2006-01-3492

Self-Organizing Maps with Unsupervised Learning for Condition Monitoring of Fluid Power Systems

Krogerus, T.; Vilenius, J.; Liimatainen, J. and Koskinen K.T.
Tampere University of Technology, Institute of Hydraulics and Automation

ABSTRACT

The goal of this paper is to study a proactive condition monitoring system for fluid power systems where the Self-Organizing Maps (SOM) with unsupervised learning is used to classify and interpret high-dimensional data measurements. If all the damages are not assumed to be known before diagnostics, an ordinary neural network with supervised learning for their detection can not be used. Operation of the proactive condition monitoring system is tested in a test system where two fault types are used. The test system is run in normal and two different fault situations. Measurement results are used for training and testing the SOM. In this paper these measurement results and also the quality of state recognition are shown.

INTRODUCTION

Neural network research has been very active for several years and there has also been interest in neural network applications to fault diagnosis problems. A neural network is one possible method which is suitable for detecting changes in the state of the fluid power system. [8, 9, 12, 13, 15]

In fluid power, the systems are very non-linear and the behavior is often highly dynamic. The amount of measurement data from these systems can be quite large and the relationships of the measurement variables can become complex. The Self-Organizing Maps (SOM) is a neural network method which can represent any functional relationship between inputs and outputs. Therefore the SOM can be used with fluid power systems very effectively and by using it, it is possible to determine the state of the system. In this paper, term state is used to distinguish different system status (normal / fault). [4, 14]

The SOM converts complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. So it compresses data while preserving the most important topological and metric relationships of the original data. It consists of neurons organized on a regular low-dimensional grid. These neurons are organized so that similar neurons are near and different ones far away each other. [4, 14]

The SOM algorithm is used here to classify and interpret data measurements. Usually when neural networks are used for fault detection, supervised training is used. Supervised training means that desired outputs (state of the system, normal / fault) are used in training. But often the real case is that fault situations are not known before and an ordinary neural network with supervised learning for fault detection can not be used. Therefore unsupervised learning is used although here the fault situations are known before.

In unsupervised learning, the network is not trained towards specified outputs. Networks that are trained without outputs learn by evaluating the similarity between the input patterns presented to the network. They make use of the statistical properties of the input data as frequently occurring patterns will have a greater influence than infrequent ones. So the weights are modified in response to network inputs only. Most of algorithms, which use unsupervised learning, perform some kind of clustering operation where they categorize the input patterns into a finite number of classes [3]. Unsupervised learning makes use of the redundancy present in the input data in order to produce a more compact representation.

The operation of the neural networks is completely based on the measurement data which it inputs to the network. The selected data must contain enough information from the system that it is possible to detect any abnormal deviation in the system. If the deviation, caused by some fault, do not show in the measurement data, then it is impossible to detect this fault with neural network. Because of this measurement variables should be selected so that they describe the behavior of the system well enough so that fault situations can be detected.

Usually in the literature, when neural network classifiers have been used to detect and diagnose fault situations, studied systems are not in their normal use but systems are diagnosed afterwards. Here studied system is in normal use and dynamic sequence measurements from the system are used to train and test the neural network.

When neural network is used for classification and it is trained using measurement data from the system, operating point changes can become a problem. Normal situations can be classified as a fault although only the operating point has been changed. If different kind of operating points can be separated and trained to the network then it is not a problem.

The training algorithm which is used in this study is simple, robust to missing values, and perhaps most importantly, it is easy to visualize the map. These properties make SOM a prominent tool in data mining [16].

Sometimes the performance of the network can be decreased if high measurement frequency and/or large number of measurements variables are used. Therefore principal component analysis (PCA) is used for visualization of the measurements and to reduce the number of measurement variables. PCA simplifies the problem by replacing a group of variables with (a single) new variable(s).

PROACTIVE CONDITION MONITORING SYSTEM

The studied proactive condition monitoring system for fluid power systems consists of five different parts which are data acquisition, (wireless) data transfer, database, neural network and user interface. Operation of the proactive condition monitoring system is described more detailed in [6]. Figure 1 shows the system structure of the proactive condition monitoring system.

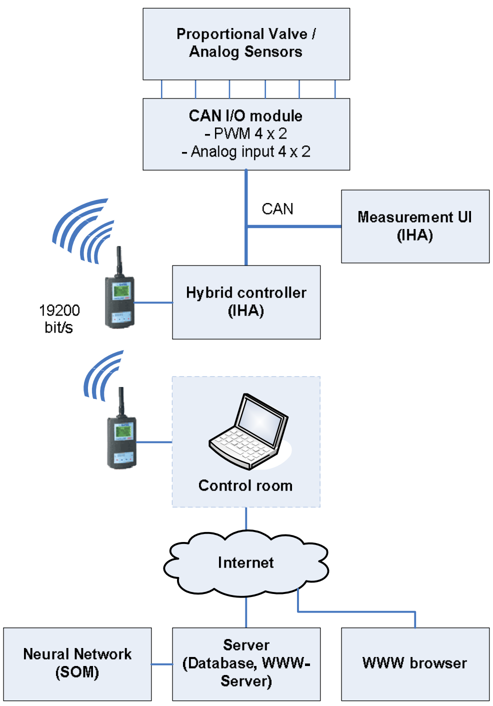

Fig. 1: System structure of the proactive condition monitoring system [6].

In this paper term proactive condition monitoring is used for condition monitoring which monitors abnormal deviation in measurement variables and also controls the state of the system and the components according to the feedback information. State of the system is controlled by tuning the control parameters or driving the system down. It is also possible to continue using the system at a lower utilization rate. This way it is possible to know the information about the conditions of the fluid power components all the time and the risk of failures can be minimized. [5, 6]

SELF-ORGANIZING MAPS

The Self-Organizing Map (SOM) converts complex, nonlinear statistical relationships between high-dimensional data items into simple geometric relationships on a low-dimensional display. So it compresses data while preserving the most important topological and metric relationships of the original data. [4, 14]

The SOM consists of neurons organized on a regular low-dimensional grid. These neurons are organized so that similar neurons are near and different ones far away each other. The SOM learns to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors [11]. Similar system states (normal / fault) have similar variable values and therefore it is possible to separate different system states. Each neuron is presented by a d-dimensional weight vector $\mathbf{m} = [m_1, \ldots, m_d]$, where d is equal to the dimension of the input vectors, and they are connected to adjacent neurons. In the SOM training algorithm the best-matching weight vector and also its topological neighbors on the map are updated. This is shown in the Fig. 2. [4, 14]

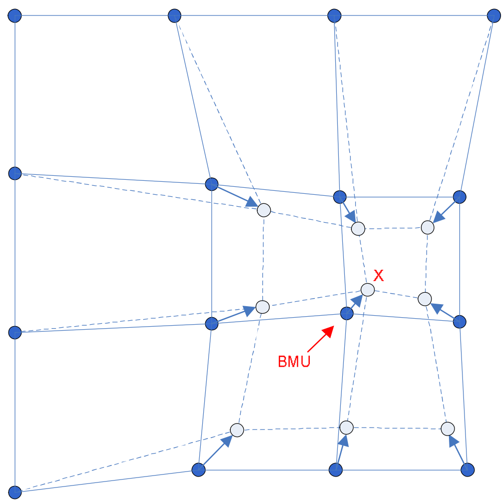

Fig. 2: Updating the best matching unit and its neighbors [14].

The SOM Toolbox, implementation of the SOM and it's visualization in the Matlab computing environment, is used to create and use the neural network. There are two variants of the SOM training algorithm in the Toolbox. In the traditional sequential training data samples are presented to the SOM one at a time and the algorithm gradually moves the weight vectors towards them. In the batch training the data is presented to the SOM as a whole and the new weight vectors are weighted averages of the data vectors. [14]

In the sequential training the SOM is trained iteratively. In each training step, one sample vector **x** from the input data set is chosen randomly and the distances between it and all the weight vectors of the SOM are calculated using distance measure, typically Euclidian distance. [4, 14]

Here the neuron whose weight vector is closest to the input vector **x** is called the Best-Matching Unit (BMU), denoted by *c* [4, 14].

$$\left\|\mathbf{x}-\mathbf{m}_c\right\| = \min_i\left\{\left\|\mathbf{x}-\mathbf{m}_i\right\|\right\} \tag{1}$$

In the Toolbox the distance computation is performed slightly differently, where *K* is the set of known (not missing) variables of sample vector **x,** x_k and m_k are k^{th} components of the sample and weight vectors and *wk* is the k^{th} mask value [14].

$$\left\|\mathbf{x}-\mathbf{m}\right\|^2 = \sum_{k\in K} w_k\left(x_k - m_k\right)^2 \tag{2}$$

When the BMU has been found, the weight vectors of the SOM are updated so that the BMU and its neighbors are moved closer to the input vector in the input space, see Fig. 2. After this the SOM update rule for the weight vector of unit *i* goes according Eq.3, where *t* is time. The $\mathbf{x}(t)$ is an input vector randomly chosen from the input data set, $h_{ci}(t)$ is the neighborhood kernel around the winner unit *c* and $\alpha(t)$ is the learning rate. [4, 14]

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)h_{ci}(t)\left[\mathbf{x}(t)-\mathbf{m}_i(t)\right] \tag{3}$$

Both learning rate and neighborhood kernel radius decrease monotonically with time. The used neighborhood kernel $h_{ci}(t)$ is gaussian. The kernel gets its biggest value for the map unit, decreases monotonically with increasing distance on the map grid $\|\mathbf{r}_c - \mathbf{r}_i\|$. The used neighborhood kernel is shown in Eq. 4, where $\mathbf{r}_c$ and $\mathbf{r}_i$ are positions of neurons *c* and *i* in the output space, and σ is the neighbourhood radius. [14, 16]

$$h_{ci}(t) = e^{-\frac{\|\mathbf{r}_c-\mathbf{r}_i\|^2}{2\sigma^2(t)}} \tag{4}$$

The training is usually performed in two phases. In the first phase relatively large learning rate and neighborhood radius is used. In the second phase they are small right from the beginning. So first is the tuning of the SOM approximately to the same space as the input data and then fine-tuning the map. [4, 14]

An alternative for traditional sequential training algorithm is the batch training algorithm, where all the data points are presented to the map before making any changes to the map. The new weight vectors are calculated according Eq.5, where *c* is the index BMU of data sample $\mathbf{x}_j$, $h_{ci}(t)$ is the neighborhood function (the weighting factor), and *n* is the number of sample vectors. [14]

$$\mathbf{m}_i(t+1) = \frac{\sum_{j=1}^{n} h_{ci}(t)\,\mathbf{x}_j}{\sum_{j=1}^{n} h_{ci}(t)} \tag{5}$$

Alternatively one can first calculate the sum of the vectors in each Voronoi set according Eq.6, where n_{V_i} is the number of samples in the Voronoi set of unit *i* [14].

$$\mathbf{s}_i(t) = \sum_{j=1}^{n_{V_i}} \mathbf{x}_j \tag{6}$$

After this the new values of the weight vectors can be calculated according Eq.7, where *m* is the number of map units. This allows a much more efficient matrix-based implementation than using Eq. 5. This way the batch algorithm has been implemented in the Toolbox. [14, 16]

$$\mathbf{m}_i(t+1) = \frac{\sum_{j=1}^{m} h_{ij}(t)\,\mathbf{s}_j(t)}{\sum_{j=1}^{m} n_{V_j} h_{ij}(t)} \tag{7}$$

The number of map units is selected according Eq.8, where *M* is the number of map units and *n* is the number of data samples [16].

$$M = 5\sqrt{n} \tag{8}$$

It has been proved that the SOM algorithms can be initialized using random values for the codebook (weight) vectors and still these vectors will be ordered in the long run. But this is not the best or fastest way. In this study linear initialization has been used. [4]

Both of the SOM algorithms classified the state of the system almost as well if the criteria is the correct state of the system. In this paper the batch algorithm is used because it is much faster, especially with Matlab functions. The main reason why the SOM neural network method is used here for the classification is that this type of network could be used in situations where the class information is not available. In this study the fault situations are known before but they are not used in training in any way. If clustering of input data is completely or partially ascertained, semantic labels may be attached to certain units of the topological map. After classification a certain threshold is set. This determines the greatest distance on which recognition occurs. If the map units are labeled and thresholds are determined properly, the network may be used as a detector of new events. [7]

TEST SYSTEM

Operation of the proactive condition monitoring system is tested in a test system where two fault types are used. The test system is run in normal and two fault situations. The effect of the fault situations in the performance of the proportional valve and the cylinder has been examined. The hydraulic components which are studied here are used for example in the cranes of the forwarders. The simplified hydraulic circuit of the test system is shown in Fig. 3. More details of the hydraulic circuit is presented in [6, 10].

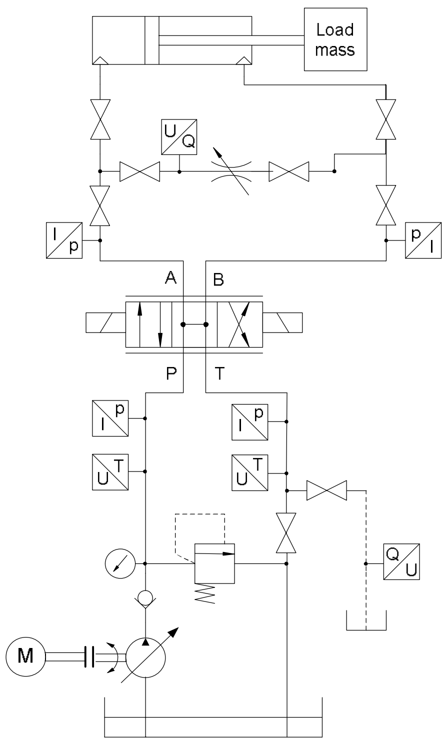

Fig. 3: Simplified hydraulic circuit of the test system [6, 10].

FAULT TYPES USED IN THE TEST SYSTEM

Fault types of interest are worn spools of pilot operated proportional valves and worn/damaged cylinders which both give increased leakage and worn spool also changes the pressure characteristics of the valve. Both of the fault types studied here are artificial faults. Used fault types are described and the impact of these modifications is shown in [6, 10].

The spool of the valve has been mechanically modified so that the control edge (P-A) from one half and the spool land from other half of the spool has been worn. These modifications try to simulate erosive and abrasive wear of the valves. They change the pressure characteristics of the valve and give extra leakage due to the increased clearance between the spool and the spool housing. The leakage between the annulus and piston sides of the cylinder is created by opening a bleed valve. The accuracy of these faults is here a secondary thing because the main issue is to demonstrate that presented neural network method, Self-Organizing Map (SOM), is suitable for detecting changes in the state of the fluid power system. [6, 10]

TRAINING AND TESTING DATA

The test system is run in normal and two fault situations. Measured variables are pressures A and B from the actuator ports and control signal of the proportional (valve current). These variables are then used in training and testing the SOM. The measurements are filtered before using them in training or testing. The measured pressures contain measurement noise, which make the classification made by neural network, more difficult. The filter used here is Chebyshev Type II lowpass filter. The filtering improved the result of the classification a little. On the other hand the effect of filtering to the total result was marginal. It was not possible to recognize right 100% of the states because the training and the testing data contains situations where the normal and fault state measurements are very close to each other and therefore some of the classified states are wrong. [10, 11]

The same sequence is measured five times. This is done because this way it is possible to improve the generalization of the network when new data is presented to the network.

In Fig. 4-7 are shown an example of the measurements. From the figures it can be seen how some of the normal and faulty measurement points are very close to each other. This causes errors in classification. In Fig. 4 is shown the control signals of the sequence measurements (control signals are one upon the other in the figure).

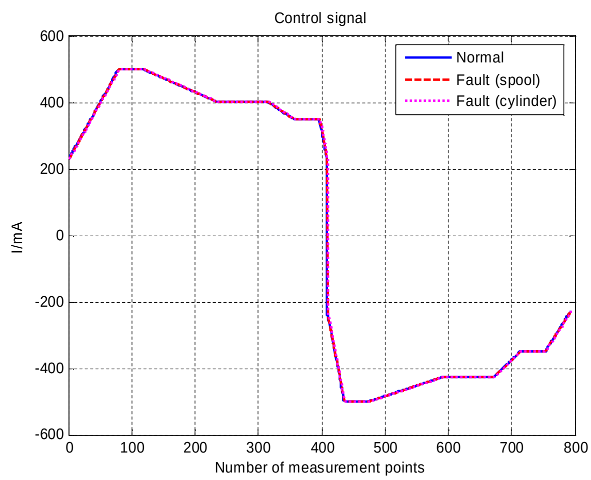

Fig. 4: Control signal from the sequence measurements [6].

In Fig. 5-6 are presented pressures that are used in classification. From the figures can be seen the differences between the normal and fault state measurements. Fault 1 (spool) measurement differs more clearly from the normal situation than fault 2 (cylinder). Difference between fault 2 and normal situation is quite small most of the time. From these figures can also see that both pressures tend to change in a same way.

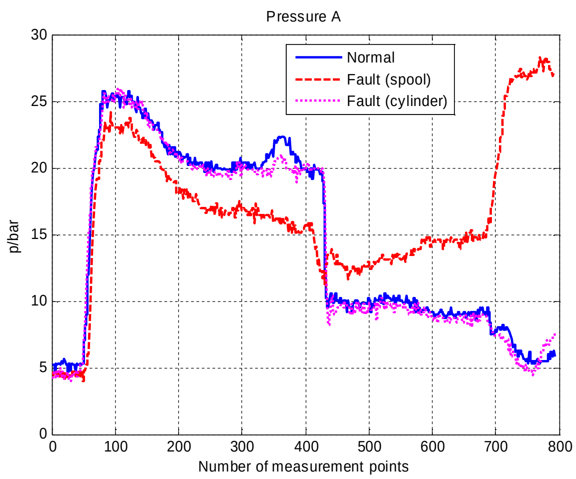

Fig. 5: Pressure A from the sequence measurements [6].

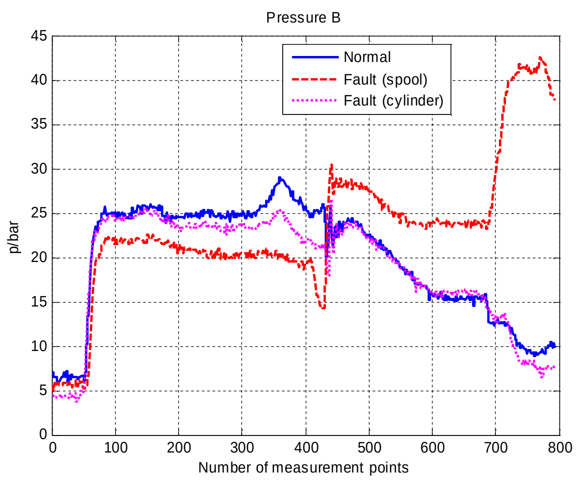

Fig. 6: Pressure B from the sequence measurements [6].

DIMENSION REDUCTION

Sometimes the performance of the network can be decreased if high measurement frequency and/or large number of measurements variables are used. But the problem dimension can be reduced and still maintain sufficient performance of the classification [7].

When there are data sets with many variables, groups of variables often move together. One reason for this is that more than one variable might be measuring the same driving principle governing the behavior of the system. In many systems there are only a few such driving forces. Situation like this, when there is redundant information, it is possible to simplify the problem by replacing a group of variables with (a single) new variable(s). [11]

The problem dimension is not a problem here because there are only three inputs and therefore the principal component analysis (PCA) is used mainly for visualization of the measurements. Now it is possible to study the properties of the measurement data in a 2-dimensional space. When there are more than three variables, it is usually very difficult to visualize their relationships.

PCA is also tested here to reduce the number of measurement variables and therefore problem dimension. The classification results are showed only for situations where PCA is not used because PCA change the final classification result only a little. PCA has been used also in [12, 16] to visualize and/or reduce the problem dimension before training the SOM. PCA generates a new set of variables which are called principal components. Each principal component is a linear combination of the original variables. There is no redundant information because all the principal components are orthogonal to each other. [11]

The first principal component is a single axis in space. When each observation is projected on that axis, the resulting values form a new variable. The second principal component is another axis in space, perpendicular to the first. Projecting the observations on this axis generates another new variable. [11]

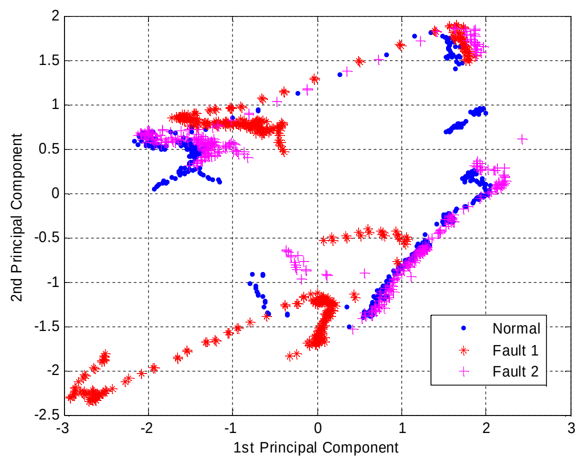

Fig. 7: The first and second principal components of the sequence measurements (Control signal, Pressure A, Pressure B).

In Fig. 7 is shown the first and second principal components of the sequence measurements which explain about 98% of the total variance of measurement data. In Fig. 7 is the original data mapped into the new coordinate system defined by the principal components. In this case pressure signals A and B and control signal were replaces with two new variables (first and second principal component). The sequence measurements are from normal and two fault situations. From Fig. 7 can be seen that fault 1 which is worn spool differs more from the normal situation than fault 2 which is worn / damaged cylinder. It is obvious from the figure that fault 1 is then easier to detect than fault 2. This overlapping of measurements complicates the classification process.

CLASSIFICATION RESULTS

The classification problem is approached here from two different viewpoints. The SOM is trained and tested with two different methods. In both cases unsupervised training is used.

In the first case the fault situations are known before and the network is trained with normal and fault situation data. Because fault situations are known before training, semantic labels can be attached to certain units of the topological map after training the map.

In the second case the fault situations are not known and only data from the normal situation is used to train the network. The map units which are hit during the training are labeled after the training as a normal situation. In this paper, term hit means that neuron in the map has been chosen (at least one time) as a BMU in the training phase. After this a certain threshold is set. This determines the greatest distance on which recognition occurs. If the map units are labeled and thresholds are determined properly, the network may be used as a detector of new events. [7]

FAULT SITUATIONS ARE KNOWN

Training the network was performed in two parts. In the first phase 4 and in the second phase 15 iteration rounds were used, when final quantization error (FQE) value was 0.0100 and topographic error 0.0945 when both fault situation are used in training.

The quantization error is the distance of an input vector from the closest codebook (weight) vector in the input space. The quantization error over all available input data is a sensitive measure of the mapping accuracy. If the configuration of the models has not yet reached the stable state in the learning process, or if there are unwanted "twists" in the map, the quantization error remains significantly higher than at the ordered optimum. It is a quite another question theoretically and also from the practical point of view whether the quantization error alone describes the topological order of the maps. The topologic error is described in The SOM Toolbox with value, which is the proportion of all the data vectors for which first and second BMUs are not adjacent units. [4, 14]

In the first case three different situations were trained to the SOM. The first one is the normal situation and spool fault, the second one is the normal situation and cylinder fault and the last one is the normal situation and both fault situations. The number of map units in these different maps are 24 x 16, 29 x 13 and 29 x 16. Table 1 shows the number of used data points, the number of wrong states and the quality of state recognition (in %).

Table 1: Quality of state recognition with map trained with data from the normal and fault situations.

Used faults	Quality of state recognition	Training data	Testing data 1	Testing data 2
Normal + Spool fault	Data points	5496	1875	1892
	Wrong states	117	58	60
	Correct state %	97,9	96,9	96,8
Normal + Cylinder fault	Data points	5491	1883	1893
	Wrong states	425	140	193
	Correct state %	92,3	92,6	89,8
Normal + Both faults	Data points	7872	2676	2687
	Wrong states	582	217	244
	Correct state %	92,6	91,9	90,9

From the Table 1 can be seen that the used neural network method was able to recognize over 90% of the states correct for the training and testing data when both fault states were used. If only the data from the normal and spool fault measurements were used for the training the recognition percent was over 96%, for both training and testing data, which could be considered a very good result. So the worn spool of the pilot operated proportional valve was easier to recognize than the worn cylinder.

In [6] classification results were little better when supervised training was used for training the network. But here the size of the maps is determined differently and they are smaller which affects to the final result of the classification.

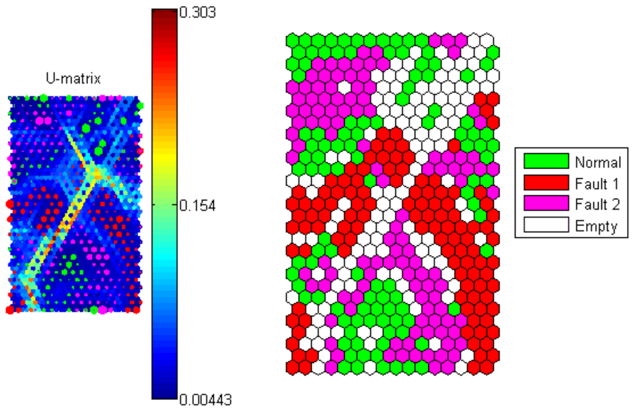

Fig. 8: Normal and two fault situations. U-matrix on the left and the distribution of neurons according to the state on the right.

Figure 8 shows the distribution of neurons in 2d-surface into clear clusters when normal and two fault states are used in training. On the left is shown the U-matrix which computes unified distance matrix and shows the clustering structure of the SOM. Also hits during the training are marked on the U-matrix. On the right is shown the distribution of neurons according to the state. This shows how the neurons are organized so that similar neurons are near and different ones far away each other, as was said in the earlier chapter [1, 4].

FAULT SITUATIONS ARE NOT KNOWN

In this case the basic presumption is that only the normal situation is known and the fault situations are not known. This is usually the case in real life.

Also here the training of the network was performed in two parts. In the first phase 4 and in the second phase 15 iteration rounds were used, when final quantization error (FQE) value was 0.0069 and topographic error 0.0960.

Here the training data consist of measurements from the normal situation. The map units which are hit during the training are labeled after training as a normal situation. After this a certain threshold is set. When the network is tested the distances between sample vectors from the testing data and all the codebook (weight) vectors of the SOM are calculated using Euclidian distance. If the minimum distance is bigger than beforehand set threshold value then this sample vector is treated as a fault state measurement.

After fault state has been found out the network can be trained again and use also this fault and normal situations or train a completely new network using only data from the new fault situation.

Three different testing data sets and also the training data are presented to the network after the training. Every one of these testing data sets includes two sequences of the measurements. The first one consists of the measurements from the normal situations but these are not used in training. The second one is the measurements from the fault 1 situation and the last one is the measurements from the fault 2 situation.

Table 2: Quality of state recognition with map trained with data from the normal situation.

Quality of state recognition	Training data	Normal	Fault 1	Fault 2
Data points	3115	2180	1587	1590
Wrong states	158	90	1504	729
Correct state %	94,9	95,9	5,2	54,2

Figure 9 shows the distribution of neurons in 2d-surface when only data from the normal situation is used in the training. On the left is shown the U-matrix which computes the unified distance matrix and shows the clustering structure of the SOM. Also the hits during the training are marked on the U-matrix. On the right is shown the distribution of neurons according to the state.

In Fig. 10-13 are shown the calculated minimum distances between the sample vectors of the testing data and all the codebook vectors of the SOM.

From Fig. 10-11 can be seen that the trained network recognizes data from the normal situation very well. Only few spikes can be noticed from the figure and the quality of the recognition is close to 95% in both cases.

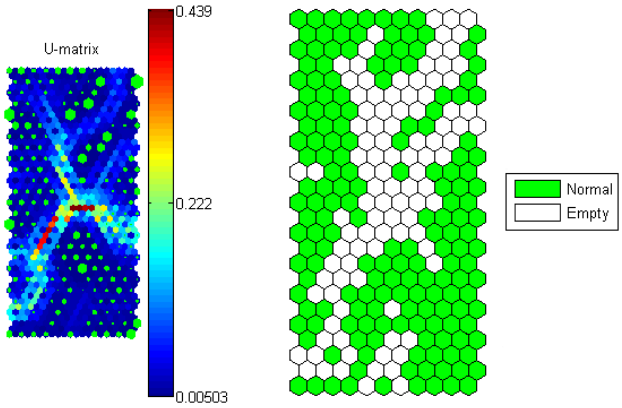

Fig. 9: Normal situation. U-matrix on the left and the distribution of neurons according to the state on the right.

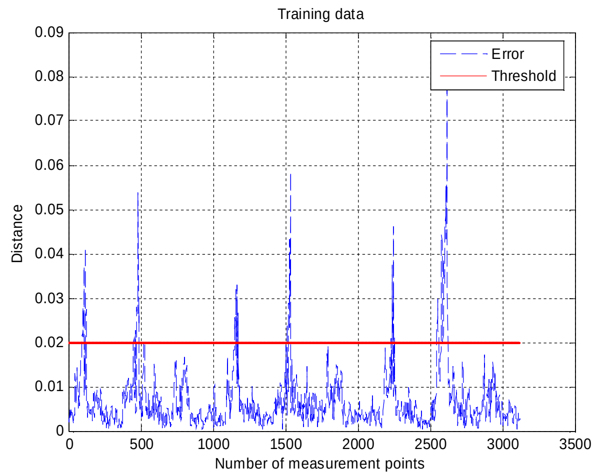

Fig. 10: Minimum distances between the sample vectors of the training data and all the codebook vectors.

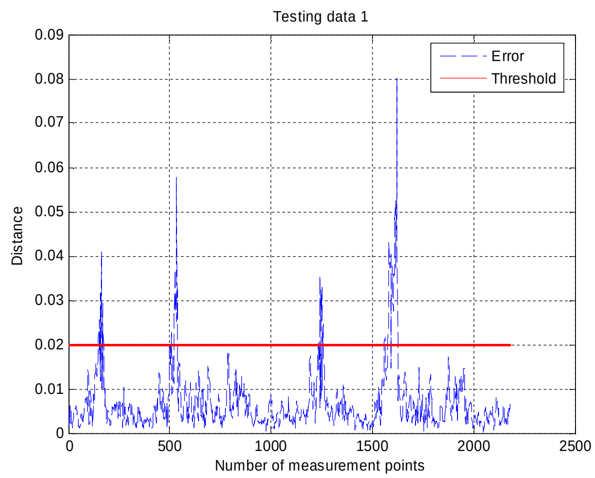

Fig. 11: Minimum distances between the sample vectors of the testing data 1 (normal) and all the codebook vectors.

The network is trained to detect normal situation and all other situations are fault situation. From Fig. 12-13 can be seen that the network detects fault situations from the testing data which is from the fault situation. With the fault 1 this is more obvious but also fault 2 can be detected. With fault 1 only 5.2% of the measurements are classified as a normal state and with fault 2 54.2%. So fault 2 is much harder to detect than fault 1.

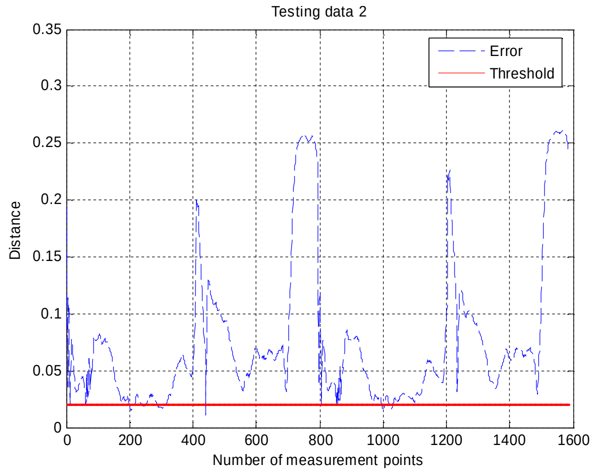

Fig. 12: Minimum distances between sample vectors of testing data 2 (fault 1) and all the codebook vectors.

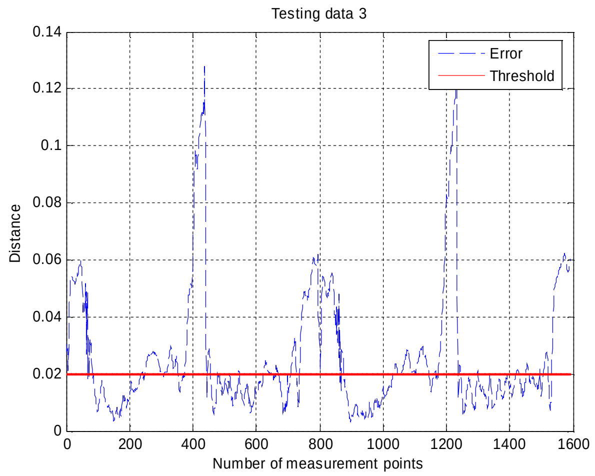

Fig. 13: Minimum distances between the sample vectors of the testing data 3 (fault 2) and all the codebook vectors.

CONCLUSION

The main goal of this research was to study a proactive condition monitoring system for fluid power systems where the neural network is used for fault detection. The faults used in this paper were the worn spool of the pilot operated proportional valve and the worn cylinder.

The classification problem was approached from two different viewpoints. The SOM was trained and tested with two different methods. In the first case the network is trained with data from the normal and fault situations. In the second case only normal situation data is used to train the network. This type of network could be used in situations where class information is not available as a detector of new events. In both cases the unsupervised training was used.

In the first case the SOM was able to recognize over 90% of the states correct, if both faults situations were used for the training and testing. If only the spool fault were used for the training the recognition percent was over 96%, which could be considered a very good result.

In the second case the network was tested with three different testing data sets and also with the training data after the training. The network classified 95% of the data as a normal state with the training data and the testing data set 1, which is from normal situation. With fault 1 only 5.2% and with fault 2 54.2% of the measurements are classified as a normal state.

The second fault (cylinder leakage) was in general harder to detect. The classification result of the second fault could be perhaps improved with different sensor selection and the location of the sensors. Both of these neural network training methods detected the presented the fault situations well although in the first case quality of the state recognition was much better with fault 2. But usually there is no fault situation data in the beginning of the training so the second case is closer to the real case when designing condition monitoring system.

Both of the classification cases included classification errors which have to take into account in the condition monitoring system, where the system do not react until the number of the fault states in certain time period exceeds beforehand set threshold value. The performance of the network is finally determined by how well it works with data that have not been trained to it. It does not bring any extra value to the condition monitoring system, if network works fine in a single trained situation. More important is the generalization capability of the network. In this study it was demonstrated that neural network method, Self-Organizing Maps (SOM), is suitable for detecting changes in the state of the fluid power system.

Using this kind of system it is possible to detect fault situations and control the state of the system through feedback information from the condition monitoring system. All the necessary maintenance and repairing work can be done before any major damage or even continue using the system at a lower utilization rate.

ACKNOWLEDGMENTS

This is an Author's Accepted Manuscript of SAE Technical Paper 2006-01-3492, published: 31 October 2006, copyright SAE International, available online at: http://papers.sae.org/2006-01-3492.

The research presented in this paper has been partly funded by KITARA (Information Technology in Mechanical, Civil and Automation Engineering) research program of the Academy of Finland.

REFERENCES

1. Alhoniemi, E.; Hollmén, J.; Simula, O. and Vesanto, J. 1999. Process Monitoring and Modeling Using the

Self-Organizing Map. *Integrated Computer-Aided Engineering,* Vol. 6, No. 1, pp. 3-14.
2. Crowther, W.; Edge, K.; Burrows, C.; Atkinson, R. and Woollons, D. 1998. Fault Diagnosis of a Hydraulic Actuator Circuit Using Neural Networks – A State Space Classification Approach. *Proc IMechE, Part I, Journal of Systems and Control Engineering,* Vol. 212.
3. Hagan, M.; Demuth, H. and Beale, M. 2002. *Neural Network Design*. PWS Publishing Company, Boston, 1st edition.
4. Kohonen, T. 2001. *Self-Organizing Maps.* Springer-Verlag Berlin Heidelberg, New York, 3rd edition.
5. Krogerus, T.; Vilenius, J.; Liimatainen, J.; Hyvönen, M. and Koskinen, K.T. 2005. Wireless Proactive Condition Monitoring of Pilot Operated Proportional Valve. *The Ninth Scandinavian Conference on Fluid Power, SICFP'05.* Linköping, Sweden.
6. Krogerus, T.; Vilenius, J.; Liimatainen, J. and Koskinen, K.T. 2006. Applying Self-Organizing Maps to Condition Monitoring of Fluid Power Systems. *4th FPNI-PhD Symposium Sarasota/ Florida 2006.* Florida, USA.
7. Kuravsky, L. and Baranov, S. 2001. Application of Self-Organizing Feature Maps for Diagnostics of Vibroacoustic Systems. *International Conference on Condition Monitoring.* St. Catherine's College, Oxford, UK.
8. Le, T.; Watton, J. and Pham, D. 1997. An artificial neural network based approach to fault diagnosis and classification of fluid power systems. *Proc IMechE, Part I, Journal of Systems and Control Engineering,* Vol. 206, 215-214.
9. Liangchai, Z.; Kuisheng, C. and Guozheng, S. 2003. Characteristic Curves Based Faulty Model Identification for Electro-Hydraulic Servo Valve Neural Network. *Fourth International Symposium on Fluid Power Transmission and Control, ISFP' 2003*. Beijing, China.
10. Liimatainen, J. 2006. *Dynamic User Interface for Proactive Condition Monitoring of Proportional Valve.* MSc thesis, Tampere University of Technology, Finland. (In Finnish)
11. Mathworks Inc. 2006. http://www.mathworks.com/ [Referred 28.4.2006]
12. Mundry, S. and Stammen, C. 2002**.** Condition Monitoring für die Fluidtechnik. *Beitrag in O+P Ölhydraulik & Pneumatik*, 46 Nr. 2.
13. Ramdén, T. 1998. *Condition Monitoring and Fault Diagnosis of Fluid Power Systems – A Approaches with Neural Networks and Parameter Identification.* PhD thesis No. 514. Linköping University, Sweden.
14. SOM Toolbox. 2006. http://www.cis.hut.fi/projects/somtoolbox/ [Referred 28.4.2006]
15. Sorsa, T. 1995. *Neural Network Approach to Fault Diagnosis*. PhD thesis No. 153. Tampere University of Technology, Finland.
16. Vesanto, J. 2002. *Data Exploration Process Based on the Self-Organizing Map*. PhD thesis, Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 115, Helsinki University of Technology, Finland.

CONTACT

Tomi Krogerus
Department of Intelligent Hydraulics and Automation
Tampere University of Technology
P.O.Box 589
FIN-33101 Tampere, Finland
E-mail: tomi.krogerus@tut.fi

DEFINITIONS

$\alpha(t)$	Learning rate at time t
σ	Neighbourhood radius
c	Best-Matching Unit (BMU)
$c(j)$	BMU of sample vector $\mathbf{x}_j$
d	Dimension of the input vectors
$h_{ci}(t)$	Neighborhood kernel around the winner unit at time t
I	Control current of the proportional valve [A]
K	Set of known (not missing) variables of sample vector $\mathbf{x}$
$\mathbf{M}$	Weight vector
M	Number of map units
m_k	k^{th} component of the weight vector
n	Number of sample vectors
n_{V_i}	Number of samples in the Voronoi set of unit i
p	Pressure [bar]
Q	Flow [l/min]
$\mathbf{r}_c$	Position of neuron c
S	Sum of the vectors in each Voronoi set
t	Time
T	Temperature [˚C]
w_k	k^{th} mask value
$\mathbf{x}$	Input vector
x_k	kth component of the sample vector