smaller numbers of neurons but a larger error. This is reflected in the results. To show a good range of a_n, a range of 60 test runs was made, with a_n varying from 0.01 to 0.6 in steps of 0.01. The parameters used for this test run are given in Table 3.4.

Table 3.4 PSOM Test 2 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.01 – 0.6
Clustering Threshold (a_{cl})	0.23
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0.2

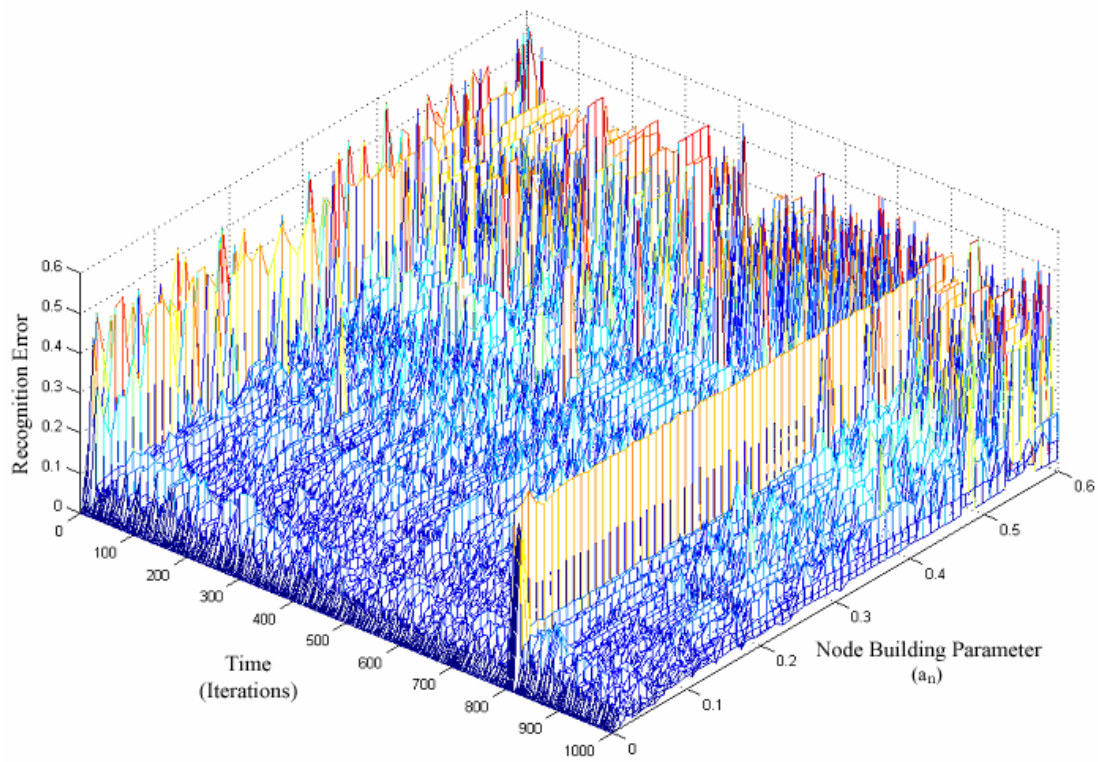

Figure 3.19 PSOM Test 2: The effect of the Node Building parameter on the Recognition Error.

As expected, a_n has a large effect on the error, as Figure 3.19 shows, as a_n increases, the error increases. At a value of approximately 0.4, a_n is too large for the PSOM to create any new neurons and the error becomes very large. Regardless of the value of a_n, there is a large error at iteration 800. This is expected as at this point Class 5 is seen for the first time. Values of a_n larger than 0.6 were briefly investigated but it was found that the error was large as can be seen on the graph as a_n approaches 0.6.

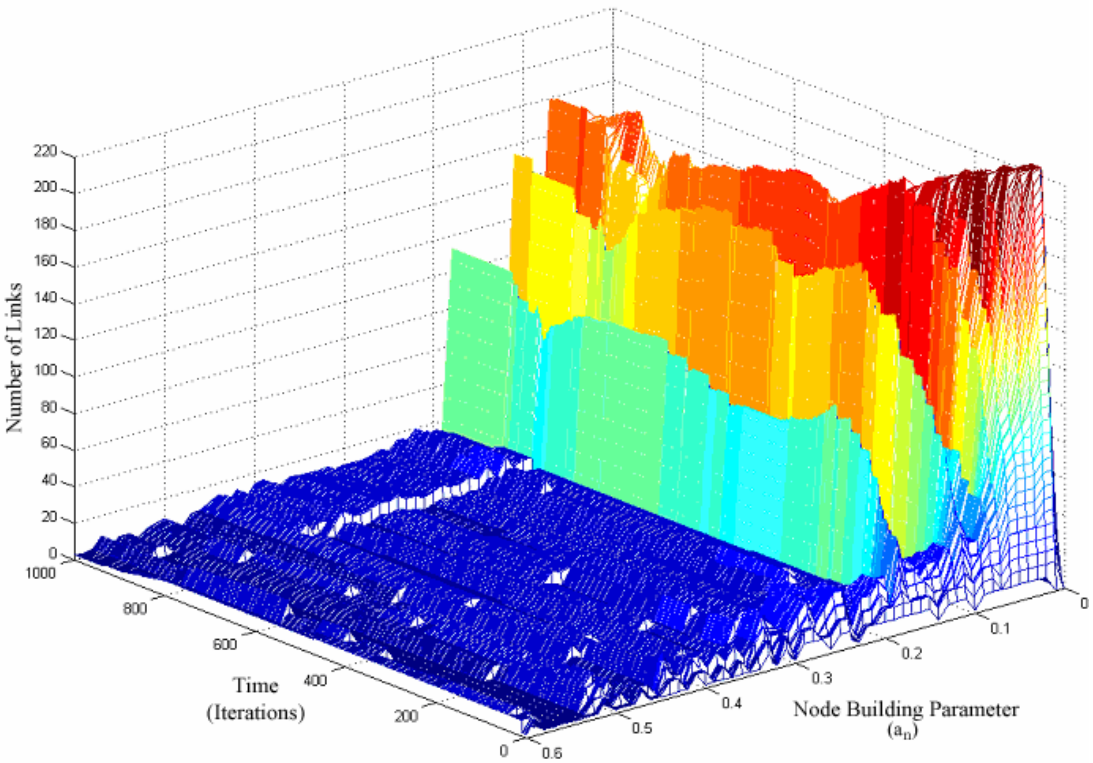

Figure 3.20 PSOM Test 2: The effect of the Node Building Parameter on the number of links. Scaled on the Node Building Parameter and reversed from Figure 3.19.

Figure 3.20 shows the effect of a_n on the number of links. As was expected, a smaller a_n would increase the number of neurons and links added by the network. Figure 3.20 shows this well. When a_n is very small then the number of links remains above 100 for the entire operation of the network. However, at a value of approximately 0.2, the number of neurons required is drastically reduced. This sudden transition is caused by a_n becoming less than the variation of a class. Therefore, new neurons are created for patterns of a single class. This does not have a great effect on the error (as can be seen in Figure 3.19). As a_n becomes large, the number of neurons created is

reduced to a point where the PSOM has no neurons or links at all. A small increase in the number of links required is seen at iteration 800, due to Class 5 being seen for the first time.

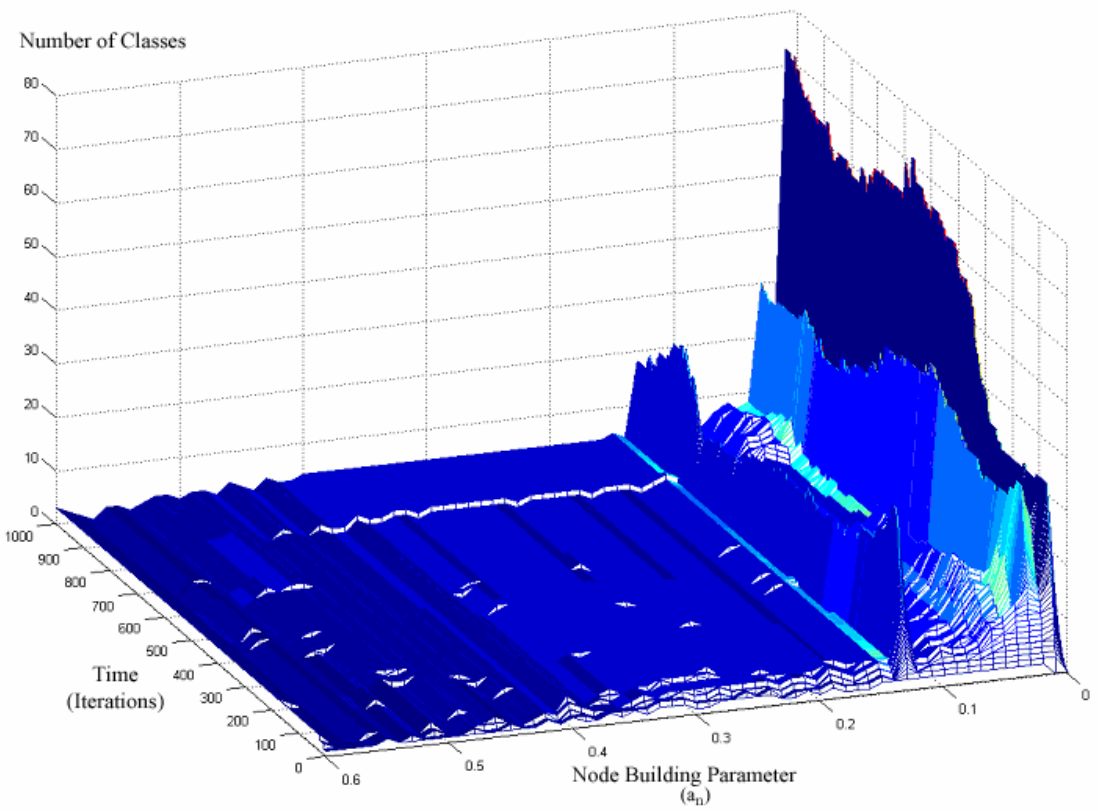

Figure 3.21 PSOM Test 2: The effect of the Node Building Parameter on the number of classes over time.

Figure 3.21 shows the number of classes held by the network at any one time. For small values of a_n, the network is creating many new neurons and with them many new classes. Although the focus will spread its class identification number to the nodes around it, there are simply too many neurons for this to work effectively. As a_n increases, the number of classes found decreases (as does the number of neurons in the network). While a_n is between 0.2 and 0.45, the network classifies correctly, picking out only 5 classes. Beyond 0.45, the PSOM works unreliably as there may not be enough neurons to correctly represent all the data as neurons are not created readily enough.

3.4.3.1 Pathological Network Topologies

Snapshots of the network topologies were taken at the very limits of the PSOM's operation. These show network topologies when the PSOM is not working correctly. Iteration 21 was chosen for comparison with Figure 3.11, which shows the network in the process of growing.

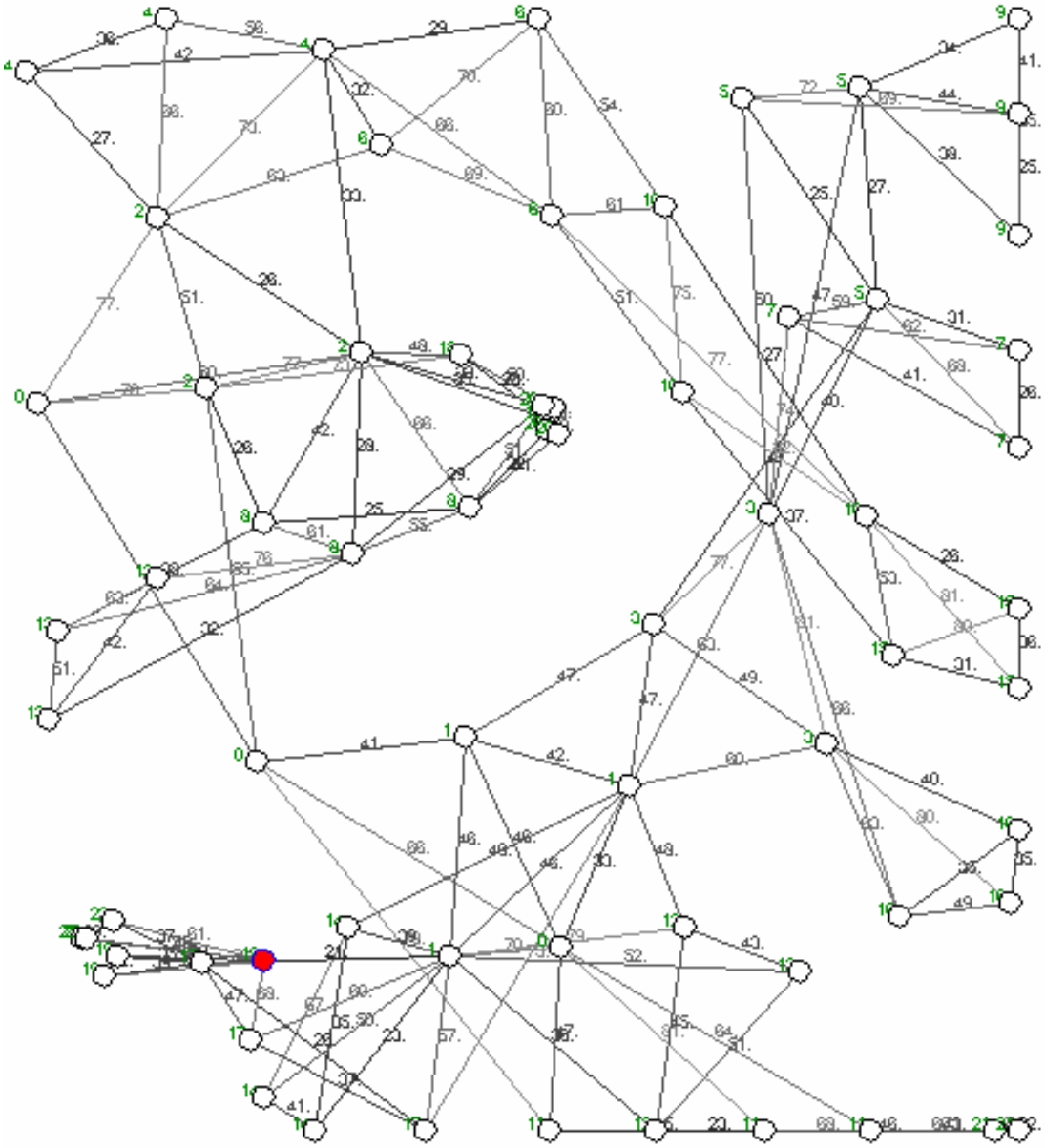

Figure 3.22 PSOM Test 2: The PSOM network after 21 iterations with an a_n of 0. This point is just before the PSOM reaches it programmatic limit of 100 neurons.

Figure 3.22 shows the PSOM when the node building parameter, a_n, is 0. The PSOM will add neurons when the distance between the input and the *focus* is larger than a_n. When a_n is 0, it is rare that any input will have an exact match with a neuron and so neurons are created at every iteration. The network becomes very large very quickly

and hits the programmatic (not theoretical) limit of 100 neurons. If allowed, the PSOM would continue to add neurons until all available memory was consumed.

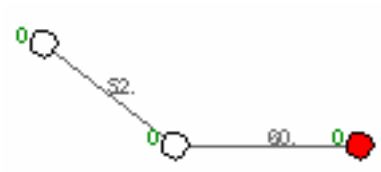

Figure 3.23 PSOM Test 2: The PSOM network after 21 iterations with an a_n of 0.6.

The other end of the scale is shown in Figure 3.23, where no new neurons are added to the network because no distance between the input and a neuron ever exceeds the node building parameter. This leaves the existing (initial) neurons to be updated and moved around the input space given each input pattern.

3.4.3.2 Test Summary

This test has shown the PSOM operating with a range of values for the Node Building Parameter on the Artificial Data set. The operating range of the Node Building Parameter between 0.2 and 0.45 is only of interest to this data set as this value is tuned to the variance of an input class. However this test has shown how the PSOM reacts to an incorrectly set a_n. When the value is small (< 0.14), many neurons are created, the recognition error is small, but the network does not classify well. At large values of a_n, few neurons are created, but the recognition error is high and the network does not produce enough good neurons to correctly represent the number of classes.

3.4.4 *Test 3: The Effect of the Cluster Threshold*

The cluster threshold (a_{cl}) controls whether neighbourhood update is positive or negative. This threshold is compared with the Euclidean distance between the focus and each neuron to which it is connected. If this Euclidean distance is greater than a_{cl}, then the neuron is made more dissimilar and vice-versa. Very small values of a_{cl} will lead to neurons that hold similar patterns being made dissimilar and eventually links will be broken. A large of value of a_{cl} will lead to networks with tight groups of very similar neurons. The parameters used for this test run are given in Table 3.5.

Table 3.5 PSOM Test 3 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.29
Clustering Threshold (a_{cl})	0.01 – 0.6
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0.2

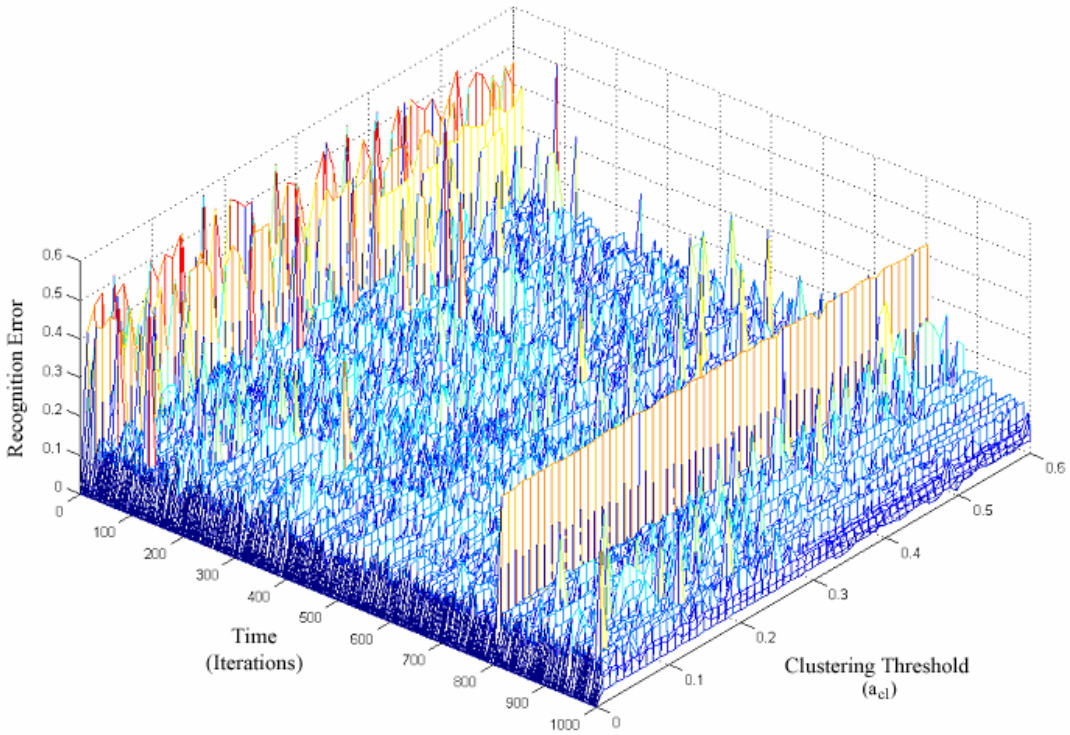

Figure 3.24 PSOM Test 3: The effect of the cluster threshold on the Recognition Error.

Figure 3.24 shows the effect of a_{cl} on the recognition error. As expected, the large peaks of error occur at 0 and 800 iterations, as new patterns are shown to the

network. However, between these values, a_{cl} does not have much effect on the error for this dataset. One might expect many links to be broken, neurons to be lost and subsequently, a large error with a small a_{cl}.

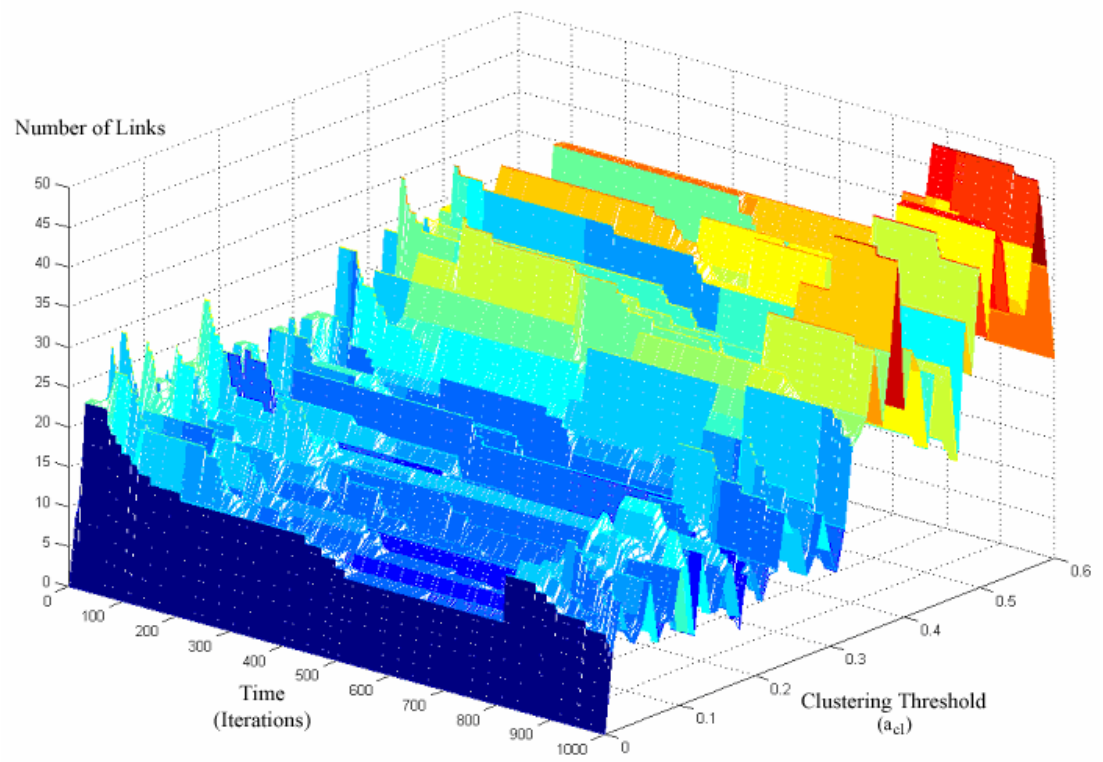

Figure 3.25 PSOM Test 3: The effect of the Clustering Threshold on the number of links.

Figure 3.25 shows the effect of a_{cl} on the number of links in the network. Small values of a_{cl} lead to low number of links as the neighbourhood function will be aiding the ageing process in lengthening links. At large values of a_{cl}, the neighbourhood function pulls its groups together, making them more resistant to the ageing process, thus there are many more links.

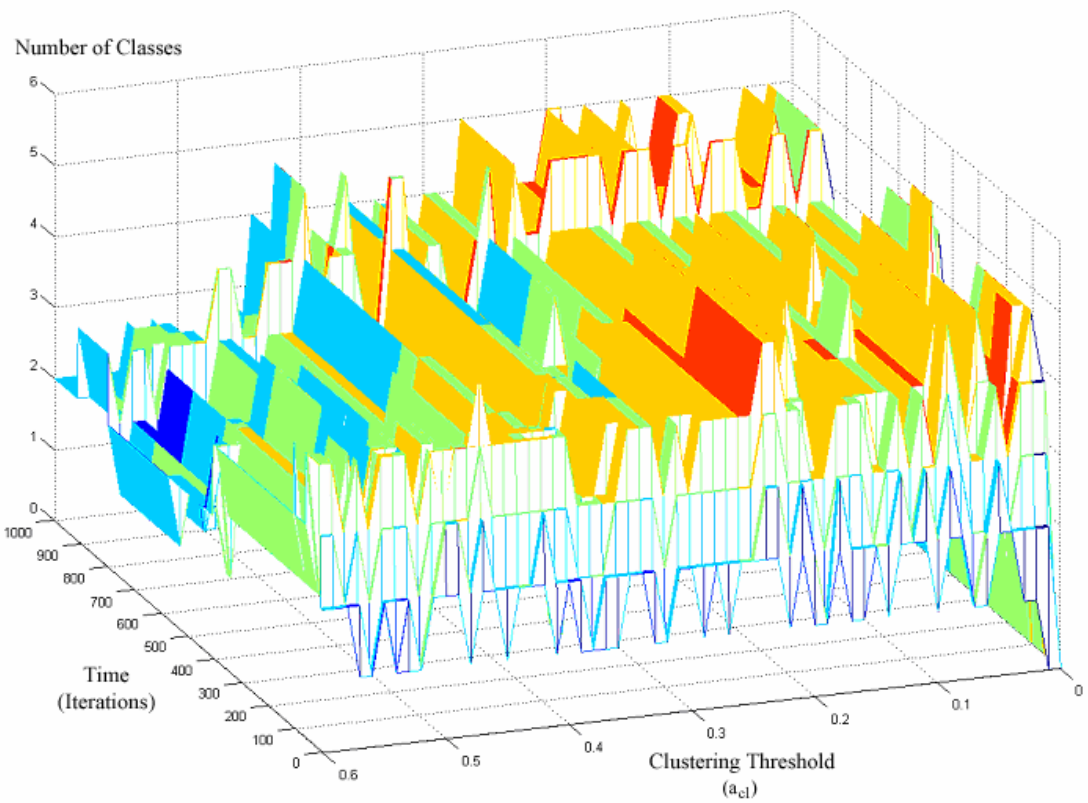

Figure 3.26 PSOM Test 3: The effect of the Cluster Threshold (a_{cl}) on the number of classes held by the network.

Figure 3.26 shows the effect of a_{cl} on the number of classes held by the network. The clustering threshold only really affects the network when its value is large. This is because the class labels are set at neuron creation and when the neighbourhood updates made. When clusters of different classes are heavily connected together, there may be mislabelling at the point when two dissimilar classes are merged into one by a strong neighbourhood function. Thus, when the network comes to count the number of different classes held by the network, there are too few class labels.

3.4.4.1 Pathological Network Topologies

This section gives examples of the network topologies produced at the limits of the clustering threshold, a_{cl}. Figures are given only where the network topology differs from the topologies in Test 1 (Section 3.4.2.1).

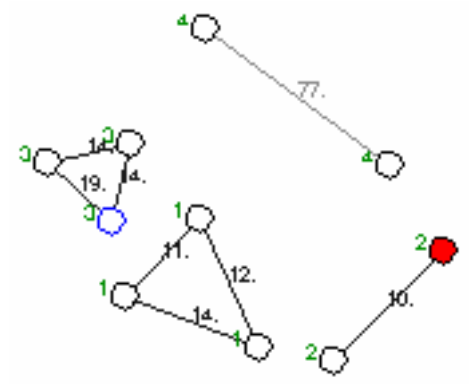

Figure 3.27 PSOM Test 3: PSOM Network after 228 iterations with a clustering threshold of 0.01. Class 4 is on the brink of being lost.

Figure 3.27 shows the network after 228 iterations with a clustering threshold of 0.01, the lower limit of this test. Although the start of the network operation was identical to Test 1, at this point, one can see that the classes are represented with only two neurons, rather than three. This is not surprising as the low clustering threshold, a_{cl}, would mean that even similar neurons would be pushed apart. Class 4, which represents the irregular class (coincidentally, class 4 in the input set) is represented by only two neurons with a long link between them. This class is in danger of being lost – even though it remains in the input space.

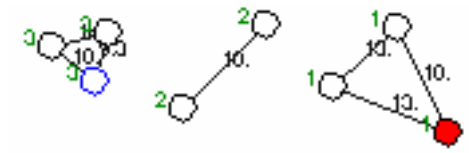

Figure 3.28 PSOM Test 3: PSOM Network after 241 iterations with a clustering threshold of 0.01. Class 4 has been removed from the network.

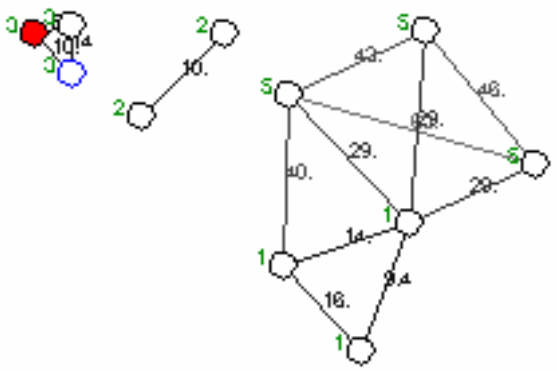

Figure 3.29 PSOM Test 3: The PSOM Network after 257 iterations with a clustering threshold of 0.01. The next presentation of the forgotten class 4 triggers the creation of new neurons.

Class 4 is lost 13 iterations later, as seen in Figure 3.28. This occurred because the irregular input class 4 had a particularly long gap between iterations. If the neighbourhood function, controlled by a_{cl} had been operating correctly, then the group representing class 4 would have been small enough to survive the ageing process. Inevitably, on the next presentation of class 4, the PSOM will be required to relearn the irregular input class anew, assigning it a new class number. This is demonstrated in Figure 3.29. Here, the forgotten class 4 is presented to the network and re-classed as class 5. To an independent observer, this would appear to be a new class, which is in error.

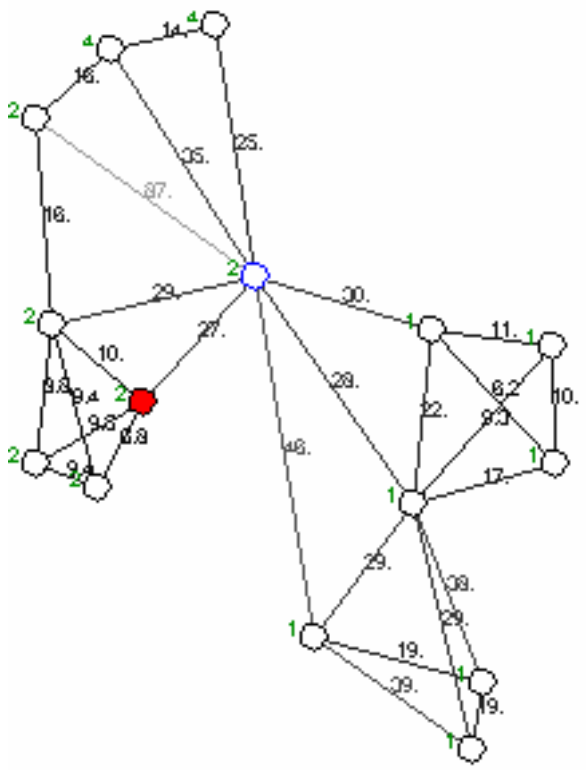

Figure 3.30 PSOM Test 3: PSOM Network after 62 iterations with a clustering threshold of 0.6. Class label '3' has already been overwritten by the more prevalent class 1.

When the clustering threshold is set to the other end of the scale, at 0.6, the neighbourhood algorithm is quick to re-classify neurons and the neighbourhood function will not aid in the separation of classes. Together, these two effects cause large networks and misclassification. Figure 3.30 shows the network after 62 iterations. Compared to when the PSOM was configured correctly (Figure 3.12) in Test 1, one can see that no groups have been split apart and class label 3 has been

completely overwritten by class 1. With such a large a_{cl}, any value of link length will cause the neighbourhood to pull together and so the links connecting dissimilar classes will persist.

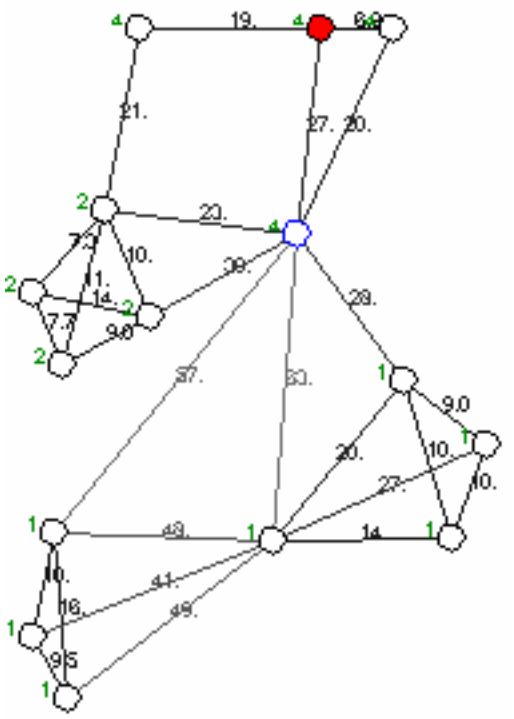

Figure 3.31 PSOM Test 3: PSOM Network after 178 iterations with an a_{cl} of 0.6. Link between dissimilar classes remain.

Even after 178 iterations, the network is still massively interconnected. Compared with normal operation (Figure 3.13), there are no discernable separate groups. This situation persists until iteration 600 where the network looks much the same but with all neurons having the class identifier of 1. With the introduction of class 5 at 802 iterations, the network expands but these new neurons are reclassified as class 1 as well within 30 iterations.

3.4.4.2 Test Summary

Test 3 has shown that the clustering threshold does not affect the error of the network to a great degree. Also, under these conditions, the size of the network remains bounded. Extreme values of a_{cl} cause the network to lose its ability to classify properly and the topographical maps are difficult to interpret.

3.4.5 Test 4: The Effect of Link Ageing

The link ageing parameter (b_a) controls the speed at which links are increased (or aged). This increase of length leads to links being broken and then unlinked neurons being removed. When a neuron is removed, the pattern it represents is lost and will need to be re-learnt. When a link is connected to the focus its length is reset to the Euclidean distance between the two neurons it connects. Intuitively, large values of b_a will lead to information being lost quickly by the network and small values will lead to large, inefficient networks. The parameters used for this test run are given in Table 3.6.

Table 3.6 PSOM Test 4 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.29
Clustering Threshold (a_{cl})	0.23
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0 - 4.0

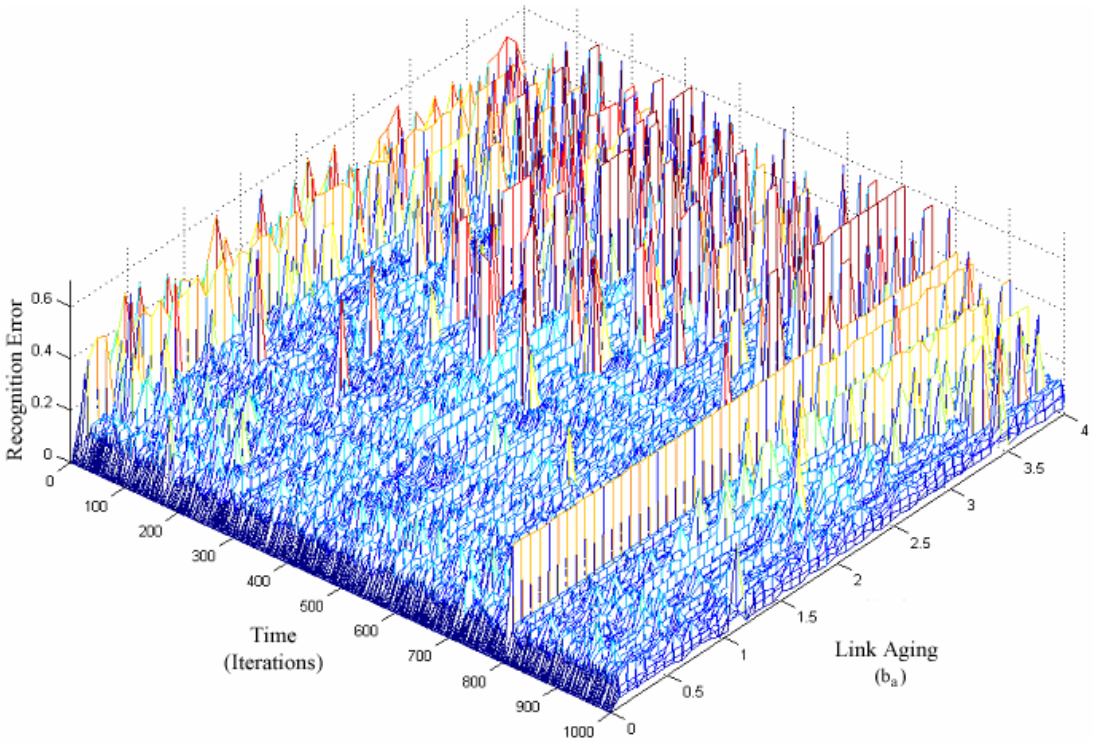

Figure 3.32 PSOM Test 4: The effect of the link ageing parameter (b_a) on the recognition error.

At a value of 0, b_a is not having any effect. Figure 3.32 shows the effect of b_a on the recognition error. With small values of b_a, there is hardly any link ageing, so no

patterns are lost by the network. A b_a of 0 means that patterns are never removed from the network and thus all patterns will be recognised. As b_a grows beyond 2, the network becomes unreliable as classes are quickly lost.

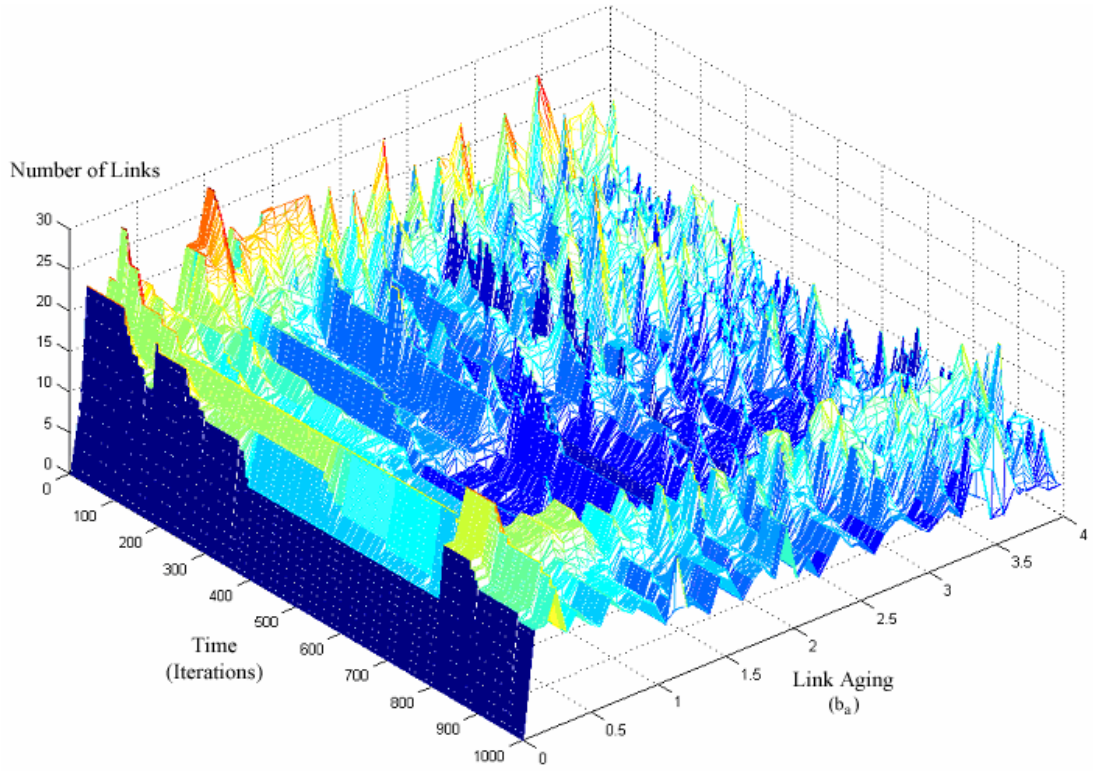

Figure 3.33 PSOM Test 4: The effect of Link Ageing (b_a) on the number of links

Figure 3.33 shows the effect of b_a on the number of links in the network. When b_a is small, link removal only occurs as a result of the neighbourhood function pushing neurons apart. As b_a becomes large, the ageing parameter forces the network to remove links quickly. The result of this is that neurons are removed and persistent patterns forgotten. This can be seen as a large number of peaks as new links and neurons are created to re-learn the forgotten patterns. From this, it can be clearly seen that a smaller b_a leads to fewer links. It is unlikely that the number of links will drop below 3 as the network will keep adding links as new patterns are seen. The creation and removal of the links was found to be a continuous process.

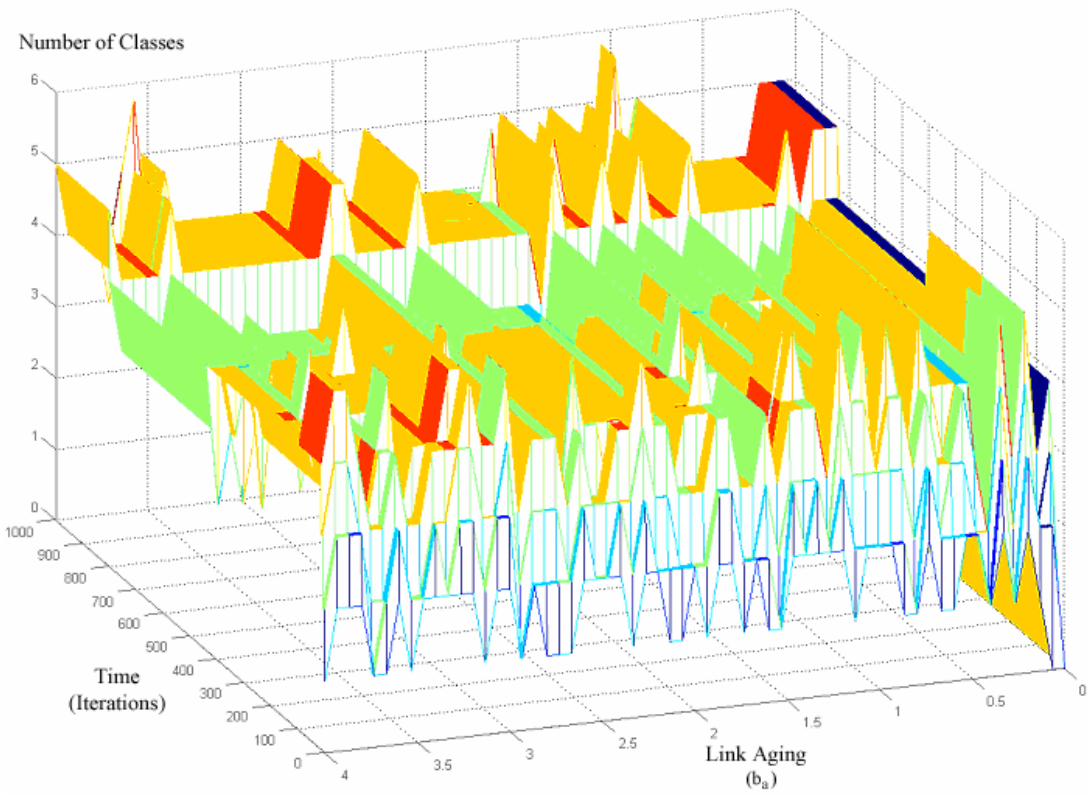

Figure 3.34 PSOM Test 4: The effect of Link Ageing (b_a) on the number of classes held within the network.

Figure 3.34 shows the effect of link ageing on the number of classes held within the network over time. The graph shows that regardless of the value of b_a, the number of classes held by the network remains between 4 and 5. The graph in Figure 3.34 appears to show that the network is classifying correctly, but this is not the case; Table 3.7 shows the class associations between Input Set Class number and the PSOM class number (b_a of 4). As before, the input set class of '0' is an initial class and should be ignored. Table 3.3 shows the correct class association table. There should be one PSOM class assigned to each input class. However, Table 3.7 shows that there are many PSOM classes assigned to a single input class. Input classes 2,3 and 4 suffer from this persistent re-classification by the PSOM. This reclassification can be seen in many row entries (PSOM classes) for a single column (Input classes). This reclassification is because the frequency of the classes is not high enough for the classes to remain persistent in the network between presentations of example patterns. Input classes 1 and 5 are persistent classes and thus the PSOM manages to classify them with some success, even with a b_a of 4.

Table 3.7 PSOM Test 4: The relationship between the input class number and the PSOM class number with a b_a of 4.

		Input Set Class Number					
		0	1	2	3	4	5
PSOM Class Number	0	1	47	10	4	1	0
	1	0	0	0	3	0	0
	2	0	0	1	0	1	0
	3	0	1	84	1	19	0
	4	0	0	0	0	1	0
	5	0	0	0	2	0	0
	6	0	0	0	0	0	0
	7	0	0	0	3	0	0
	8	0	479	0	14	10	0
	9	0	0	0	1	0	0
	10	0	0	0	3	0	0
	11	0	0	0	4	0	0
	12	0	0	1	0	3	0
	13	0	0	64	1	3	0
	14	0	0	0	2	0	0
	15	0	0	0	0	6	0
	16	0	0	0	1	0	0
	17	0	0	0	6	0	0
	18	0	0	0	0	3	0
	19	0	0	0	0	16	0
	20	0	0	0	3	0	0
	21	0	0	0	11	0	0
	22	0	0	0	0	13	0
	23	0	0	0	0	5	0
	24	0	0	0	3	0	0
	25	0	0	0	0	13	0
	26	0	0	0	2	0	0
	27	0	0	0	1	0	0
	28	0	0	0	0	29	0
	29	0	0	0	2	0	0
	30	0	0	0	3	0	0
	31	0	0	0	2	0	0
	32	0	0	0	4	0	0
	33	0	0	0	19	0	1
	34	0	0	0	0	0	80
	35	0	0	0	0	0	0
	36	0	0	0	0	2	0
	37	0	0	0	0	4	0
	38	0	0	0	0	2	0
	39	0	0	0	0	5	0

3.4.5.1 Pathological Network Topologies

With the link ageing parameter, b_a, set to 0, the only process that can lengthen links is the neighbourhood function. Neighbourhood only applies when a link is connected to the focus. Initially, the network appears to operate as in Test 1, with the classes separated. However, by iteration 600, 200 iterations after class 2 was last presented to the network, the PSOM still contains those neurons that represent it (Figure 3.35). Furthermore, the neurons pertaining to the original class, 0, are also held within the network. These neurons will continue to persist as there is no process to affect them.

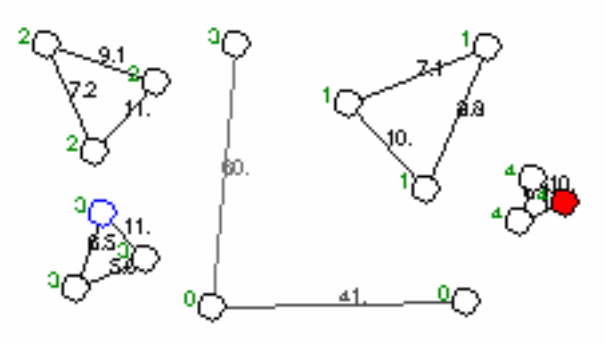

Figure 3.35 PSOM Test 4: PSOM after 600 iterations with a b_a of 0. The network has not managed to remove the neurons pertaining to the now defunct class 2.

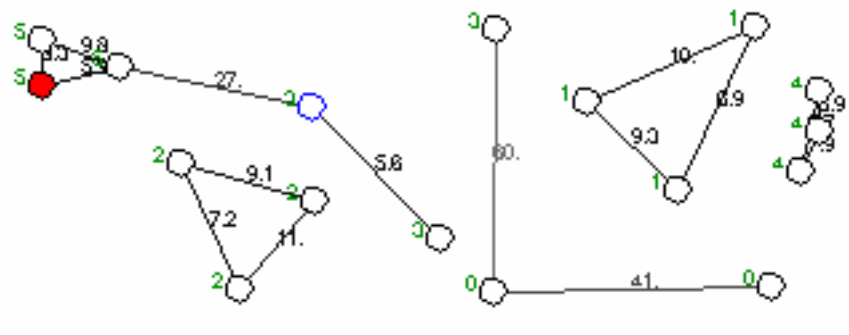

Figure 3.36 PSOM Test 4: PSOM after 952 iterations with a b_a of 0. The network continues to learn but does not remove redundant neurons.

This problem is further illustrated in Figure 3.36, where the network should only contain four distinct classes. This is not the case; the network has managed four distinct classes *and* two classes still joined together. Classes 0 and 2 still persist redundantly and now the link ageing process cannot be used to help the splitting of classes 3 and 5.

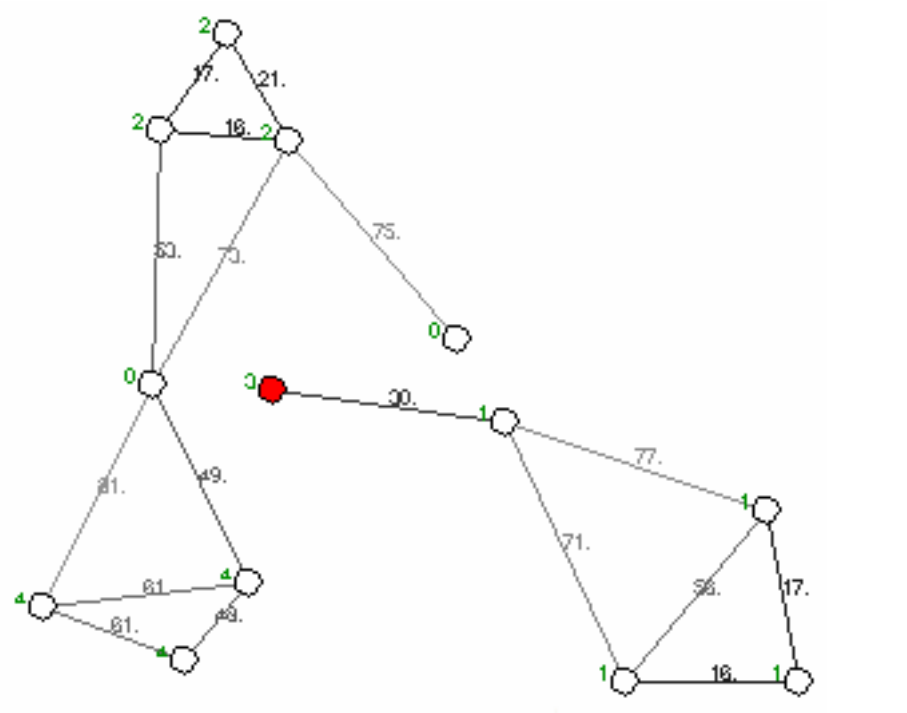

Figure 3.37 PSOM Test 4: PSOM after 21 iterations with a b_a of 4. Only classes 1 and 2 manage to persist in the network.

Figure 3.37 shows the other end of the scale, when the link ageing is 4. In this configuration, only the two very regular and persistent patterns can persist in the network. The other two, less regularly occurring classes are quickly lost and re-learnt. This has the effect of the same classes being classified by more than one neuron. This situation persists throughout the lifetime of the network.

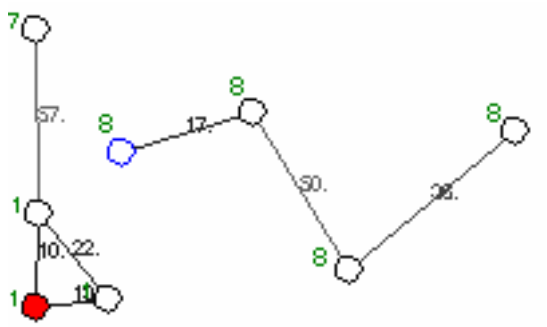

Figure 3.38 PSOM Test 4: PSOM after 952 iterations with a b_a of 4. Only class 1 has been continually successfully represented.

Figure 3.38 demonstrates that a high b_a can make even persistent patterns lose neurons. In this diagram, class 8 represents class 5 in the input space. Of the typical group of 3 neurons and 5 links that is normally used by the PSOM to represent a class, it has already lost one link. Class 1, however, has managed to persist

throughout the lifetime of the network and this is largely due to the high density of the class patterns in the input space.

3.4.5.2 Test Summary

Test 4 shows the effects of the link ageing parameter (b_a) on the operation of the network for a non-noisy artificial data set. When b_a is 0, no information is lost by the network as the links are not aged. Thus, the network structure is purely a spatial one. However, at these values, redundant neurons and links are not removed from the network as it is only the neighbourhood function that can cut links and this only occurs around the focus. As the data is noise free, the number of links is small with small values of b_a. As b_a increases, links are removed more quickly from the network, making the network more efficient. When b_a is large (>2), the network continues to recognise the right number of classes, but the classes are often reclassified. This reclassification is not desirable. For noiseless systems, very small values of b_a are advised.

3.4.6 Test 5: Noise Rejection Capabilities

The artificial data set shows the operation of the network under idealised conditions. However, the test described in this section, using the noisy version of the data set (Figure 3.4) was designed to investigate the effect of noise on the PSOM. This type of noisy data is more representative of a real world data set. The PSOM was tested with 100 test runs using the parameters in Table 3.8, which were found empirically and by using the results of previous tests. The link ageing parameter needs to be increased for noisy data sets because neurons will be created to represent this noise and those neurons should be rejected. In real world systems, the noise level can be assessed using a Signal to Noise ratio. As this value decreases, a higher link ageing parameter will be needed. The neurons pertaining to noise are unlikely to become the focus very often and so their link lengths will only change as a result of link ageing. Thus, by setting the link ageing parameter to have a higher value, the noise rejection properties of the network have been improved.

Table 3.8 PSOM Test 5 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.29
Clustering Threshold (a_{cl})	0.23
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	2

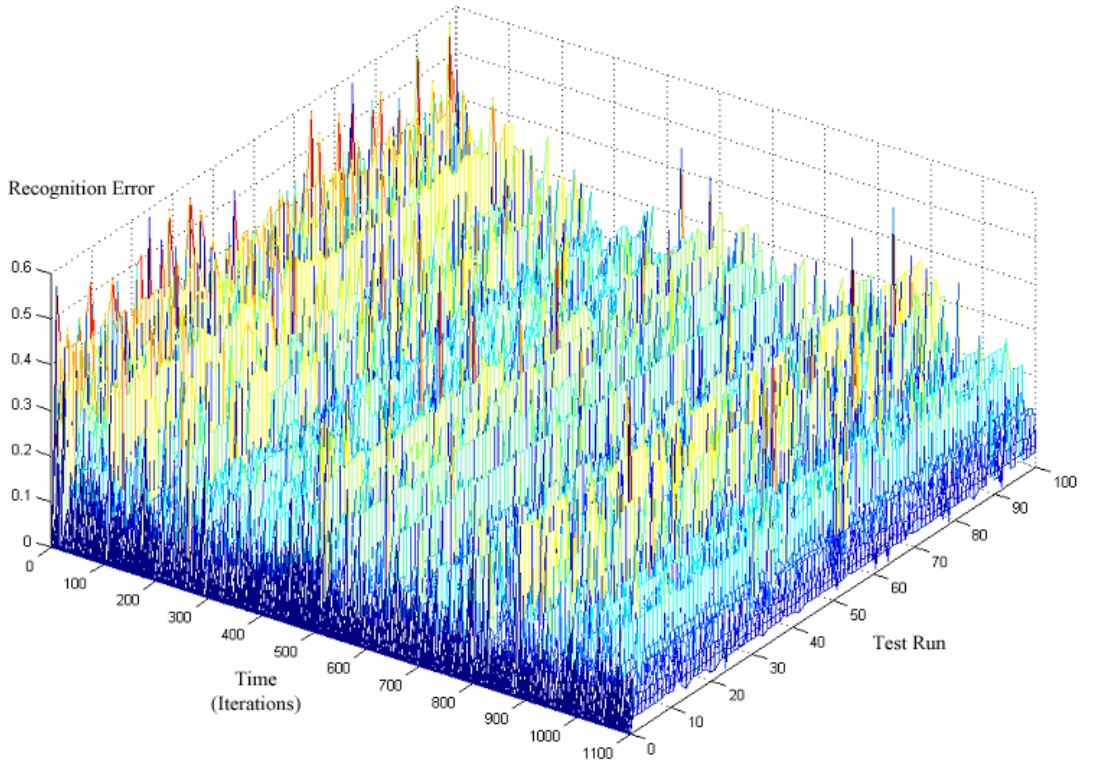

Figure 3.39 PSOM Test 5: The recognition error over time (noisy artificial data set).

Figure 3.39 shows the Euclidean distance (error) over time for 100 runs. Unlike the clean data set (PSOM Test 1, Figure 3.7) there are many spikes between the initial onset of patterns and iteration 800 when Class 5 first appears. Each of these noisy patterns is treated as a new class and new neurons are created. As the network learns all new data, it also learns the noise pattern. However, the class created by this noise pattern will soon be lost, leaving the more persistent patterns. To aid this process, a large link ageing parameter is used to remove those neurons pertaining to noisy patterns.

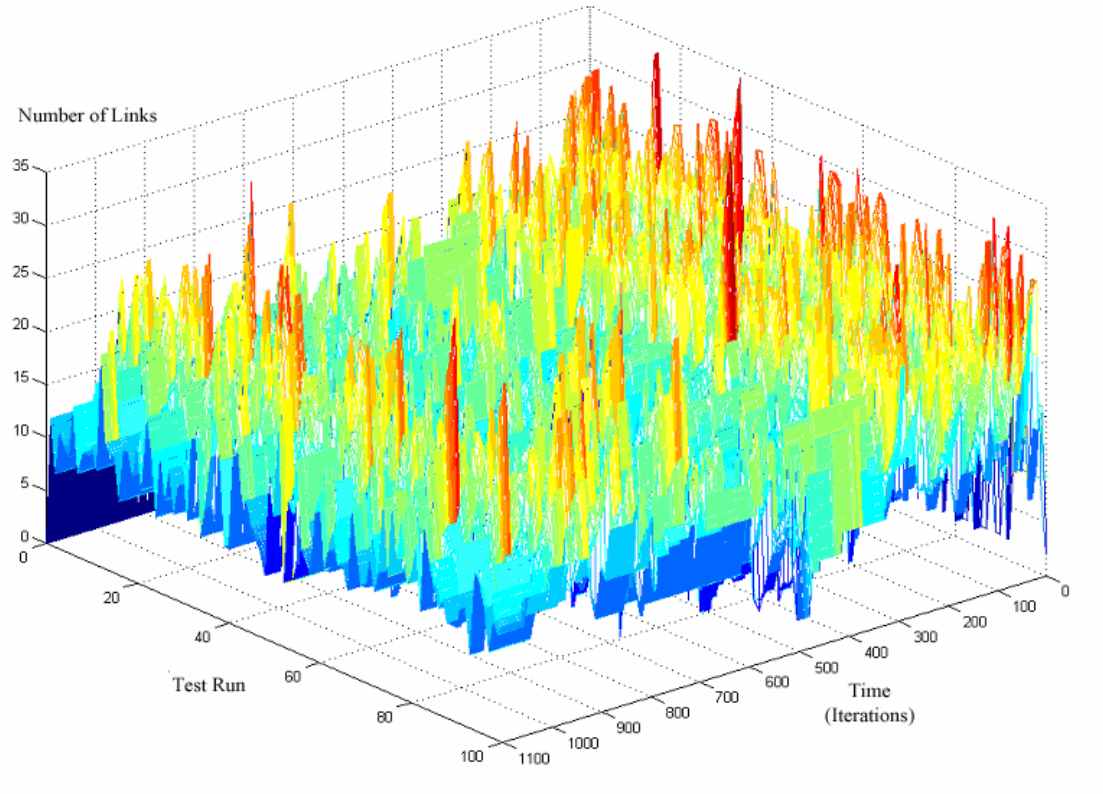

Figure 3.40 PSOM Test 5: The number of links over time (noisy artificial data set).

Figure 3.40 shows the number of links over time. With noise patterns and a high ageing process, the number of links appears to 'spike' repeatedly. This is due to new neurons being created by noise patterns and then removed shortly after by the ageing process. Only those neurons associated with persistent classes are withheld by the network. This high link ageing also suggests that the temporal persistence of class 2 (which disappears after 200 iterations) is decreased: the neurons are quickly removed from the network after the presentation of the last input pattern for this class.

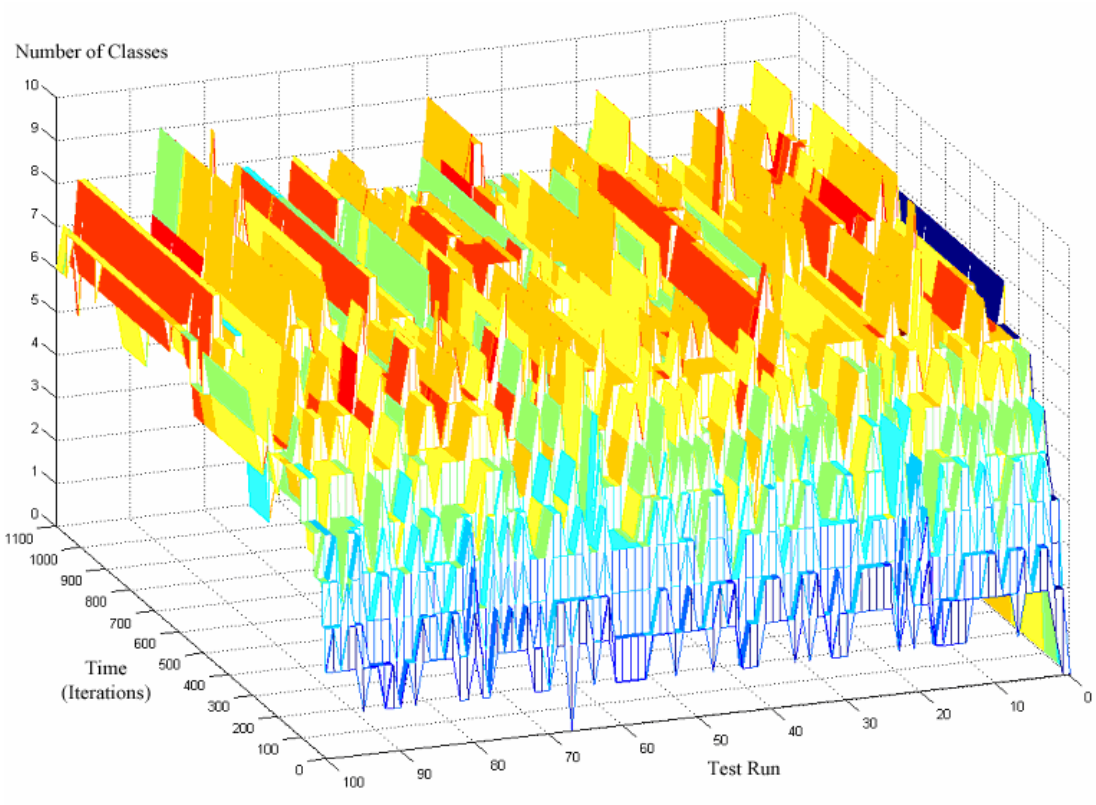

Figure 3.41 PSOM Test 5: Number of classes over time (noisy artificial data set).

Figure 3.41 shows the number of classes held by the network over time for each test run. The number of classes found by the network is bounded for all time. However, this does not show whether or not the desired classes are classified properly. Table 3.9 shows the relationship between the input set class number and the PSOM class number. Remember that some of the noise classes appear within the same space as the given classes and that it is not expected that the PSOM will distinguish between the desired classes and the error class in this case. The noise class is column 0 and columns 1 through to 5 are desired classes. It can be seen that classes 1,2,4 and 5 have been classified as well as before. However input class 3 has been split by the PSOM into two classes. Input class 3 is the infrequent, regular class and is not frequent enough always to be classified as the same class by the PSOM. This splitting of class is a rare case and suggests that the PSOM's parameters are not tuned to the data set correctly. A more likely explanation is that this problem is caused by the reduction in density of input class 3 in comparison to the rest of the data set. Class 5 appears to have fewer patterns but it is learnt easily because it is frequently seen by the network.

Table 3.9 PSOM Test 5: The relationship between the input class number and the PSOM class number (noisy artificial data set).

		Input Set Class Number					
		0	1	2	3	4	5
PSOM Class Number	0	12	526	1	1	1	0
	1	17	0	159	0	0	0
	2	9	0	0	94	0	0
	3	8	0	0	0	135	0
	4	0	0	0	0	0	0
	5	2	0	0	0	0	0
	6	10	0	0	0	0	0
	7	17	0	0	0	0	0
	8	9	0	0	0	0	0
	9	0	0	0	0	0	0
	10	9	0	0	0	0	0
	11	2	0	0	95	0	0
	12	6	0	0	0	0	81

3.4.6.1 Pathological Network Topologies

When perturbed by noise, the network topology gains additional neurons that it would otherwise not have. Also, the topology is less compact, with neurons representing noise patterns being kept for the sole reason that they are regularly connected to a focus.

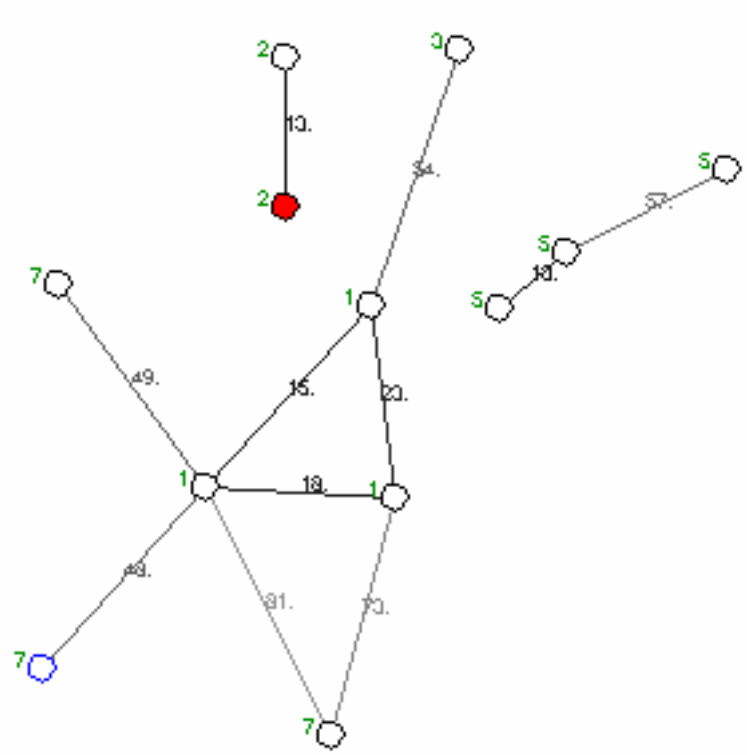

Figure 3.42 PSOM Test 5: PSOM Network after 230 iterations.

Figure 3.42 shows an example of the PSOM network after 230 iterations, each run of the network is likely to give differing patterns, of which this is an example. In this figure, input classes 1 and 2 (the persistent classes) are classified as network classes 1 and 2. Irregular class 4 is classified as class 3 and input class 3 as PSOM class 5. Class 7 is a noise class and is only still connected to the network because it has previously been connected to a persistent class (input / PSOM class 1). This diagram is a good example of the topologies achieved in a noisy environment. Persistent classes are represented as small clusters and irregular inputs are clustered as large groups. These infrequent inputs are more prone to being forgotten and re-learnt by the network because the links of infrequent classes are more susceptible to being lost through link ageing. Noise is represented by neurons attached by long links to other classes.

3.4.6.2 Test Summary

This test shows that the PSOM can differentiate persistent classes from noise classes while using a bounded number of neurons and links. This test also shows how known classes can be classed as two different PSOM classes when the temporal density of the input class is not large enough given the noise level.

3.4.7 Test 6: The Effect of Node Building Parameter on noisy data

This test is a similar to Test 2, except that the data set which is presented to the PSOM is the noisy version of the artificial data set. As the input data set is corrupted by noisy patterns (those patterns which are not part of a class) one would expect a larger error during the operation. In test 2, it was shown that a small value of a_n led to large number of links but a small error. It is expected that this should still be the case even when there are noisy patterns. No topographical maps were taken for this test because the maps were similar to those of Test 2, the only difference being the addition of noise neurons as with Test 5. The parameters used are shown in Table 3.10.

Table 3.10 PSOM Test 6 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.01 – 0.4
Clustering Threshold (a_{cl})	0.23
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0.2

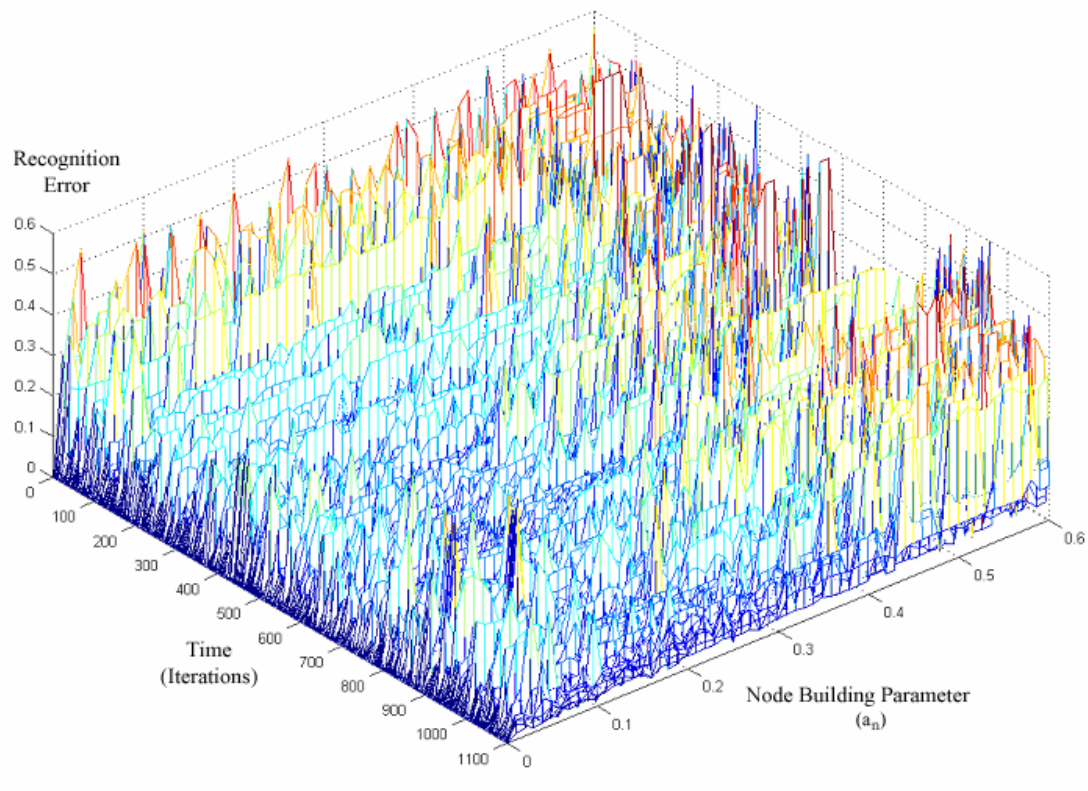

Figure 3.43 PSOM Test 6: The effect of the Node Building Parameter (a_n) on the recognition error (Euclidean distance).

Figure 3.43 shows the effect of a_n on the Euclidean distance (error). With small values of a_n, the network has a lower error. As a_n increases, the network is not creating enough neurons to cope with the new information. The overall error output of the network is greater with a corrupted data set. Also, the large error at 800 iterations is not prominent as it is in the uncorrupted data set in Figure 3.19. This is because the network is being perturbed by random noise and the arrival of a new class is not such a distinct event.

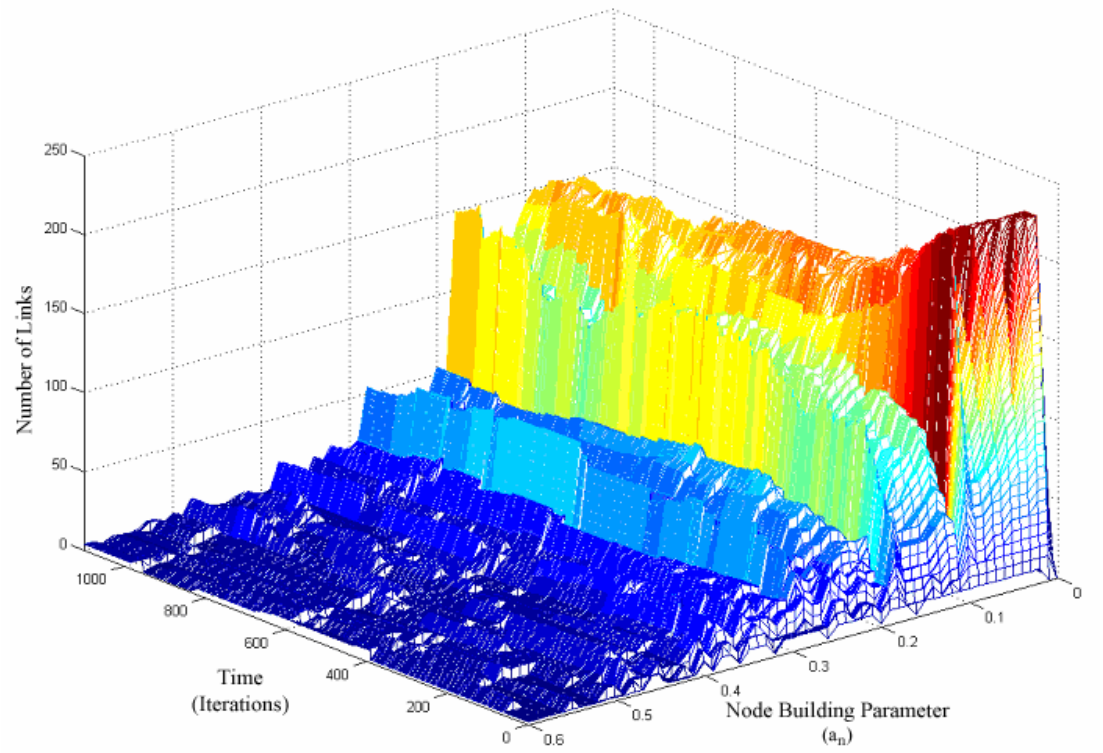

Figure 3.44 PSOM Test 6: The effect of the Node Building Parameter (a_n) on the number of links (noisy data set).

Figure 3.44 shows the effect of a_n on the number of links. As expected, a small value of a_n led to a large network and vice-versa. The uncorrupted data set yielded a similar graph, except that in this case, the maximum number of links is 270 (capped in Figure 3.44 for graphical reasons) whereas without the noise patterns, the maximum number of links was 220. This is expected as there are more perturbations spread across the input space in this test and thus more neurons are required to represent it.

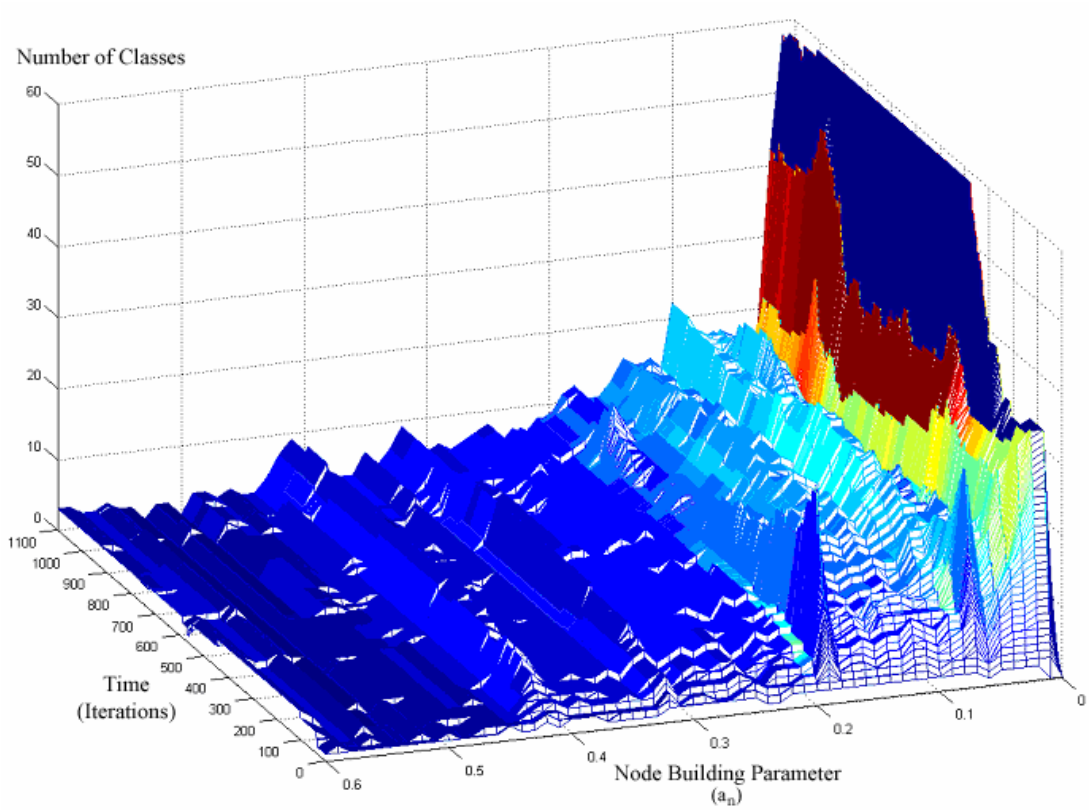

Figure 3.45 PSOM Test 6: The effect of the Node Building Parameter (a_n) on the Number of Classes (noisy data set).

Figure 3.45 shows the effect of a_n on the number of classes classified by the network. For small values of a_n, the network sees every input as a different class and thus classifies them separately. For large values of a_n, the network will classify many different input classes as the same class. In the middle region, the PSOM uses an acceptable number of classes, i.e., there are the desirable 5 classes and then a number relating to the noise data. Without creating a class relationship matrix for each value of a_n, it is difficult to gauge how well the PSOM classifies as a_n is altered. Also, these relationship matrices would be associated with only this data set and would not necessarily be correct for another data set. A short empirical study showed that at small and large values of a_n the network failed to classify correctly.

3.4.7.1 Test Summary

This test has shown that the effect of the node building parameter, a_n on the operation of the network for a noisy data set is similar to its effect on the operation of the network for a noiseless data set (PSOM Test 2). In this case, the number of links and classes is larger as the network generates more neurons and links to cope with the noisy data. However, in the presence of noise, the network still responds in a predictable manner which is a positive result when the parameters will have to be tuned to an unknown data set. The range of Node Building parameter that was found to classify correctly was found to lie between 0.2 and 0.3.

3.4.8 Test 7: The Effect of the Cluster Threshold parameter on noisy data

This test mimics Test 3 (Section 3.4.4), in that it shows the effect of the Cluster Threshold, a_{cl}, on the PSOM. As previously noted, a small a_{cl} means that neurons are easily pushed apart in the neighbourhood function, allowing better clustering. A large a_{cl} implies the opposite; the neighbourhood function pulls dissimilar neurons together, countering the effect of link ageing more efficiently. This leads to larger networks. No topographical maps were taken for this test because the maps responded were similar to those of Test 3, the only difference being the addition of noise neurons as with Test 5. The parameters used can be seen in Table 3.11.

Table 3.11 PSOM Test 7 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.2
Clustering Threshold (a_{cl})	0.01 – 0.6
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0.2

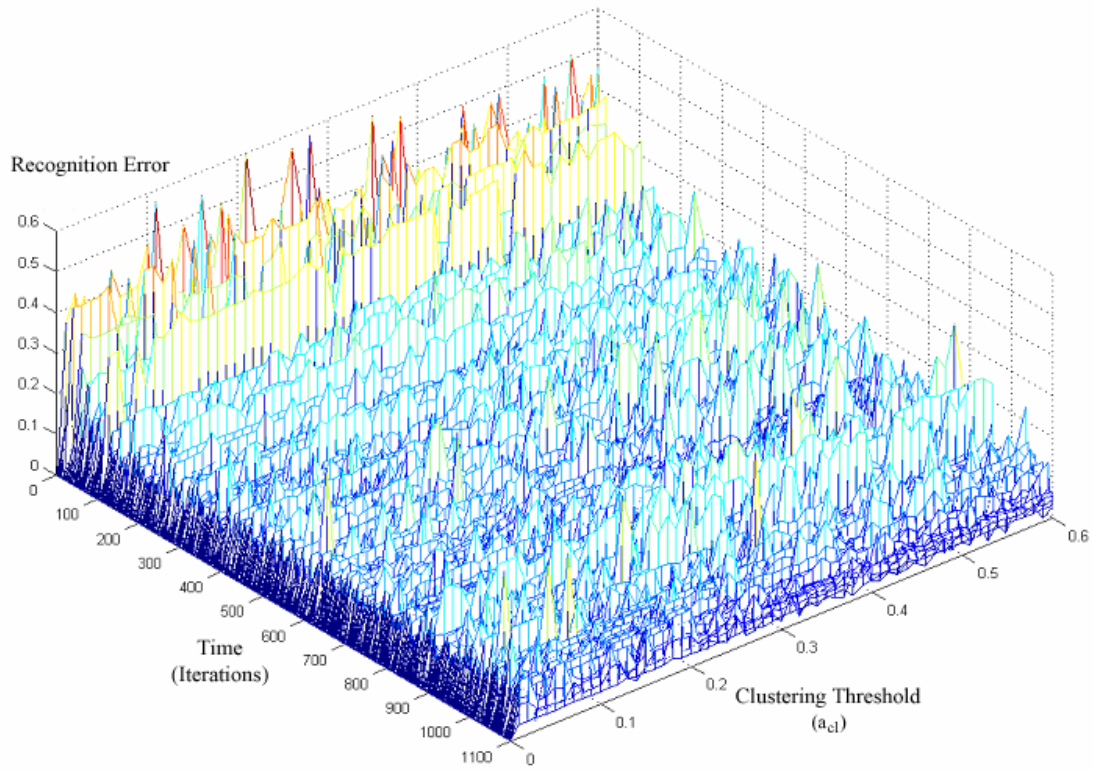

Figure 3.46 PSOM Test 7: The effect of the Clustering Threshold (a_{cl}) on the recognition error (Euclidean distance).

As with Test 3, a_{cl} has very little effect on the error (Figure 3.46). As a_{cl} approaches 0, the error of the PSOM does increase by a small amount. This is due to the neighbourhood function breaking links between similar neurons of the same class, losing neurons and then requiring new neurons on the next presentation of that class. As a_{cl} approaches 0.5, the neighbourhood function is strengthened and so makes dissimilar nodes more similar. The noisy patterns within the input set perturb this process by modifying the neurons associated with valid classes to become associated with the noise.

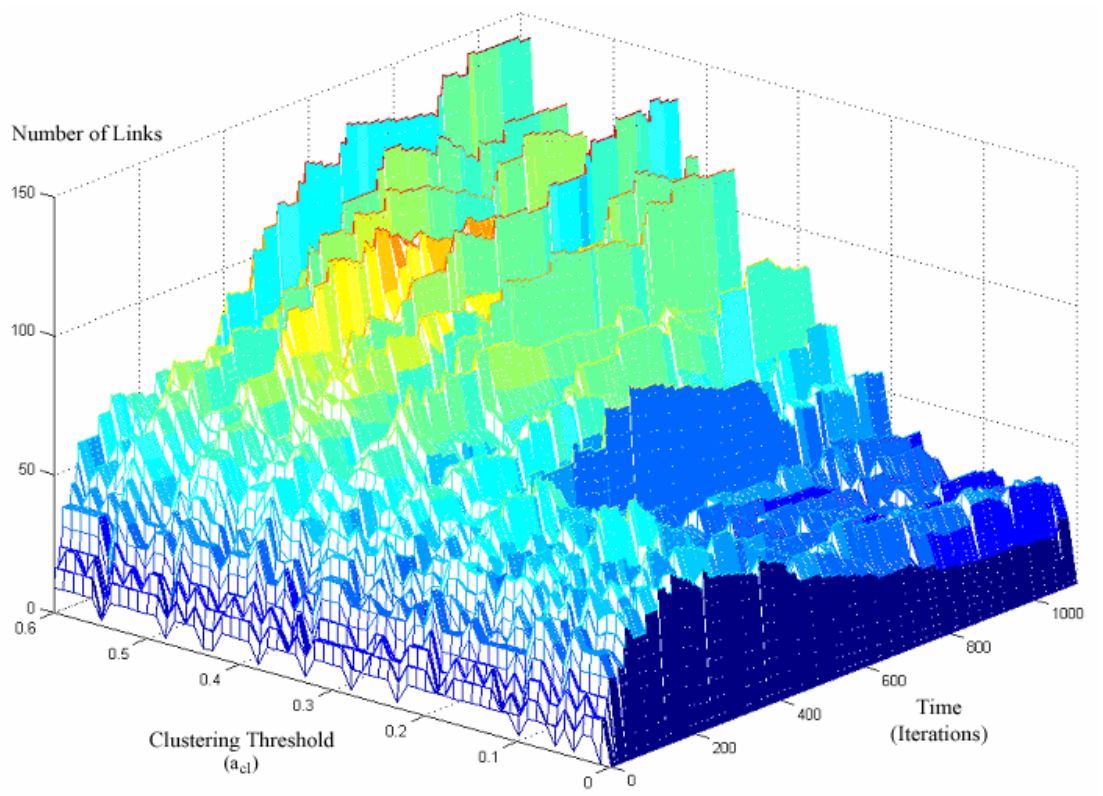

Figure 3.47 PSOM Test 7: The effect of the Clustering Threshold on the number of links.

Figure 3.47 shows the effect of a_{cl} on the number of links in the network. As in Test 2, small values of a_{cl} helps the link ageing process and thus the network size remains small. As a_{cl} is increased, the neighbourhood function is working against the link ageing process and the number of links is increased with every pattern. This retention of links and neurons suggests that the network is learning the noisy patterns, instead of rejecting them. Thus, a_{cl} should remain small for noise rejection.

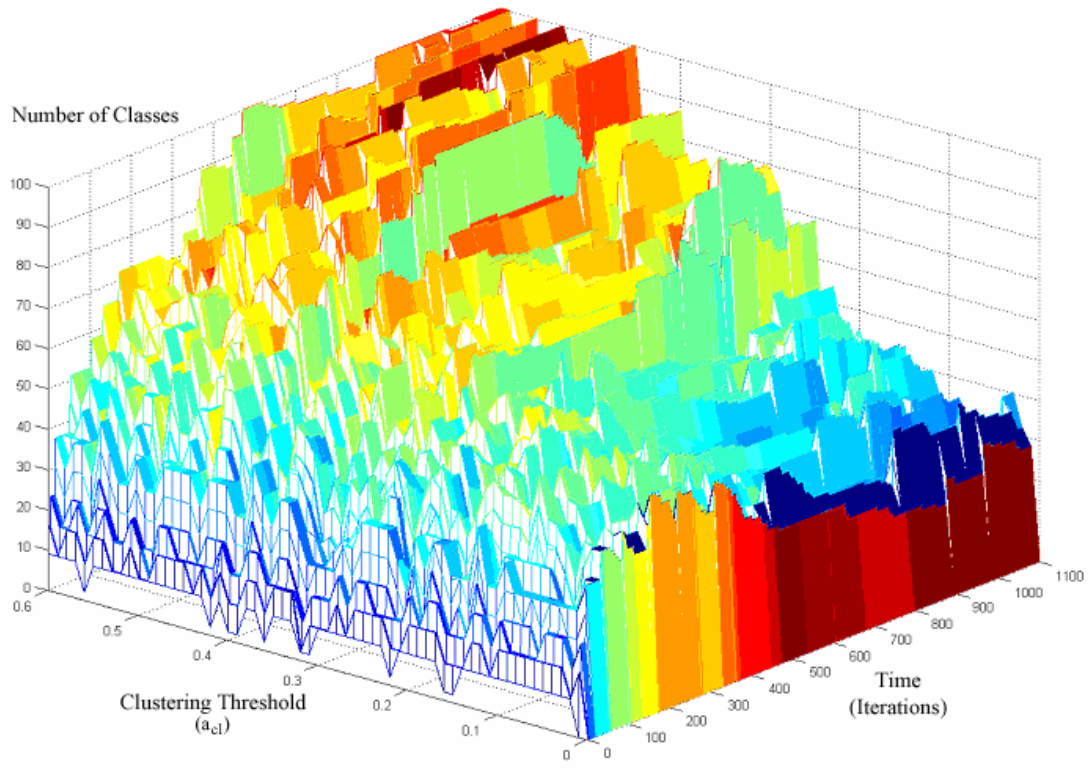

Figure 3.48 PSOM Test 7: The effect of the Clustering Threshold on the number of classes.

The effect of a_{cl} on the number of classes within the network can be seen in Figure 3.48. When a_{cl} is less than or equal to a_n, the network is able to classify as before. If $a_{cl} << a_n$ then the neighbourhoods are very tight and the number of classes is kept to a minimum. This suggests that a small a_{cl} increases the PSOM's ability to remove noise from the network. Intuitively, this is confirmed as a small a_{cl} allows the neighbourhood function to push dissimilar neurons apart. As $a_{cl} \rightarrow a_n$, this effect is reduced and the number of separate classes held by the network is greater. For values of $a_{cl} > a_n$, the neighbourhood function no longer operates correctly and the PSOM has difficulty classifying the 5 desired classes.

3.4.8.1 Test Summary

Although the clustering threshold has little effect on the network, when set incorrectly (larger than a_n) it can have negative effects on its ability to classify. For noise rejection purposes, the value of a_{cl} should remain small so that the neighbourhood function aids the ageing process rather than counter-acts it.

3.4.9 Test 8: The Effect of link ageing on noisy data.

This test mimics Test 4, in that it shows the effect of the Link Ageing Parameter (b_a) on the PSOM, using the noisy data set. Link ageing helps the network to remove unwanted neurons, enhancing noise rejection properties. Small values of b_a mean that there is no link ageing at all, and hence the network will retain all information showed to it. No topographical maps were taken for this test because the maps were similar to those of Test 4, the only difference being the addition of noisy neurons as with Test 5. The parameters for this test are shown in Table 3.12.

Table 3.12 PSOM Test 8 parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.2
Clustering Threshold (a_{cl})	0.2
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0.0 – 5

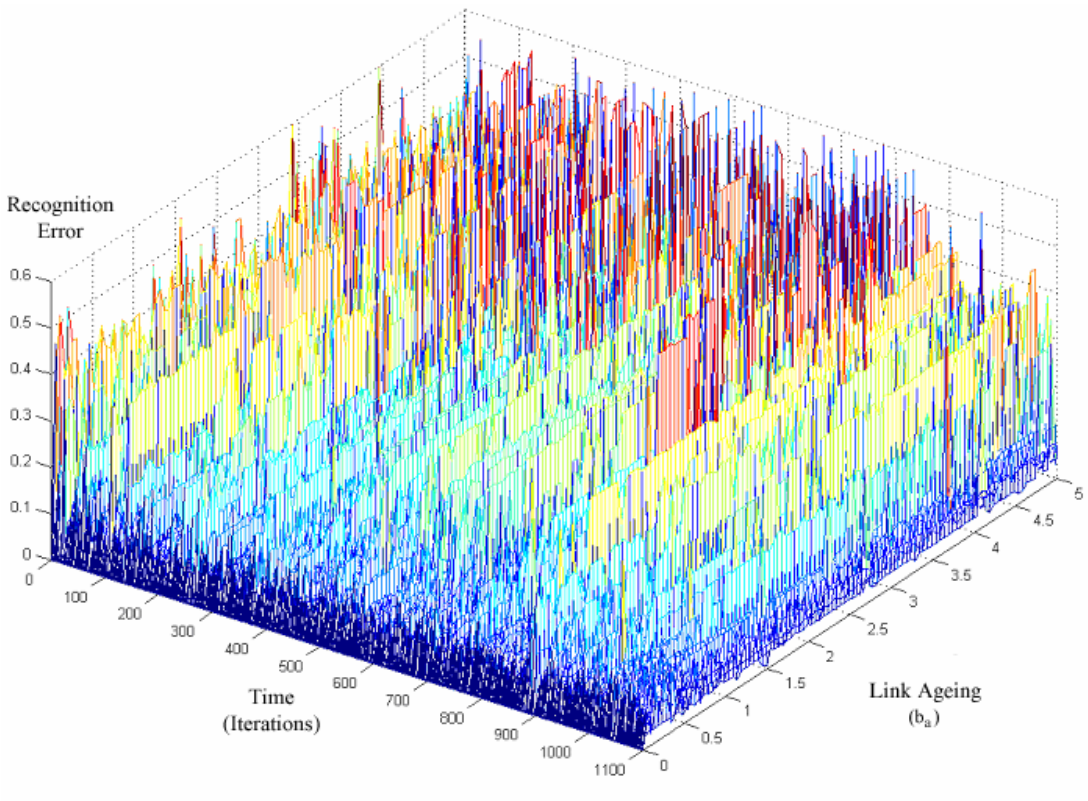

Figure 3.49 PSOM Test 8: The effect of link ageing (b_a) on the recognition error.

Figure 3.49 shows the effect of b_a on the Euclidean distance (error). At small values of b_a, the PSOM retains all the patterns it learns, thus the error level remains low. As

b_a increases, the speed at which patterns are lost is increased so the network loses knowledge quickly. This results in a larger error as every pattern shown to the network is seen for the first time.

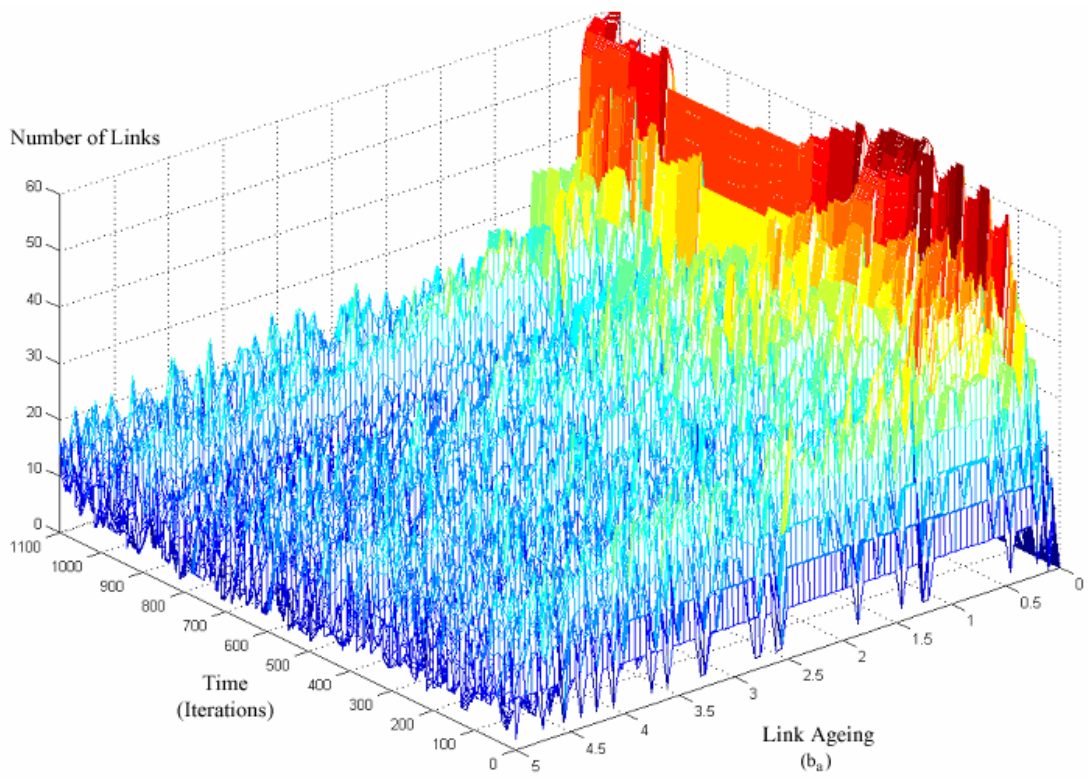

Figure 3.50 PSOM Test 8: The effect of link ageing (b_a) on the number of links.

Figure 3.50 shows the effect of link ageing on the number of links in the network. For small values of b_a, the number of links in the network is large as the ageing process is not having much effect. As b_a increases, the number of links reduces until it reaches a low level of between 6 and 9 links. These links are due to the persistent addition of new neurons and links. As the error (Figure 3.49) is large at this point, the network is learning anew with each iteration.

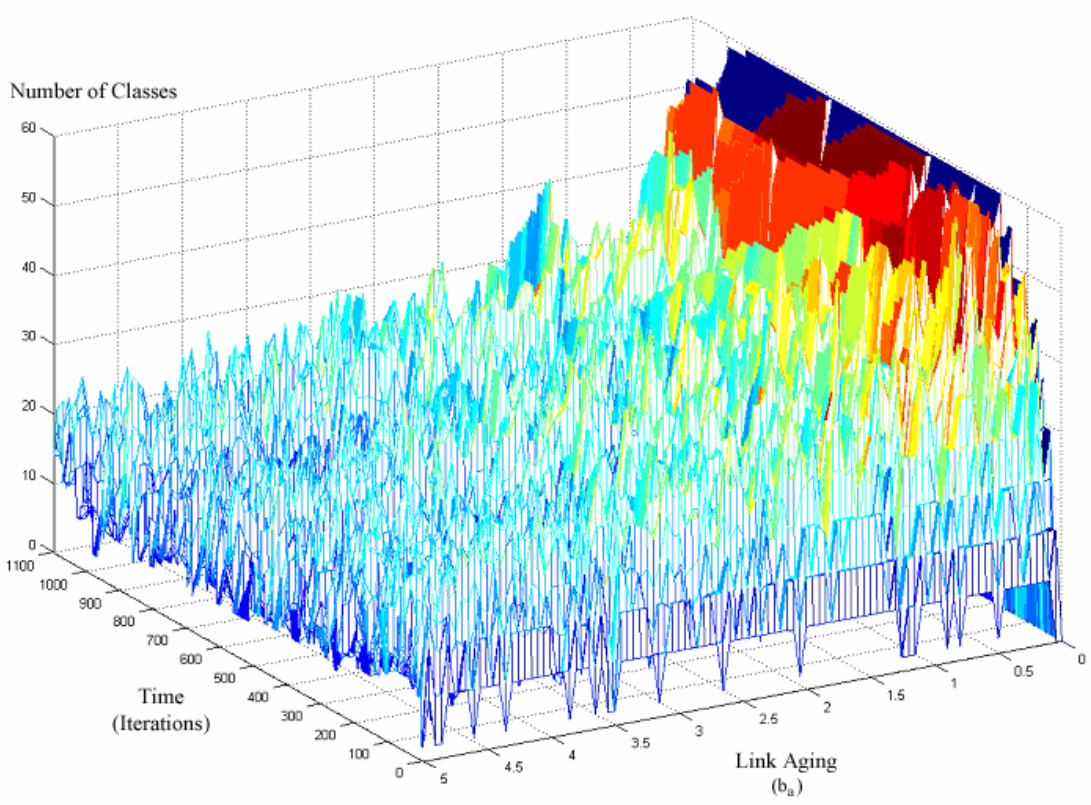

Figure 3.51 PSOM Test 8: The effect of link ageing (b_a) on the number of classes.

Figure 3.51 shows the effect of link ageing on the number of classes held by the network. For small values of b_a, the network does not remove links quickly and thus retains all the noisy classes along with the desired classes. As b_a increases, the number of classes in the network becomes small and links are quickly removed. With very high values of b_a, the network has difficulty retaining any information at all and the network is responding only to the last input.

Table 3.13 PSOM Test 8: The relationship between the input class number and the PSOM class number (noisy artificial data set) for a b_a of 4. Only significant rows are shown.

		Input Set Class Number					
		0	1	2	3	4	5
PSOM Class Number	0	5	0	1	1	9	0
	1	17	526	0	0	2	0
	2	9	0	159	0	1	0
	3	8	0	0	94	0	0
	10	1	0	0	0	13	0
	15	2	0	0	0	11	0
	24	9	0	0	0	49	0
	41	4	0	0	0	29	0
	49	3	0	0	0	0	80
	56	2	0	0	0	13	0

Table 3.13 shows the relationship between the input class number and the PSOM class number for a b_a of 4.0. For the sake of clarity, those rows with a few entries have been omitted and only those significant rows (with values of interest) remain. The pre-cropped size of the matrix can be gauged from the indices in the furthest left column. With such a high link ageing, the PSOM cuts links quickly. Therefore, one would expect those patterns that are not persistent to be rejected by the network. In this case, the link ageing parameter is large enough that the infrequent random class (Class 4) cannot be retained by the network. This is because of the long gaps between pattern presentations that can occur with this class. Although this large value of b_a might suggest that the network can no longer retain any information, classes 1,2,3 and 5 are all persistent enough to still be classified.

3.4.9.1 Test Summary

This test has shown the effect of the link ageing parameter on the PSOM. Small values of b_a lead to a low error but a large number of links and classes. This is because the network is retaining a lot of information and not removing neurons associated with classes with one or two occurrences only. High values of b_a lead to larger errors and smaller network sizes. However, these large errors are due to the neurons associated with noise being removed quickly. When a similar noise input is shown, the neuron associated with it is no longer present and thus the error appears to ‘spike’. The network still classifies the desirable, persistent classes well.

3.4.10 Test 9: Radar Data Set

This test is to show the operation of the PSOM on the Radar Data Set (Figure 3.5). It was hypothesised that each of the four classes within the data set would be classified as a unique class in the PSOM. Inspection of the data domain allowed the specification of some initial parameters (Table 3.14) that were then tuned so that the network produced a reasonable number of classes during operation.

Table 3.14 PSOM Test 9 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.21
Clustering Threshold (a_{cl})	0.2
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	0.04

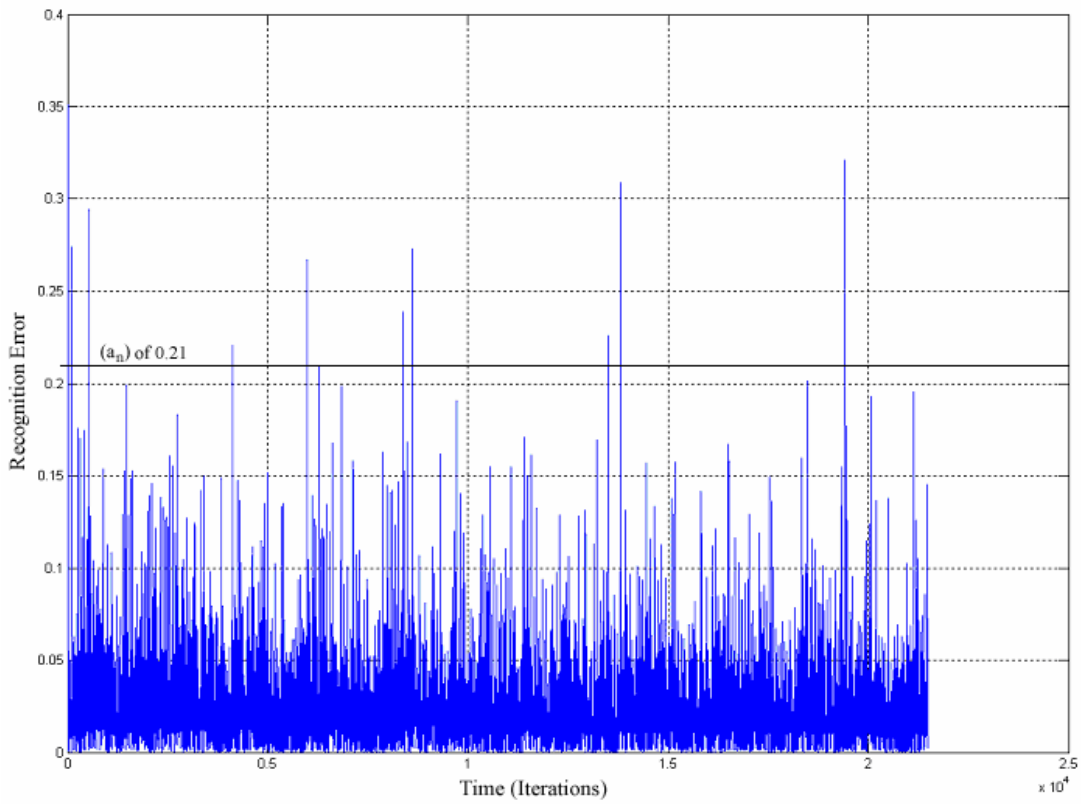

Figure 3.52 PSOM Test 9: Recognition Error over time.

Figure 3.52 shows the recognition error over time for one typical run. Although the error is highly variable, any values below 0.21 (a_n) are instances where a pattern is recognised by the network. This shows that there are very few instances where the PSOM needed to create new neurons and that the majority of inputs are recognised by the network.

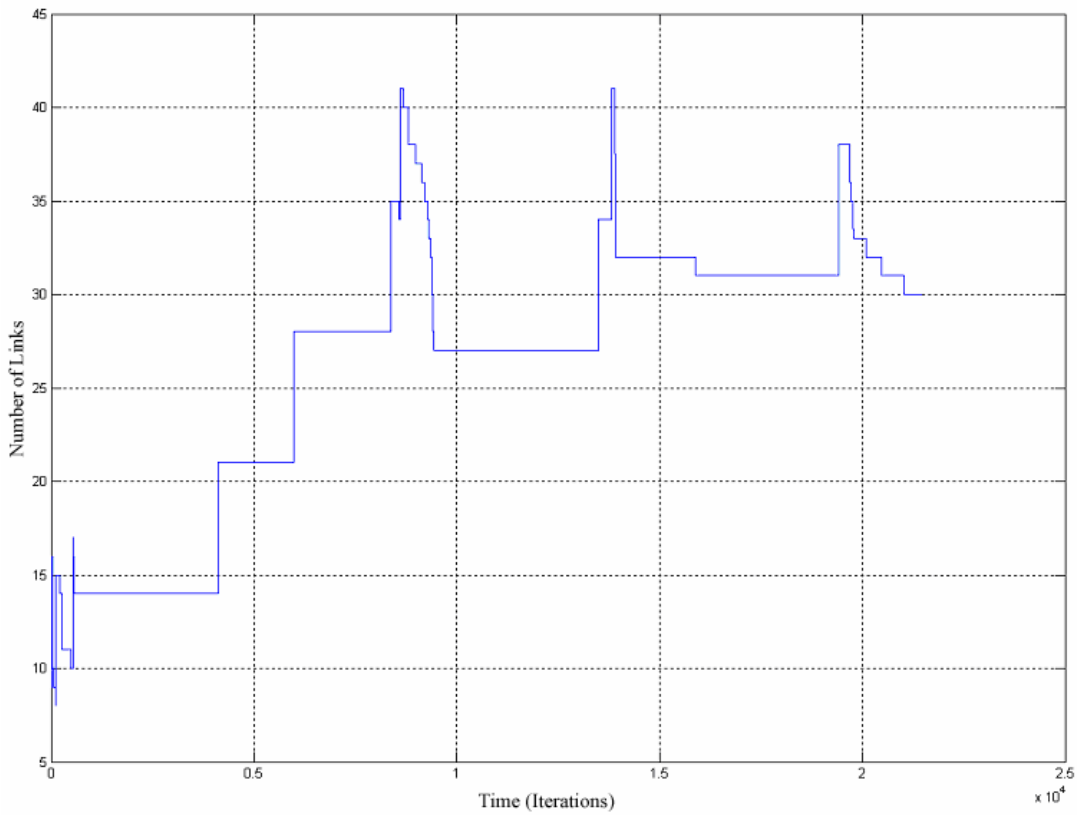

Figure 3.53 PSOM Test 9: The number of links over time.

Figure 3.53 shows the number of links over time for the Radar Data Set. The number of links can be used to help note whether a large recognition error is symptomatic of a new class or merely a noise class. If the pattern is a noisy pattern, then the links and neurons added will be quickly removed after the initial error. If the pattern corresponds to a class, then the neurons and links will remain in the network. The converse is not true, however. The addition of links cannot be used to determine whether a new class has been found. This is because an existing class may require more than the three initial neurons to cover the space the class retains. The speed at which the links are removed appears to be extremely rapid in this plot when

compared to previous tests but this is not the case because the time scale for this test was 20 times that of previous tests.

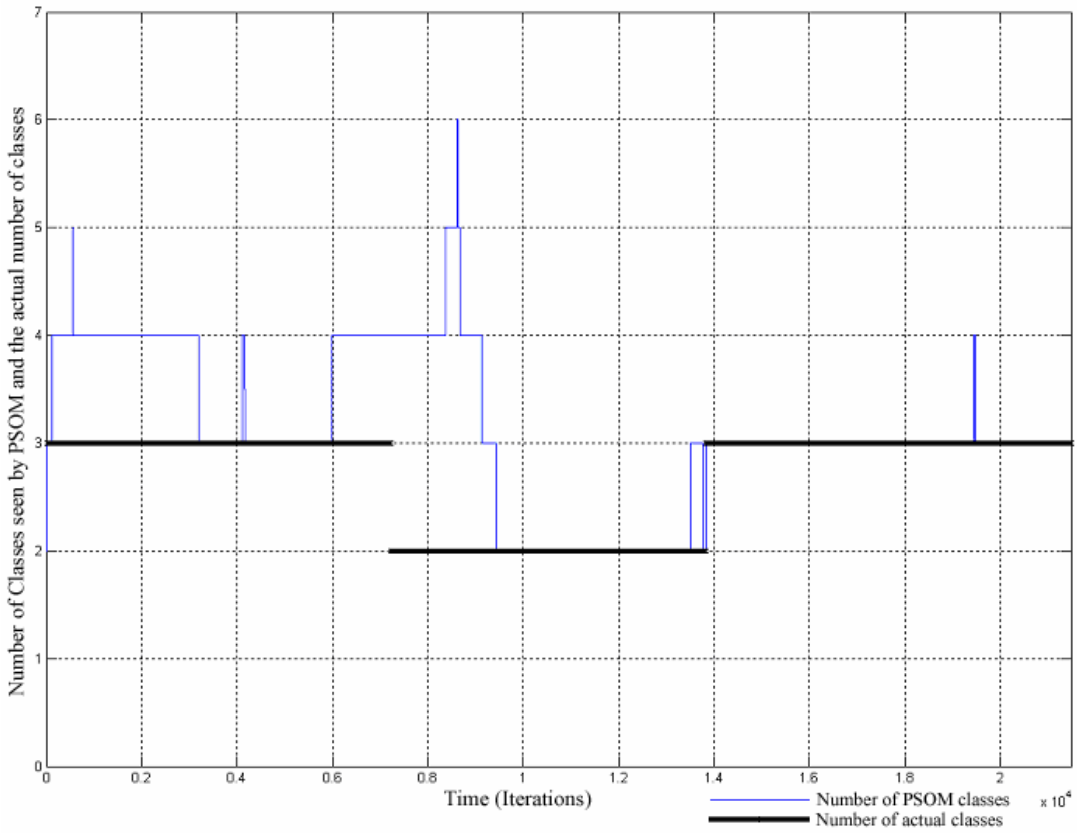

Figure 3.54 PSOM Test 9: The Number of classes held by the PSOM over time.

Figure 3.54 shows the number of classes held by the PSOM over time. The thin line represents the number of classes held by the PSOM during the length of the test and the thick black line shows the actual number of classes within the network at any time. Initially, the network grows to accept the new patterns. It should also be noted that the PSOM holds the initialisation neurons upon which all the other neurons are attached and this accounts for the additional class at the start of the test. The network takes roughly half of the length of the test to settle down. Classes that represent noise or outliers are seen as spikes: the neurons are quickly removed as no further noise patterns of this type are presented. Although this plot gives an indication of the network operation, it does not fully show whether or not the PSOM is classifying consistently or correctly.

Table 3.15 PSOM Test 9: The relationship between the input set class number and the PSOM class number.

		Input Set Class Number			
		1	2	3	4
PSOM Class Number	0	1	0	1	1
	1	13605	1	0	0
	2	0	0	6236	0
	3	4	0	0	370
	4	3	0	0	252
	5	6	0	0	0
	6	0	0	0	4
	7	0	0	0	0
	8	0	0	3	0
	9	0	0	5	0
	10	0	991	0	0

Table 3.15 shows the relationship between the input set class number and the class number give to the class by the PSOM. Input Class 1 was classified by the PSOM correctly for most of the operation. Those patterns that were classified differently are outliers of the class. An outlier is pattern of the class that has a high variance from the mean of the class. The same behaviour applies to input classes 2 and 3. Input class two was represented by PSOM class 10 and input class 3 was represented by PSOM class 2. Input Class 4 was split between PSOM classes 3 and 4. The proximity of the two class numbers suggests that the first examples of this class were from different edges of the class's spread in the input space. Inspection of the network during its operation confirms this. The result of this is that two groups were created for the same class and they were split early on. PSOM Class 3 was the dominant of the two. Over all the different runs, this split occurred infrequently (2% of the time) and this test run was included as an example of these run occurrences. In most runs, a point near the centre of the class will be shown first and a single class will therefore be used to represent it. Class splitting is a problem which is addressed in the discussion Chapter.

3.4.10.1 Network Topologies

The following Network topologies show the network performing on the Radar Data set under normal circumstances.

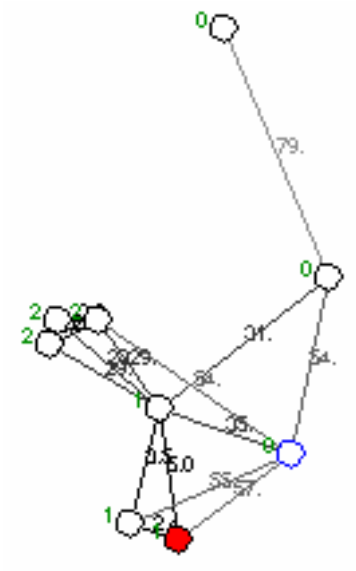

Figure 3.55 PSOM Test 9: PSOM after 21 iterations. With two patterns, the PSOM's early structure is much the same as the artificial data sets.

Figure 3.54 shows the initial growth of the PSOM after the first two classes have been shown to the network. At this early stage, the PSOM behaves in exactly the same manner as it did with the artificial data sets.

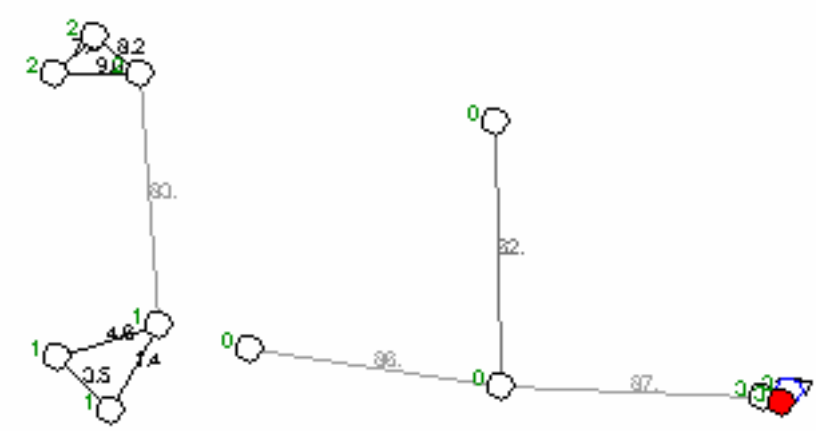

Figure 3.56 PSOM Test 9: Network after 218 iterations. Three pattern structure is in the process of breaking into distinct clusters.

Figure 3.56 shows the network after 218 iterations. Distinct clusters in the process of being broken apart can be seen. The initial neurons of class 0 still persist in the network due to a low link ageing parameter of 0.04. This low link ageing parameter is necessary to allow those classes with statistically long gaps between presentations to be remembered by the network.

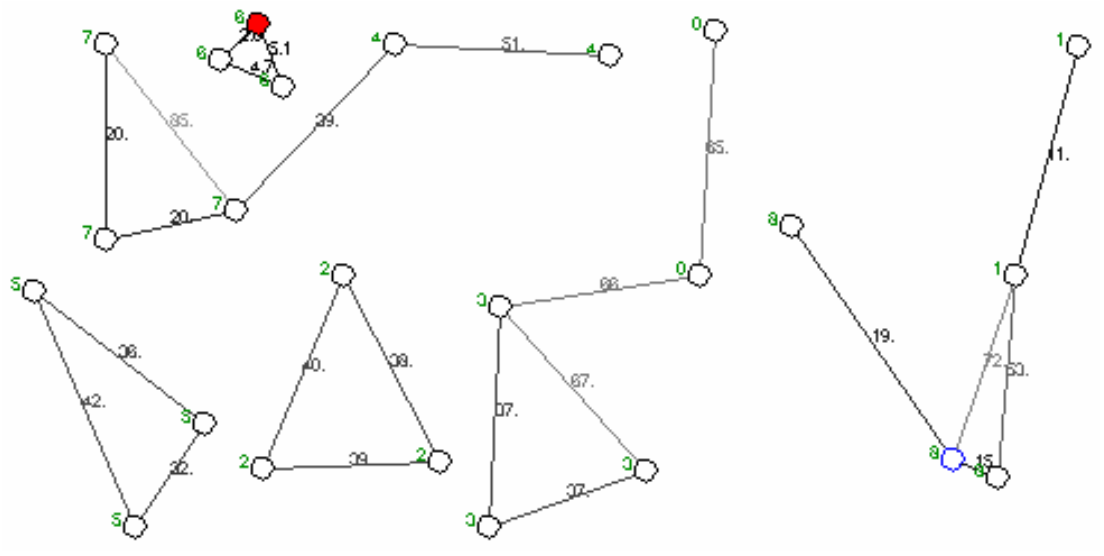

Figure 3.57 PSOM Test 9: Network after 1000 iterations. This shows the persistence of the initial neurons and some noise neurons.

Figure 3.57 shows the network after 1000 iterations and illustrates the type of topology that the network has for the rest of the test. The nature of the input set (typical for a radar application) presents many example patterns from a single class in a 'burst', while other classes are not shown. During this time, the other classes (of temporarily low temporal density) need to be remembered. In Figure 3.57, the neurons of class 6 represent the input class of the current burst and the other neurons represent other classes, neurons that were created due to noise, as well as the initial neurons. During the remainder of the test, noise neurons will be removed and new ones added in response to the presentation of new patterns and any new classes but the structure given in Figure 3.57 is typical of the remainder of the operation of the network.

3.4.10.2 Test Summary

Test 9 has shown that the PSOM can classify individual radar pulses given Bearing, Pulse Width and Radio Frequency. The PSOM automatically splits the different classes and correctly classifies three of the four different classes. The error of the test is bounded, as is the number of links. The network manages to keep a bounded topology, even in the presence of outliers. It is often the case in such systems that the presence of an outlier is of interest. Such an event can be trapped by monitoring the number of links or the recognition error. In some cases, the network will not

distinguish between an outlier and a new class. In this case, the Node Building Parameter might need to be modified.

3.4.11 Test 10: Comparison with GNG-U

This test compares the GNG-U and PSOM algorithms, using the 2 Dimensional data set of Figure 3.6. In the original GNG-U work (Fritzke 1997), throughout this section referred to as "Fritke's Work", parameter values were offered for showing the network running at its best. A brief investigation of parameter tuning was performed but no appreciable improvement to the speed of GNG-U could be made without sacrificing topographical mapping performance. The GNG-U parameters used are reproduced in Table 3.16.

Table 3.16 The parameters used for the GNG-U algorithm.

Parameter	Value or Range
Node adding parameter (λ)	500
Node update amount (ε_b)	0.05
Neighbourhood update amount (ε_n)	0.0006
Maximum link length (a_{max})	120
Error Reduction (α)	0.5
Forgetting factor (β)	0.0005
Utility Threshold Measure (k)	3
Maximum number of nodes	30

An official implementation of the algorithm has been created and made available by Fritzke[3] and Loos. This implementation was used to check and confirm the operation of the author's code used to create this test. The results presented here can be reproduced using the Fritzke and Loos' implementation. The results shown in Fritzke's 1997 paper, however, could not be reproduced by either the implementation of GNG-U described in this thesis *or* the Fritzke and Loos' of GNG-U implementation available on the web site. Both implementations operate in an identical manner as described below.

[3] Web Reference for the GNG-U source is:
http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html

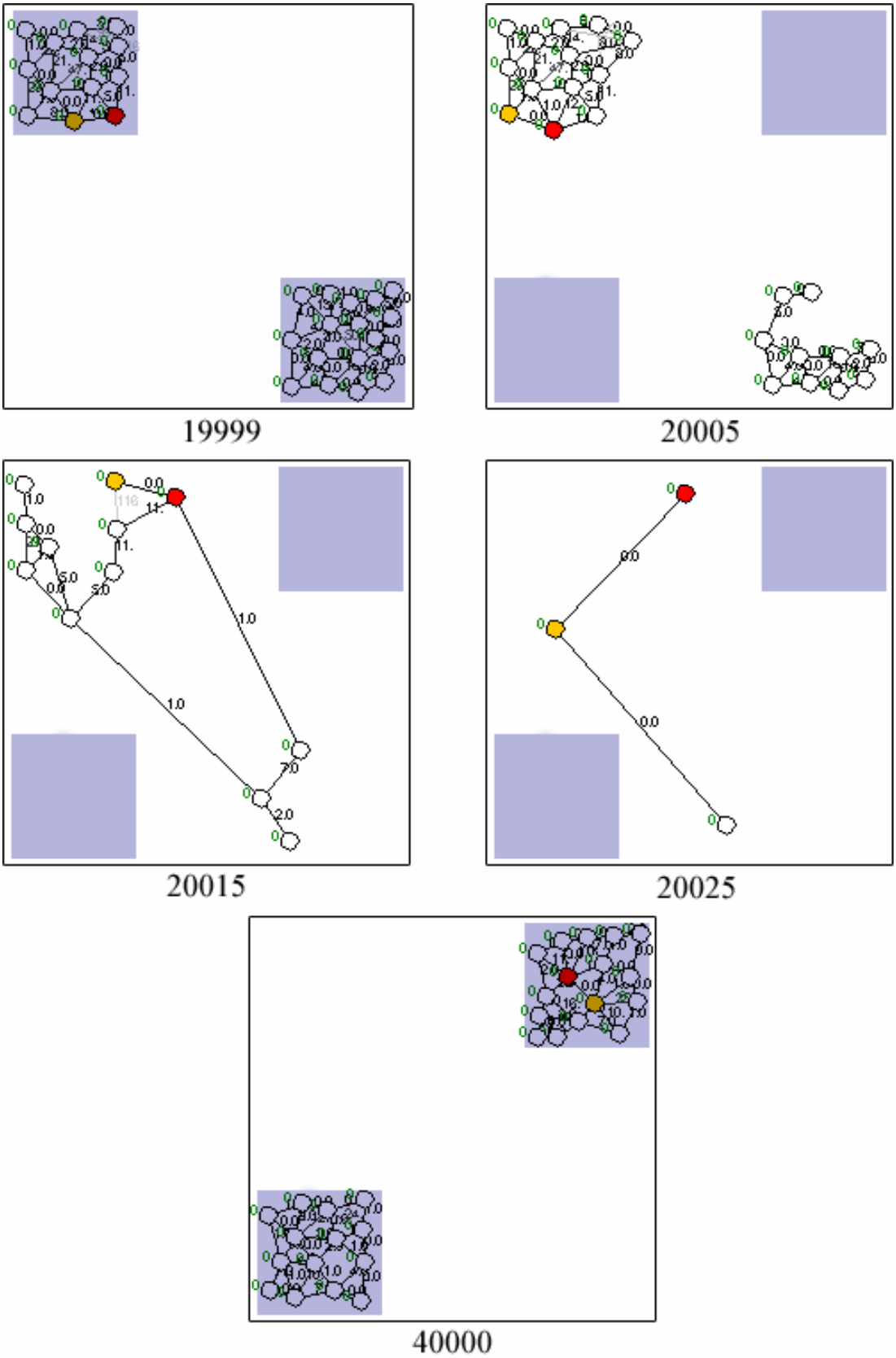

Figure 3.58 Test 10: The GNG-U algorithm's topology at different stages in the network's life. Each image is labelled with its iteration. The input set is represented by blue squares.

Figure 3.58 shows the GNG-U network's topologies over 40,000 iterations (as used in the original work). After 19,999 iterations, the network has adapted to learn the two class input set. GNG-U has no notion of containing different classes and this is for the observer to determine. At 20,000 iterations, the network switches to the second data source (not shown as the image is almost identical to 19,999 but with the squares in the locations as seen in the other images). At this stage, the network is behaving exactly as described in Fritzke's work.

The differences occur after the input distribution switch has been made. In Fritzke's work, the existing nodes take many thousands of iterations before they are removed. However, as is shown in Figure 3.58 and the web site algorithm (not shown), GNG-U begins removing nodes as soon as the switch is made. This is because the network has over-fitted the original input set. This can occur when the network is shown many examples of the same input classes or the density of the nodes is too high. In either case, the usefulness (Utility, U_c) of each node is less, which means that the Utility of the node may be small enough in comparison to the largest error in the network (E_q) for most of the nodes in the network to be removed. By the time the distribution switches, the nodes all have a very low Utility as there are many nodes to represent a small input space. The switch causes a sudden increase in the maximum error and this leads to many nodes being over the E_q / U_c threshold of 3. Being over this threshold means that the neurons are removed. Each iteration removes a number of neurons and the attrition is extremely fast – much greater than published in the original work.

Large holes in the topology can already be seen just five iterations after the switch (20005) and by 20025, the network has reduced itself back to its minimal form of 2 nodes and a single link. In this example, the locations of each input are shown to the network whereas in Fritzke's work a transformation may have been used.

The PSOM was tested on a transformed version of the data set. The transformation performed normalised the dataset from a 0-230 scale (the width and height of the input set) to a 0-1 scale. This was simply performed by dividing each input by 230. This kind of transformation requires a priori knowledge of the input set – the upper

bounds of each dimension. Another small alteration was made to the dataset: the time at which the input switch from the first distribution (Figure 3.6, pattern 1) to the second (Figure 3.6, pattern 2). It was not necessary for the PSOM to be shown 20000 instances of the input distribution (as shown later in Figure 3.59), so the change was made after 500 iterations. This number was chosen empirically to allow the PSOM enough iterations for the recognition error to stabilise.

The parameters used for the PSOM test are shown in Table 3.17. The initial values for the parameters were derived using the heuristics given in Chapter 1 and then tuned to achieve the results given below.

Table 3.17 PSOM Test 10 Parameters.

Parameter	Value or Range
Node Building Parameter (a_n)	0.2
Clustering Threshold (a_{cl})	0.1
Link Removal Threshold (a_r)	90
Link Ageing (b_a)	1

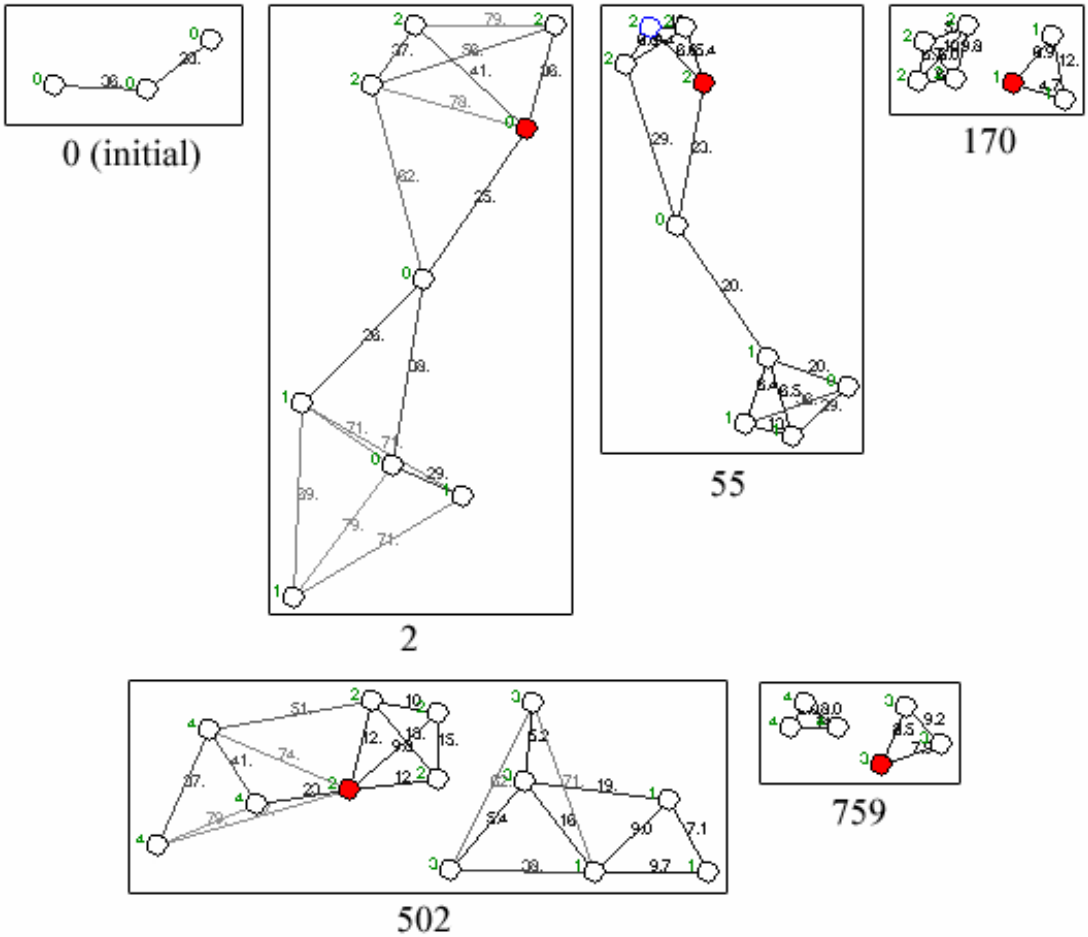

Figure 3.59 PSOM Test 10: The PSOM at key points during Test 10. The iteration time is given with each image.

Figure 3.1 shows the network topologies at key stages during the network's operation for a typical test run. Repeat tests gave similar results. At iteration 0, no inputs have been shown and so this is the initial state of the network. On this test run an input from each class was produced in iterations 1 and 2. Therefore, the network was able to grow for both the classes in the input set early on. This is not always the case; it is possible for 10 or 15 example patterns of the same input class to be shown to the network. This does not present a problem for the PSOM as it grows to accommodate the input data.

After 55 iterations, the network is in the process of breaking the patterns apart and it is already possible to begin to distinguish two groups. At 170 iterations, the network has separated the classes and is now representing the input distributions. This network state does not change greatly until 500 iterations when the input distribution is changed.

After 502 iterations, the network has seen examples of the two new input classes. The nodes representing the new patterns have attached to the nodes belonging to the two old classes. It is possible that both new classes attach to the same existing node group but this does not affect the operation of the network. Unlike with GNG-U the forthcoming transition from one stable state to the other is a gradual and slow process. This is because the PSOM has not over-fitted to the input data. At 759 iterations, the network only contains nodes corresponding to the two classes of the second input distribution. Nodes associated with the old input distribution have been completely removed as expected.

3.4.11.1 Test Summary

Test 10 compared the two similar algorithms of GNG-U and the PSOM. GNG-U retains the topology of the input space where as PSOM does not. The PSOM learns much quicker than GNG-U but this may be due to unknown tuning in GNG-U. Both networks find both classes and can change to track the change in the input space. GNG-U does so rapidly, having over-fitted the input set, whereas the PSOM does so in a gradual manner. A further comparative study is presented in Chapter 6, in the Discussion section.

3.5 Chapter Summary and Conclusions

In this Chapter, the PSOM was shown to cluster and classify in both artificial and real world non-stationary systems. Each of the parameters of the PSOM was investigated. For each of the network's parameters, there is a range of values that will allow the network to classify correctly. In noisy environments, the network temporarily creates neurons to represent the new patterns. These neurons are removed over time but there will always be a few noisy neurons in the topology. The PSOM was demonstrated operating on a real world data set of Radar Pulses, which is classified correctly. Finally, the PSOM was compared numerically to GNG-U and the PSOM reached a stable state quicker than GNG-U. To conclude, the PSOM has been shown to cluster and classify in non-stationary, noisy environments with careful tuning of the parameters.