Proceedings of ESDA04
7th Biennial Conference on Engineering Systems Design and Analysis
July 19-22, 2004, Manchester, United Kingdom

ESDA2004-58197

NEURAL NETWORK MODELLING APPLIED FOR MODEL-BASED FAULT DETECTION

Zhanqun Shi, Yibo Fan, Fengshou Gu, Abdul-Hannan Ali & Andrew Ball
Manchester University, School of Engineering
Oxford Road, Manchester, M13 9PL, UK
Email: mbgmszs1@fs1.eng.man.ac.uk

Key Words: Model-based approach, fault detection, control system, neural networks, modelling

ABSTRACT

This paper aims to combine neural network modelling with model-based fault detection. An accurate and robust model is critical in model-based fault detection. However, the development of such a model is the most difficult task especially when a non-linear system is involved. The problem comes not only from the lack of concerned information about model parameters, but also from the inevitable linearization. In order to solve this problem, neural networks are introduced in this paper. Instead of using conventional neural network modelling, the neural network is only used to approximate the non-linear part of the system, leaving the linear part to be represented by a mathematical model. This new scheme of integration between neural network and mathematical model (NNMM) allows the compensation of the error from conventional modelling methods. Simultaneously, it keeps the residual signatures physically interpretable.

INTRODUCTION

Model-based fault detection and diagnosis has received considerable attention in recent years [1]. By comparing the model estimation and the real behaviour, the residual signal contains only fault or noise information of the monitored system [2]. Different methods can be used to generate and analyse the residuals. The most frequently used methods are observer-based methods, parameter estimation methods and parity space methods [3-6]. Because the residual is generated by comparing the model prediction and the real measurement, all model-based techniques rely heavily on the mathematical model. Unfortunately, the majority of applications use linear models or linearised models for those nonlinear systems [7]. Since there usually exists modelling errors under realistic conditions, no accurate or sufficiently accurate mathematical models can be obtained [8]. This is particular in the case with complex nonlinearity and the case that lacks adequate information. Consequently, the linearised model-based methods suffer from poor performances [9]. In order to overcome the problem in modelling accuracy, some efforts are made in the aspects such as non-linear observer development [10-11] and many neural networks [7, 12-13].

There are two purposes for introducing neural networks into the model-based fault approach. When there is a lack of analytical knowledge of the system, a neural network can approximate the input and output performance of the system and provide a non-parametric model. This non-parametric model can then replace the mathematical model to implement model-based fault detection and diagnosis, as used by [7, 14-15]. Besides, neural networks provide a convenient approach to solve the fault classification problem in the model-based approach [16]. However, although neural networks can be used both in modelling and in classification, many problems are not satisfied and need further research, especially the neural network modelling [3]. This paper focuses on this topic and presents a combination of a neural network model and model-based approach.

NEURAL NETWORK MODELLING

In recent years, neural networks have found more and more applications in system modelling and system control [7, 17]. Neural network modelling does not need any prior knowledge of the component or of the system. A trained neural network can approximate the behaviour of any system using the historic data of the system, whether linear or non-linear [18]. When used in fault diagnosis, the healthy inputs and outputs of a control system will be used to train the neural network and to represent a system with a non-parametric model. This model can consequently be used in a model-based approach.

For modelling purposes, an input-output structure shown in equation (1) is suitable for a non-linear system model. u represents the system input and y denotes system output,

Copyright © 2004 by ASME

Downloaded From: https://proceedings.asmedigitalcollection.asme.org on 06/28/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

and $\hat{y}$ denotes the output of neural network (NN) model, which approximates the output of the system.

$$\hat{y}(k) = f(u(k), u(k-1)\cdots u(k-m), y(k-1), y(k-2)\cdots y(k-n)) \quad (1)$$

Two determinations are necessary in this modelling. One is neural network topology, which is related to both the network complexity and the speed in implementation. The other is the number of time delay (m, n), which affects the network complexity. The topology is a dominant factor to the complexity therefore it is chosen before the determination of the number of time delay.

The Choice of Radial Basis Function Network

Among the variety of neural networks, the most used networks are multi-layer perceptron (MLP) networks and radial basis function (RBF) networks [19]. Both them are capable of approximating any nonlinear unique static function. Their difference is that the RBF networks possess good interpolation but the MLP networks possess good extrapolation abilities. The RBF network is chosen in this paper because of the advantages outlined below.

As shown in Fig. 1, an RBF network has a single hidden layer, and its output layer is merely a linear combination of the non-linear hidden layer signals. In contrast, an MLP network has at least two layers and all layers are no-linear [19]. Therefore, an RBF network allows for a much simpler weight updating procedure.

Another feature of the RBF network is its localised approximation to non-linear input-output mapping due to the use of an exponentially decaying structure (e.g. the Gaussian function). Besides, the neurons of the hidden layer can be increased according to the approximation requirement. This provides the RBF with the ability to model any non-linear function in a relatively straightforward way [19]. It has also been shown that given enough hidden neurons, an RBF network can approximate any continuous function with

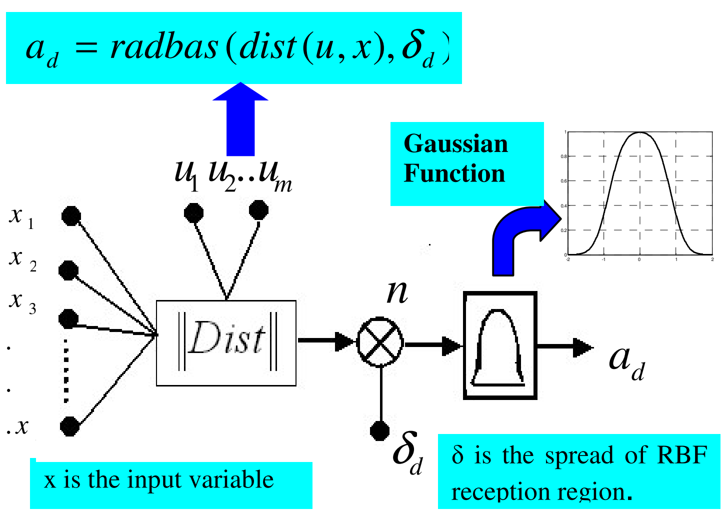

Figure 1: The structure of an RBF network

arbitrary accuracy. This is just the factor that is used to improve modelling accuracy. Moreover, the RBF network is faster in convergence than that of multi-layer perceptron networks [19]. This is because the simpler structure and the localised approximation mean that this type of network is capable of fast learning. Also, its rapid training makes it suitable for on-line implementation and model adaptation, which is desirable in this project.

The RBF network has poor extrapolation performance. This is to say that the RBF gives poor approximation when untrained data are processed. This drawback can be partially overcome by normalisation and regression.

Improvement of the RBF Network

As shown in Fig. 2, the RBF network can be improved by connecting to a special output layer. This network is called a generalised regression neural network (GRNN) and has three major advantages over the RBF network. Firstly, a special output layer as shown in Fig. 2 is added to the output layer. Each output value in the output layer is divided by the sum of all hidden layer outputs. This provides the network with an optimal normalisation function. The normalisation keeps the output within a specific range. Secondly, a generalised regression neural network is a method for estimating system behaviour given only one training set. Because the probability density function (pdf) is derived from the data with no preconception about its form, the network is perfectly general. This generalisation enables the inputs that are not identical to those encountered in the training set to be correctly classified. This property provides the generalised regression neural network with the capability to tolerate noises and some random factors from the system. Finally, a generalised regression neural network has the desirable property of requiring no iterative training. This is because the weights in this network are simply assigned to the target value directly from the training set

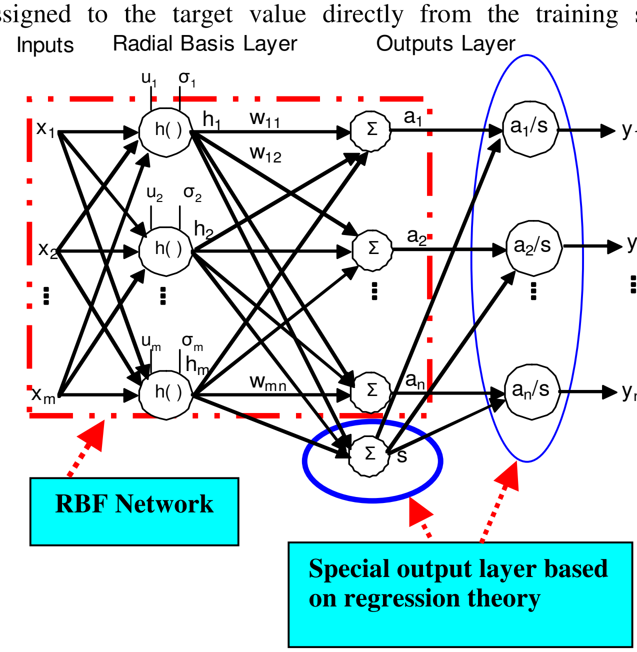

Figure 2: The structure of a generalised regression neural network

Copyright © 2004 by ASME

Downloaded From: https://proceedings.asmedigitalcollection.asme.org on 06/28/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

associated with the input training vector and its corresponding output vector. This property makes the training procedure simple and fast.

IMPLEMENTATION ON A CONTROL SYSTEM

System Description

The neural network model is applied on the modelling of a real system. As shown in Fig. 3, it is an electro-hydraulic servo system. The system consists of an actuator, an electro-hydraulic servo valve and a displacement transducer. The computer is used as a controller in this system to receive feedback signal from the transducer and to send out control commands to the electro-hydraulic servo valve, which control the flow rate through the valve to drive the actuator. The servo valve is such a high integrity unit that no structure parameters can be obtained by measurement. Furthermore, the fluid power part of it often behaves in a non-linear way. Therefore, an accurate mathematical model of it is very difficult to be developed although some performance information can be found in its specification. In consideration of these difficulties, a non-parametric model is developed using neural networks.

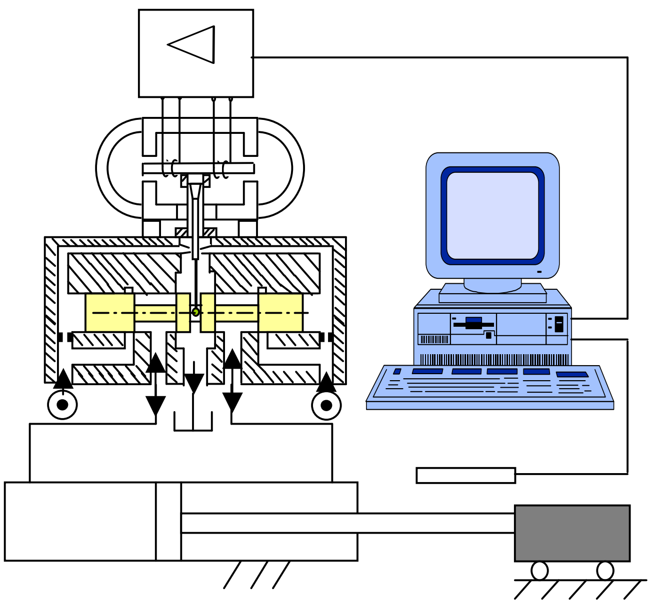

Figure 3: The control system to be monitored

Data Set Preparation

Theoretically, the training data set should include all possible operating conditions of the system. The movement of the actuator is forwards and backwards between 100mm and 200mm. The stroke of the actuator is arbitrary within 0~100mm. If all inputs and their responses are used to train the neural network, the data set will be extremely large, and this will make the number of hidden neurons unacceptably large.

As is well known, there is a receptive field for each neuron of an RBF network. This receptive field can handle the data around the neuron centre within this field. This is called generalisation capability of the neural network [19]. If the receptive field is enlarged, the generalisation capability will be improved, although the modelling accuracy will be slightly decreased. The neural network performance can also be improved by data set pre-processing, in which the input and target data can be normalised so that they always fall within a specific range, [-1 1] for instance. All these methods are used in conjunction with the application of neural networks to the monitored system.

In order to cover all possible situation of the system, operation, the measured data from the maximum stroke of the actuator (100mm) is used for normalisation. The normalised data are then used in the training of the network. During the training, the receptive field is optimised in accordance with the training data. The time delay in equation (1) is also needed to be determined during training. It is discussed below.

Determination of Time Delay

Theoretically, the number of time delay must be equal or greater than the system order [3]. In practice, the order of the system is not known beforehand, and therefore the time delay needs to be determined by a trial and error method.

Different numbers of time delay are tried. In equation (1), $m=0$ is tried first with $n=1$. As shown in Fig. 4(a), the maximum training error is as large as 4.04mm, and it does not decrease significantly as n increases to 2 and 3. Therefore, further increase of n is given up and instead of increasing m to 1. The maximum training error decreases to 0.35mm when m is 1 and n is either 2 or 3. When m increases to 2 with n being 3, the maximum training error decrease to 0.273mm. It seems that the increase of m can decrease the training error. However, Fig. 4(b) shows that the estimation error in real applications will increase when n is larger than 3 and m is larger than 2. This means that further increase of time delay can

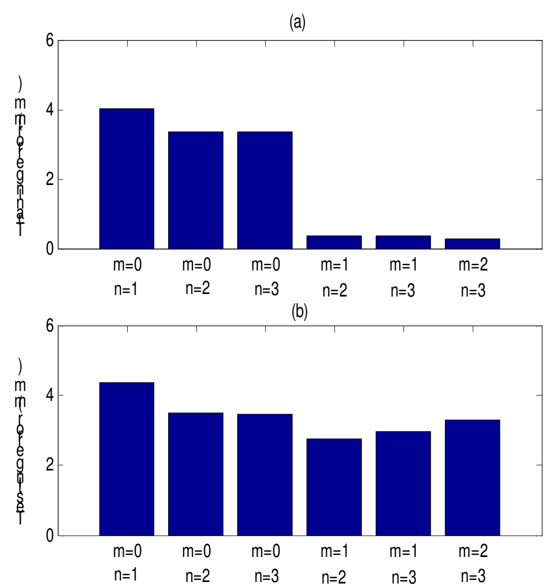

Figure 4: The estimation errors in different time delay

Copyright © 2004 by ASME

Downloaded From: https://proceedings.asmedigitalcollection.asme.org on 06/28/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

not decrease the modelling accuracy but decrease the capability in data generalisation. Besides, the increase of time delay also increases the network complexity. Therefore, the optimal choice of the time delay is $m=1$ *and* $n=2$. This model structure is used in conjunction with an RBF network to approximate the monitored system.

COMBINATION OF NN MODEL WITH MATHEMATICAL MODEL (NNMM)

Conventional Application

After training, the neural network is used as a model, the output of which can be used in model-based fault detection. The general scheme of NN-model based approach is given by literature [4]. As shown in Fig. 5, a neural network replaces the mathematical model to generate the residual. To fulfil this task, the network has to be trained under normal condition, as illustrated in Fig. 5(a). After finishing the training, the neural network output can be used for comparison with the real measurement to generate residual signals. Figure 5(b) illustrates this procedure, which is similar to what happens in a conventional model-based fault detection approach. In this paper, this is called a conventional NN model-based approach. Following examples are used to evaluate the effectiveness of this approach.

Figure 6(a) shows the comparison of the NN model estimation and the measurement of the system under normal conditions in 100% stroke (100mm). Figure 6(b) shows their corresponding residual. The normal residual is within ±1mm, which seems to be acceptable.

However, when this trained network is used for an 80% stroke of the actuator (80mm), the network output deviates from the correct value. Figure 7 shows this situation. The modelling error is more than 20mm. This implies that the

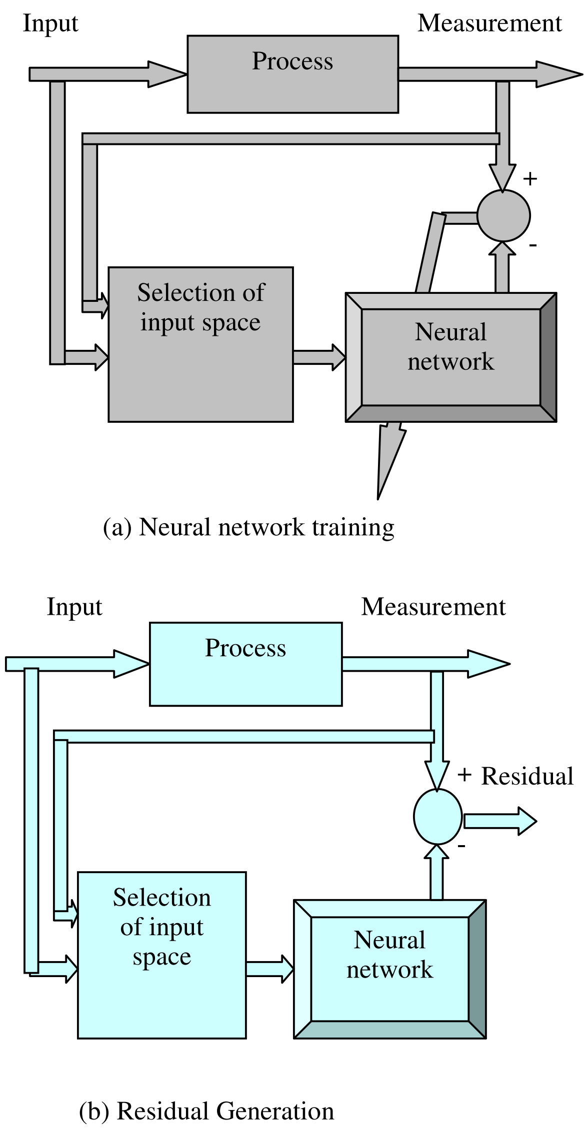

Figure 5 Training and application of NN for residual generation

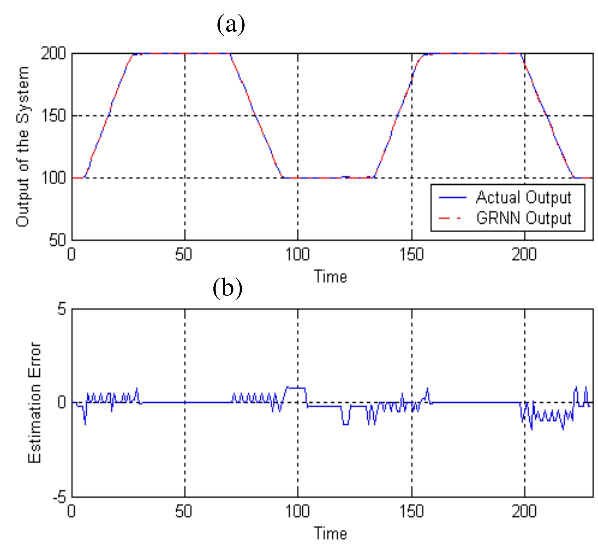

Figure 6: NN model error in maximum stroke

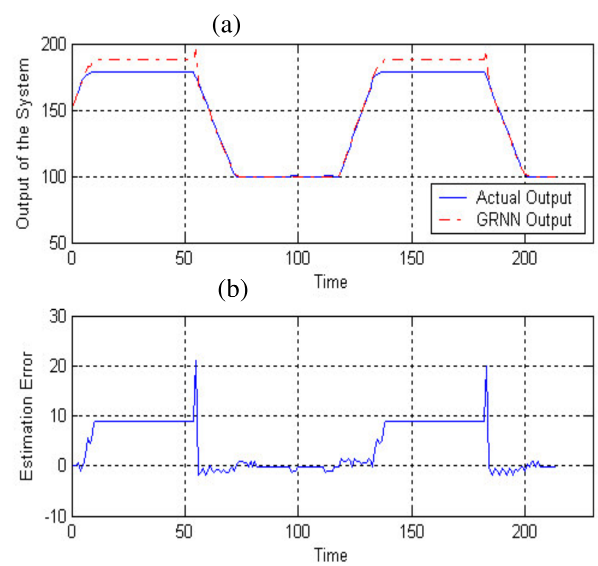

Figure 7: Estimation error for an 80% stroke

Copyright © 2004 by ASME

Downloaded From: https://proceedings.asmedigitalcollection.asme.org on 06/28/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

network is not reliable in all system operations. The reason is that the real input, although within 0~100mm, is going outside of the receptive field.

The situation gets even worse when the stroke is further deduced. Enlarging receptive field does not give any help unless all data set from different strokes are taken into training. If all data in the scope of the overall operation are used for network training, the modelling accuracy may be adequate but the complexity of the network will be unaffordable and the network will work slowly due to this complexity. Obviously, this network model will cause a false alarm or time delay if it is used in model-based fault detection.

The Scheme of NNMM and its application

In the conventional application, neural network models replace the mathematical model completely. This type of non-parametric model relies heavily on the training data. As demonstrated later, the training data should cover the overall input scope of the system under investigation. This makes the structure of the neural network model very complex. Although the network can be simplified by dividing the overall input scope into small pieces according to the receptive fields, a look-up table or another layer of neural network is needed in application to identify the input and determine which piece of NN-model should be used. In this paper, a new scheme of integrating the neural network with the mathematical model is developed. In this approach, only one set of system input and its response are necessary in the network training but the modelling is more accurate than that of a pure NN model.

Figure 8 shows the new scheme of integration. The integration of the neural network with the output observer is based on the following considerations. Firstly, although the mathematical model is not accurate enough, it is more robust than a neural network. A mathematical model can respond to arbitrary system inputs, but a neural network can only give a relative accurate respond to those system inputs that are covered by the training data set. Secondly, the generalised regression neural network has high capability to approximate a

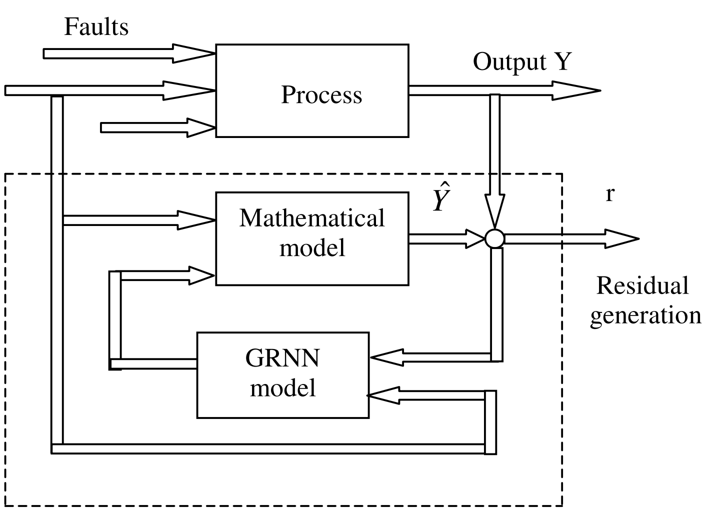

Figure 8: NNMM-based approach for residual generation

non-linear system. This performance of the generalised regression neural network can be used to compensate for the inaccuracy of the mathematical model. Thirdly, the modelling error of a mathematical model is much less than that of the model output itself. If it is normalised and used in network training, the receptive field of the network provides an acceptable result. This network model is only used for modelling the inaccurate part of the mathematical model. Thus, the mathematical model can be linearised and simplified and leave the non-linearity and modelling error to be approximated by the generalised regression neural network. A NN model is provided in this way. This novel idea is represented in equations (2) and (3) and also illustrated in Fig. 8.

$$\dot{\hat{x}} = A\hat{x} + Bu + f(u(k), u(k-1), r(k-1), r(k-2)) \qquad (2)$$

$$\hat{y} = C\hat{x} \qquad (3)$$

where $r(k) = f(u(k), u(k-1), r(k-1), r(k-2))$ is the prediction error of the generalised regression neural network.

The model accuracy will be improved significantly. The effectiveness is shown in Fig.9. An 80% stroke is used to evaluate its modelling accuracy. The modelling error is between [-2 1] mm. This accuracy can be maintained even when a 20% stroke is operated. Comparing Fig. 9 with Fig. 7 shows that the modelling error of a NNMM is only 5% of that of a pure NN model. This means that the NNMM is more accurate than the pure generalised regression neural network model and is more suitable for model-based fault detection.

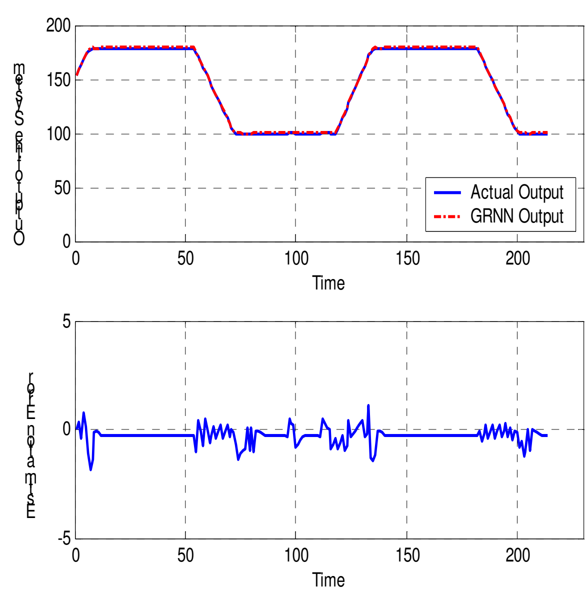

Figure 9: Modelling accuracy of NNMM with an 80% stroke

Copyright © 2004 by ASME

Downloaded From: https://proceedings.asmedigitalcollection.asme.org on 06/28/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

IMPLEMENTATION TO MODEL-BASED FAULT DETECTION

The new developed approach is applied to fault detection of the electro-hydraulic control system as shown in Fig. 3. An actuator leakage is simulated in the system.

Figure 10 shows the residuals under normal condition but generated by different approaches. The solid line shows the NNMM model-based results and the dashed line shows the result of the mathematical model-based approach. The modelling error of the NNMM is within the range of [-1.5 1.5] mm and is almost a random signal, whilst the modelling error of the mathematical model varies significantly with a peak amplitude as high as [-2.8 2.8] mm. The comparison in modelling errors indicates that the NNMM is more accurate than the mathematical model.

Recalling the modelling accuracy comparison that is made in figures 9 and 7, a NNMM is more accurate than either a pure NN model or a conventional mathematical model.

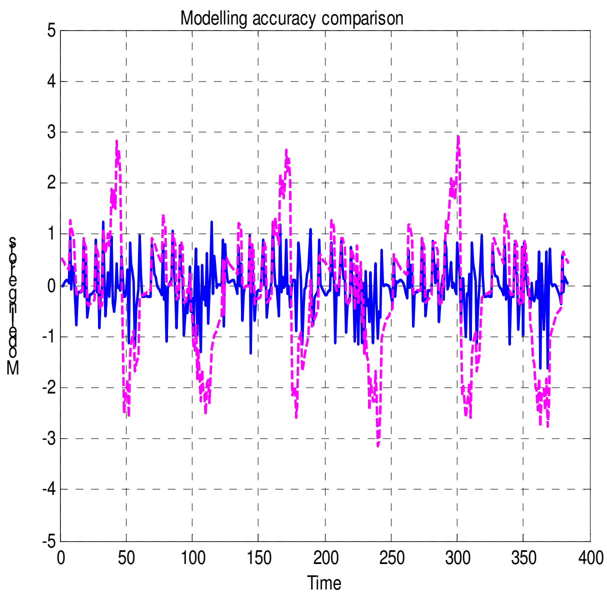

Figure 10: Modelling accuracy comparison

Figure 11 shows the sensitivity of NNMM-based approach comparing to that of a mathematical model-based approach. Both residuals are generated under a fault condition of a small leakage in the actuator. g residual comparing to that of conventional mathematical model-based approach. The NNMM-based residual (Solid line) gives out higher peak values during the transient period than the mathematical model does (Dashed line). The threshold is set to ±5mm, and clearly it is exceeded by the residual form NNMM-based approach in some periods (transient periods), but not touched by the residual from the mathematical model-based approach. This means that the NN model-based approach is more sensitive to faults than a mathematical model.

CONCLUSION

In this paper, a new scheme of using NNMM for model-based fault detection is developed combining the conventional mathematical model with a generalised regression neural network (GRNN). Rather than using the neural network to model the whole system, this scheme uses it only to model the nonlinear portion that can not be approximated well by linearised mathematical models. Its modelling accuracy is compared to that of both a pure NN model and a mathematical model. It can be concluded that the NNMM is more accurate than either of them. The fault sensitivity of this new scheme is evaluated by an incipient fault of an actuator leakage. Compared with the conventional model-based approach, the NNMM is more sensitive to the incipient fault. The detection results show that it is superior over either a pure NN model or a mathematical model in the aspects of accuracy, robustness and effectiveness when applied to model-based fault detection and diagnosis.

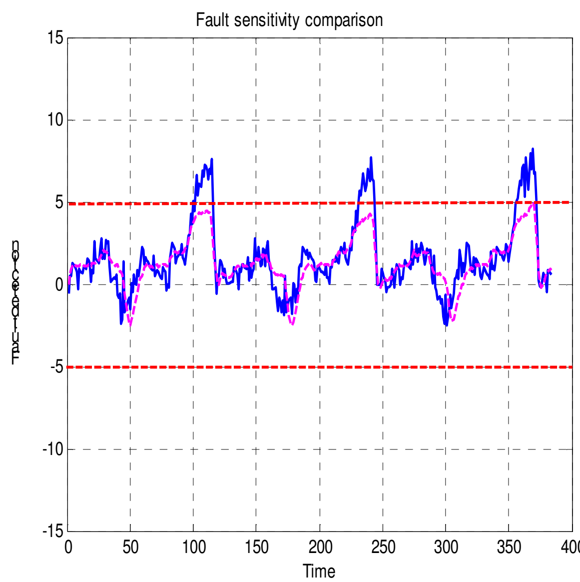

Figure 11: Sensitivity comparison

REFERENCES

[1] Simani, S., Fantuzzi, C., & Patton, R. J., 2003, *Model-based fault diagnosis in dynamic systems using identification techniques,* Springer, London.

[2] Chen, J., & Patton, R. J., 1999, *Robust model-based fault diagnosis for dynamic systems,* Kluwer Academic Publisher, London.

[3] Frank, P.M., Ding, S.X., & Marcu, T., 2000, “Model-based fault diagnosis in technical processes”, Trans. of the Institute of Measurement and Control, 22(1), pp57-101.

[4] Isermann, R., & Balle, P., 1997, “Trends in the application of model-based fault detection and diagnosis of technical processes”, Control Engineering Practice, 5(5), pp709-719.

Copyright © 2004 by ASME

Downloaded From: https://proceedings.asmedigitalcollection.asme.org on 06/28/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use

[5] Patton, R. J., 1991, “A review of parity space approaches to fault diagnosis”, *Proc of IFAC symposium*, Baden-Baden, 1991, pp65-81.

[6] Gertler, J., 1998, *Fault detection and diagnosis in Engineering systems*. Marcel Dekker, New York.

[7] Alessandri, A., 2003, “Fault diagnosis for non-linear systems using a bank of neural estimators”, Computers in Industry, 52, pp271-289.

[8] Frank, P. M, & Köppen-seliger, B., 1997, “New developments using AI in fault diagnosis”, Engineering Application of Artificial Intelligence, 10(1), pp3-14.

[9] Rengaswamy, R., *et al*., 2001, “A comparison of model-based and neural network-based diagnostic methods”, Engineering Application of Artificial Intelligence, 14, pp805-818.

[10] Tan, H., & Sepehri, N., 2002, “Parametric fault diagnosis for electro-hydraulic cylinder drive units”, IEEE Trans. on Industrial Electronics, 49(1), pp96-106.

[11] Janusz, M. E., & Venkatasubramanian, V., 1991, “Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis”, Engineering Application of Artificial Intelligence, 4(5), pp329-339.

[12] Isermann, R., & Nelles, O., 1997, “Identification with dynamic neural networks –architectures, comparisons, application”, *Proc of the IFAC Symposium on System Identification*, Fukuoka, Vol. 3, pp997-1022.

[13] Thomas, P., & Lefebvre, D., 2002, “Fault detection and isolation in non-linear systems by using oversized neural networks”, Mathematics and Computers in Simulation, 60, pp181-192.

[14] Yu, D. L., Gomm, J.B., & Williams, D., 1999, “Sensor fault diagnosis in a chemical process via RBF neural networks”, Control Engineering Practice, 7, pp49-55.

[15] NØrgaard, M., *et al*., 2000*, Neural networks for modelling and control of dynamic systems*. Springle-Verlag, London.

[16] Simani, S. & Fantuzzi, C., 2000, “Fault diagnosis in power plant using neral networks”, Information Sciences 127, pp125-136

[17] Marki, Y., & Loparo, K.A., 1997, “A neural network approach to fault detection and diagnosis in industrial processes”, IEEE Trans. On Control System Technology, 5(6), pp529-541.

[18] Wilson, D.J.H., Irwin, G.W., & Lightbody, G., 1999, “RBF principal manifold for process monitoring”, IEEE Trans. on Neural Networks, 10(6), pp1424-1434.

[19] Irwin, G. W., Warwick, K., & Hunt, K. J., 1995, *Neural network applications in control*. Short Run Press Ltd., UK.

Copyright © 2004 by ASME

Downloaded From: https://proceedings.asmedigitalcollection.asme.org on 06/28/2019 Terms of Use: http://www.asme.org/about-asme/terms-of-use