Computer Assisted Mechanics and Engineering Sciences, 12: 279–291, 2005.

Designing neural-network-based fault detection systems with D-optimum experimental conditions

Marcin Witczak, Przemysław Prętki
Institute of Control and Computation Engineering,
University of Zielona Góra,
ul. Podgórna 50, 65–246 Zielona Góra, Poland.

(Received March 31, 2005)

The paper deals with an application of the theory of optimum experimental design to the problem of selecting the data set for developing neural models. Another objective is to show how to design a robust fault detection scheme with neural networks and how to increase its fault sensitivity by decreasing model uncertainty. It is also shown that the optimum design is independent of the parameters that enter linearly into the neural network. The final part of this paper shows a comprehensive simulation study regarding modelling and fault detection with the proposed approach. In particular, the DAMADICS benchmark problem is utilized to verify the performance and reliability of the proposed technique.

1. Introduction

In spite of its incontestable appeal, the strategy of taking measurements for parameter estimation of neural networks [11] according to optimum experimental design (OED) theory [1, 16, 15] has been studied in extremely few works so far [3, 9, 10, 19]. Fukumizu [9, 10] developed the so-called statistical active learning technique based on the general theory of OED. He also developed a technique that allows obtaining the non-singular Fisher Information Matrix (FIM) [8]. These conditions are very important because if at least one of them holds then it is impossible to use the FIM with the classical algorithms of OED, e.g. the Wynn–Fedorov algorithm [1, 16]. In [3], the authors employed a D-optimum experimental design [1, 16] obtained for a quadratic regression model for collecting data for a neural network. The main drawback of this approach is the fact that it can be used for very restricted design spaces. An obvious remedy for such a problem is to develop an algorithm that utilizes an exact structure of the neural network rather than a polynomial of the second degree. In [19], the author proposed a sequential experimental design algorithm that allows obtaining a one-step-ahead D-optimum input. This algorithm can be perceived as a hybrid one since it can be used for both training and data development. Moreover, the author proposed a fault detection scheme for neural networks that is also used in this paper.

A fault detection process can be perceived as a two-stage procedure, i.e. residuals generation and symptom evaluation based on these residuals [2, 12]. Under the assumption of a perfect mathematical description of the systems being considered, a perfect residual generation should provide a residual that is zero during the normal operation of the system and considerably different than zero otherwise. This means that the residual should ideally carry information regarding a fault only. Under such assumptions, faults can be easily detected. Unfortunately, this is impossible to attain in practice since residuals are normally uncertain, corrupted by noise, disturbances and modelling uncertainty. That is why, in order to avoid false alarms, it is necessary to assign a threshold to the residual that is significantly larger than zero. The most common approach is to use a fixed threshold [2, 12, 17]. The main difficulty with this kind of thresholds is that they may cause many serious problems regarding false alarms as well as undetected faults. In other words, it is very difficult to fix such a threshold

and there is no optimal solution that can be applied to settle such a task. One way out of this difficult problem is to use robust residual generators that aim at decoupling model uncertainty from the residual [2, 12, 17]. When such techniques are applied it is much easier to select an appropriate fixed threshold providing reliable fault detection. Undoubtedly, the most common choice is to use robust observers, such as the Unknown Input Observer (UIO) [2, 7, 17, 18]. In such observers, the model-reality mismatch is represented by the so-called unknown input and hence the state estimation error and, consequently, residuals are obtained taking into account model uncertainty. This means that the effect of an unknown input on the residual is minimized. The main difficulty is that there are situations for which the direction of faults is very similar to that of an unknown input. This may lead to a situation in which the effect of some faults is minimized and hence they may be impossible to detect. Other approaches that make use of the idea of an unknown input also inherit this drawback, e.g. robust parity relation [2].

If the above-mentioned approach fails, then describing model uncertainty in a different way seems to be a good remedy. One of the possible approaches is to use statistical techniques [1, 16] (for an example regarding different approaches the reader is referred to [6]) to obtain parameter uncertainty of the model and, consequently, model output uncertainty. Such parameter uncertainty is defined as the parameter confidence region [1, 16] containing a set of admissible parameters that are consistent with the measured data. Thus it is evident that parameter uncertainty depends on measurement uncertainty, i.e. noise, disturbances, etc.

The knowledge about parameter uncertainty makes it possible to design the so-called adaptive threshold [7]. The adaptive threshold, contrary to the fixed one (cf. Fig. 1), bounds the residual at a level that is dependent of model uncertainty, and hence it provides more reliable fault detection.

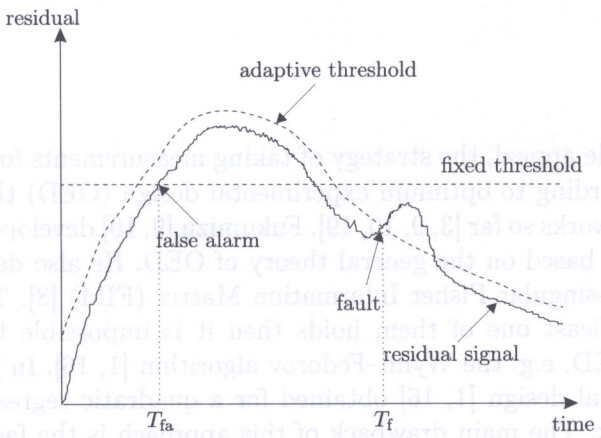

Fig. 1. Principle of an adaptive threshold

The paper is organized as follows: Sec. 2 proposes to use a neural-network-based technique that makes it possible to design an adaptive threshold. It is also shown that the size of residual bounds generated with the proposed technique is strongly related with measurement schedule that is used while developing the neural model, i.e. it is shown that the application of OED may decrease these bounds. Section 3 presents important properties regarding OED applied to neural networks. These properties are very useful while using OED in practice. Finally, the last part is devoted to an illustrative example.

2. Motivation and preliminaries

Let us consider a feed-forward neural network given by the following equation:

$$y_{m,k} = \mathbf{g}\left(\mathbf{P}^{(n)}\mathbf{u}_k\right)^{\mathrm{T}}\mathbf{p}^{(l)}, \tag{1}$$

where $y_{m,k} \in \mathbb{R}$ stands for the model output, $\mathbf{g}(\cdot) = [g_1(\cdot), \ldots, g_{n_h}(\cdot), 1]^{\mathrm{T}}$, where $g_i(\cdot) = g(\cdot)$ is a nonlinear differentiable activation function,

$$\mathbf{P}^{(n)} = \begin{bmatrix} \mathbf{p}^{(n)}(1)^{\mathrm{T}} \\ \vdots \\ \mathbf{p}^{(n)}(n_h)^{\mathrm{T}} \end{bmatrix} \tag{2}$$

and $\mathbf{p}^{(l)} \in \mathbb{R}^{n_h+1}$ are the matrix and the vector, respectively, representing the parameters (weights) of the model, and n_h is the number of neurons in the hidden layer. Moreover, $\mathbf{u}_k \in \mathbb{R}^{n_r+1}$, $\mathbf{u}_k = [u_{1,k}, \ldots, u_{n_r,k}, 1]^{\mathrm{T}}$, where $u_{i,k}$, $i = 1, \ldots, n_r$ are the system inputs. For the sake of notational simplicity, let us define the following parameter vector:

$$\mathbf{p} = \left[\left(\mathbf{p}^{(l)}\right)^{\mathrm{T}} v, \mathbf{p}^{(n)}(1)^{\mathrm{T}}, \ldots, \mathbf{p}^{(n)}(n_h)^{\mathrm{T}} \right]^{\mathrm{T}},$$

where $\mathbf{p} \in \mathbb{R}^{n_p}$, $n_p = n_h(n_r + 2) + 1$. Consequently, Eq. (1) can be written in a more compact form:

$$y_{m,k} = f(\mathbf{p}, \mathbf{u}_k), \tag{3}$$

where $f(\cdot)$ is a nonlinear function representing the structure of a neural network.

Let us assume that the system output satisfies the following equality:

$$y_k = y_{m,k} + \epsilon_k = f(\mathbf{p}, \mathbf{u}_k) + \epsilon_k, \tag{4}$$

where ϵ is i.i.d and $\mathcal{N}(0, \sigma^2)$. It can be shown [1, 16] that the bounds ($100(1-\alpha)$ confidence interval) of the system output can be approximated as follows:

$$y_k^m \leq y_k \leq y_k^M, \tag{5}$$

$$y_k^m = f(\hat{\mathbf{p}}, \mathbf{u}_k) - t_{n_t-n_p}^{\alpha/2} \hat{\sigma} \left(1 + \mathbf{r}_k^{\mathrm{T}} \mathbf{P} \mathbf{r}_k\right)^{1/2}, \tag{6}$$

$$y_k^M = f(\hat{\mathbf{p}}, \mathbf{u}_k) + t_{n_t-n_p}^{\alpha/2} \hat{\sigma} \left(1 + \mathbf{r}_k^{\mathrm{T}} \mathbf{P} \mathbf{r}_k\right)^{1/2}, \tag{7}$$

where $\hat{\mathbf{p}}$ stands for a least-square estimate of $\mathbf{p}$, n_t denotes the number of input-output measurements used for parameter estimation, $t_{n_t-n_p}^{\alpha/2}$ is the t-Student distribution quantile, $\hat{\sigma}^2$ is the variance estimate, $\mathbf{r}_k = \partial f(\mathbf{p}, \mathbf{u}_k)/\partial \mathbf{p}$, and $\mathbf{P}^{-1}$ is the Fisher information matrix defined as

$$\mathbf{P}^{-1} = \sum_{k=1}^{n_t} \mathbf{r}_k \mathbf{r}_k^{\mathrm{T}}. \tag{8}$$

As shown in [4], the above technique can be effectively applied to confidence interval prediction of neural network models. In this paper, it is proposed to use (5) for the purpose of robust fault detection. If the bounds (5) are given, then the adaptive threshold for the residual $z_k = y_k - y_{m,k}$ is given by

$$-t_{n_t-n_p}^{\alpha/2} \hat{\sigma} \left(1 + \mathbf{r}_k^{\mathrm{T}} \mathbf{P} \mathbf{r}_k\right)^{1/2} \leq z_k \leq t_{n_t-n_p}^{\alpha/2} \hat{\sigma} \left(1 + \mathbf{r}_k^{\mathrm{T}} \mathbf{P} \mathbf{r}_k\right)^{1/2}. \tag{9}$$

Consequently, the decision logic can be realized as follows:

If the residual $z_k = y_k - y_{m,k}$ satisfies (9), then there is no fault symptom, else (9) indicates that a fault symptom occurs.

One of the main advantages of the proposed approach is the fact that it is possible to increase fault sensitivity by decreasing parameter uncertainty of (1). This can be achieved with the application of the theory of experimental design [1, 16]. Indeed, it is easy to see that the length of the

confidence interval (5) is strongly related with the Fisher information matrix (8), which depends on the experimental conditions $\xi = [\mathbf{u}_1, \ldots, \mathbf{u}_{n_t}]$. Thus, optimal experimental conditions can be found by choosing $\mathbf{u}_i$, $i = 1, \ldots, n_t$, so as to minimize some scalar function $\Phi(\cdot)$ of (8). Such a function can be defined in several different ways [1, 16]:

- D-optimality criterion:

$$\Phi(\mathbf{P}) = \det \mathbf{P}, \tag{10}$$

- E-optimality criterion ($\lambda_{\max}(\cdot)$ stands for the maximum eigenvalue of its argument):

$$\Phi(\mathbf{P}) = \lambda_{\max}(\mathbf{P}), \tag{11}$$

- A-optimality criterion:

$$\Phi(\mathbf{P}) = \operatorname{trace} \mathbf{P}, \tag{12}$$

- G-optimality criterion:

$$\Phi(\xi) = \max_{\mathbf{u} \in \mathbb{U}} \phi(\xi, \mathbf{u}), \tag{13}$$

where

$$\phi(\xi, \mathbf{u}) = \mathbf{r}_k^{\mathrm{T}} \mathbf{P} \mathbf{r}_k. \tag{14}$$

A D-optimum design minimizes the volume of the confidence ellipsoid approximating the feasible parameter set of (1). An E-optimum design minimizes the length of the largest axis of the same ellipsoid. An A-optimum design suppresses the average variance of parameter estimates. A G-optimum design minimizes the variance of the estimated response of (1). Bearing in mind the fact that the primary purpose of this work is to develop a reliable technique for designing an adaptive threshold, it is clear from (9) that the G-optimality criterion should be selected.

When some experiments are repeated, the number n_e of distinct $\mathbf{u}_i$'s is smaller than the total number of observations n_t. The design resulting from this approach is called the continuous experimental design. The Fisher information matrix can then be written as

$$\mathbf{P}^{-1} = \sum_{k=1}^{n_e} \mu_k \mathbf{r}_k \mathbf{r}_k^{\mathrm{T}}, \tag{15}$$

where $\mu_k = w_k / n_t$, w_k is the number of repetitions of measurements under the experimental condition $\mathbf{u}_k$. The fundamental property of continuous experimental design is the fact that the D-optimality and G-optimality criteria are equivalent (the Kiefer–Wolfowitz equivalence theorem [1, 16]). Another reason for using D-optimum design is that it is probably the most popular criterion. Indeed, most of the algorithms that are presented in the literature are developed for D-optimum design. Bearing in mind all of the above-mentioned circumstances, the subsequent part of this paper is devoted to D-optimum experimental design. First, let us consider a neuron model with the logistic activation function:

$$y_{m,k} = \frac{p_1}{1 + e^{-p_2 u_k - p_3}}. \tag{16}$$

It is obvious that continuous experimental design for the model (16) should have at least three different support points ($n_p = 3$ for (16)). For a three-point design, the determinant of the Fisher information matrix (15) is

$$\det \mathbf{P}^{-1} = \frac{p_1^4}{p_2^2} \mu_1 \mu_2 \mu_3 e^{2x_1} e^{2x_2} e^{2x_3} \cdot \frac{((e^{x_2} - e^{x_1})x_3 + (e^{x_3} - e^{x_2})x_1 + (e^{x_1} - e^{x_3})x_2)^2}{(e^{x_1} + 1)^4 (e^{x_2} + 1)^4 (e^{x_3} + 1)^4}, \tag{17}$$

where $x_i = p_2 u_i + p_3$. Knowing that the minimization of (10) is equivalent to the maximization of (17), a numerical solution regarding D-optimum continuous experimental design can be written as

$$\xi = \left\{ \begin{array}{ccc} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ \mu_1 & \mu_2 & \mu_3 \end{array} \right\}$$

$$= \left\{ \begin{array}{ccc} \left(\dfrac{1.041 - p_3}{p_2}, 1\right) & \left(\dfrac{-1.041 - p_3}{p_2}, 1\right) & \left(\dfrac{x_3 - p_3}{p_2}, 1\right) \\ \dfrac{1}{3} & \dfrac{1}{3} & \dfrac{1}{3} \end{array} \right\}, \tag{18}$$

where x_3 is an arbitrary constant satisfying $x_3 \geq \zeta$, $\zeta \approx 12$. In order to check if the design (18) is really D-optimum, the Kiefer–Wolfowitz equivalence theorem [1, 16] can be employed. In the light of this theorem the design (18) is D-optimum when

$$\phi(\mathbf{u}_k) = \mathbf{r}_k^{\mathrm{T}} \mathbf{P} \mathbf{r}_k \leq n_p, \tag{19}$$

where the equality holds for measurements described by (18). It can be seen from Fig. 2 that the design (18) satisfies (19). In order to justify the effectiveness of (18), let us assume that the nominal parameter vector is $\mathbf{p} = [2, 0.5, 0.6]^{\mathrm{T}}$. It is also assumed that $n_t = 9$. This means that each of the measurement, consistent with (18) should be repeated 3 times. For the purpose of a comparison, a set of n_t points was generated according to the uniform distribution $\mathcal{U}(-4, 40)$. It should also be pointed out that ϵ was generated according to $\mathcal{N}(0, 0.1^2)$. Figure 3 presents feasible parameter sets obtained with the strategies considered.

These sets are defined according to the following formulae [16]:

$$\mathbb{P} = \left\{ \mathbf{p} \in \mathbb{R}^{n_p} \Big| \sum_{i=1}^{n_t} (y_i - f(\mathbf{p}, \mathbf{u}_i))^2 \leq \sigma^2 \chi^2_{\alpha, n_t} \right\}, \tag{20}$$

where χ^2_{α, n_t} is the Chi-square distribution quantile. This simple example exhibits serious advantages that can be gained while applying OED theory to a more complex model (1).

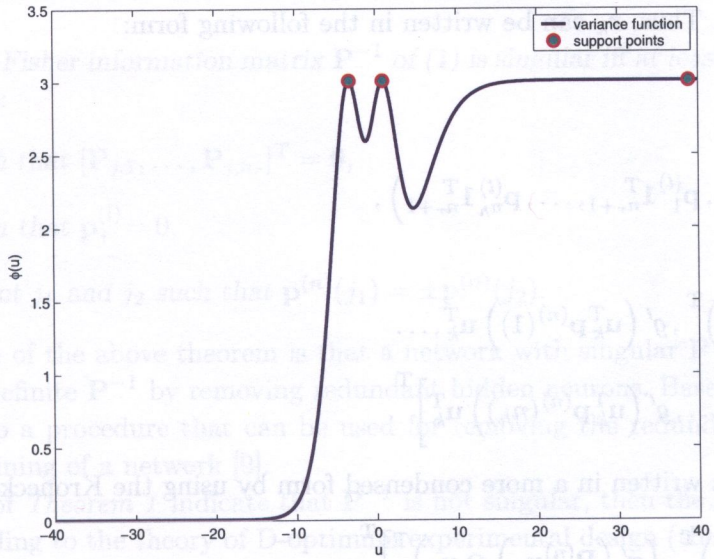

Fig. 2. Variance function for (18) and $\zeta = 20$

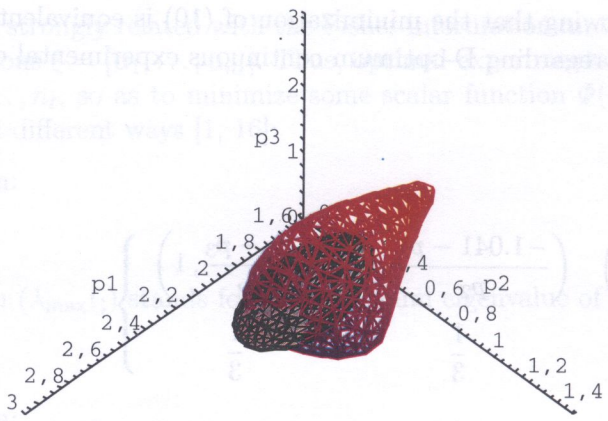

Fig. 3. Feasible parameter set obtained for (18) (smaller) and for a set of randomly generated points

3. Properties and algorithms

3.1. Independence of linear parameters

The dependence on parameters that enter nonlinearly (18) into the model is an undesirable characteristic of nonlinear optimum experimental design. In this section, it is shown that experimental design for a general structure (1) is independent of the parameters that enter linearly into (1). First, let us obtain $\mathbf{r}_k = \dfrac{\partial f(\mathbf{p}, \mathbf{u}_k)}{\partial \mathbf{p}}$:

$$\mathbf{r}_k = \left[\frac{\partial y_{m,k}}{\partial \mathbf{p}^{(l)}}^{\mathrm{T}}, \frac{\partial y_{m,k}}{\partial \mathbf{p}^{(n)}(1)}^{\mathrm{T}}, \ldots, \frac{\partial y_{m,k}}{\partial \mathbf{p}^{(n)}(n_h)}^{\mathrm{T}} \right]^{\mathrm{T}}, \tag{21}$$

where

$$\frac{\partial y_{m,k}}{\partial \mathbf{p}^{(l)}} = \mathbf{g}\left(\mathbf{P}^{(n)}\mathbf{u}_k\right), \tag{22}$$

$$\frac{\partial y_{m,k}}{\partial \mathbf{p}^{(n)}(i)} = \mathbf{p}_i^{(l)} g'\left(\mathbf{u}_k^{\mathrm{T}}\mathbf{p}^{(n)}(i)\right)\mathbf{u}_k, \qquad i = 1, \ldots, n_h, \tag{23}$$

and $g'(t) = \mathrm{d}g(t)/\mathrm{d}t$. Thus, $\mathbf{r}_k$ can be written in the following form:

$$\mathbf{r}_k = \mathbf{P}_1 \mathbf{r}_{1,k}, \tag{24}$$

where

$$\mathbf{P}_1 = \mathrm{diag}\left(\mathbf{1}_{n_h+1}^{\mathrm{T}}, \mathbf{p}_1^{(l)}\mathbf{1}_{n_r+1}^{\mathrm{T}}, \ldots, \mathbf{p}_{n_h}^{(l)}\mathbf{1}_{n_r+1}^{\mathrm{T}}\right), \tag{25}$$

and

$$\mathbf{r}_{1,k} = \Big[\mathbf{g}\left(\mathbf{P}^{(n)}\mathbf{u}_k\right)^{\mathrm{T}}, g'\left(\mathbf{u}_k^{\mathrm{T}}\mathbf{p}^{(n)}(1)\right)\mathbf{u}_k^{\mathrm{T}}, \ldots$$
$$\ldots, g'\left(\mathbf{u}_k^{\mathrm{T}}\mathbf{p}^{(n)}(n_h)\right)\mathbf{u}_k^{\mathrm{T}}\Big]^{\mathrm{T}}. \tag{26}$$

Equation (26) can be written in a more condensed form by using the Kronecker product:

$$\mathbf{r}_{1,k} = \left[\mathbf{g}\left(\mathbf{P}^{(n)}\mathbf{u}_k\right)^{\mathrm{T}}, \left(\mathbf{g}'\left(\mathbf{P}^{(n)}\mathbf{u}_k\right) \otimes \mathbf{u}_k\right)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{27}$$

where

$$\mathbf{g}'(t) = [\underbrace{g'(t), \ldots, g'(t)}_{n_h}]^{\mathrm{T}}. \tag{28}$$

The Fisher information matrix is now given by

$$\mathbf{P}^{-1} = \sum_{k=1}^{n_t} \mathbf{r}_k \mathbf{r}_k^{\mathrm{T}} = \mathbf{P}_1 \left[\sum_{k=1}^{n_t} \mathbf{r}_{1,k} \mathbf{r}_{1,k}^{\mathrm{T}} \right] \mathbf{P}_1. \tag{29}$$

Thus, the determinant of $\mathbf{P}^{-1}$ is given by

$$\begin{aligned} \det\left(\mathbf{P}^{-1}\right) &= \det\left(\mathbf{P}_1\right)^2 \det\left(\sum_{k=1}^{n_t} \mathbf{r}_{1,k} \mathbf{r}_{1,k}^{\mathrm{T}} \right) = \\ &= \prod_{i=1}^{n_h} \left(\mathbf{p}_i^{(l)}\right)^{2(n_r+1)} \det\left(\sum_{k=1}^{n_t} \mathbf{r}_{1,k} \mathbf{r}_{1,k}^{\mathrm{T}} \right). \end{aligned} \tag{30}$$

From (30) it is clear that the process of maximizing the determinant of $\mathbf{P}^{-1}$ with respect to $\mathbf{u}$ is independent of the linear parameters $\mathbf{p}^l$. This means that at least a rough estimate of $\mathbf{P}^{(n)}$ is required to solve the experimental design problem. Indeed, it is rather vain to expect that it is possible to obtain a design that is to be appropriate for all the networks of a given structure. It is very easy to imagine two neural networks of the same structure that may represent two completely different systems. If some rough estimates are given, i.e. they can be obtained with any training method for feed-forward neural networks [11], then specialized algorithms for D-optimum experimental design can be applied, e.g. the celebrated Wynn–Fedorov algorithm [1, 16].

3.2. Singularity of the Fisher information matrix

The Fisher information matrix $\mathbf{P}^{-1}$ of (1) may be singular for some parameter configurations, and in such cases it is impossible to obtain its inverse $\mathbf{P}$ necessary to calculate (5) as well as to utilize the specialized algorithms for obtaining the D-optimum experimental design [1, 16]. Fukumizu [8] established conditions under which $\mathbf{P}^{-1}$ is singular. These conditions can be formulated as follows:

Theorem 1. [8] *The Fisher information matrix $\mathbf{P}^{-1}$ of (1) is singular iff at least one of the following conditions holds true:*

1. *there exists j such that $[\mathbf{P}_{j,1}, \ldots, \mathbf{P}_{j,n_r}]^{\mathrm{T}} = \mathbf{0}$,*
2. *there exists j such that $\mathbf{p}_j^{(l)} = 0$,*
3. *there exist different j_1 and j_2 such that $\mathbf{p}^{(n)}(j_1) = \pm \mathbf{p}^{(n)}(j_2)$.*

A direct consequence of the above theorem is that a network with singular $\mathbf{P}^{-1}$ can be reduced to one with a positive definite $\mathbf{P}^{-1}$ by removing redundant hidden neurons. Based on this property it is possible to develop a procedure that can be used for removing the redundant neurons without performing the retraining of a network [9].

If the conditions of *Theorem 1* indicate that $\mathbf{P}^{-1}$ is not singular, then the strategy of collecting measurements according to the theory of D-optimum experimental design (the maximization of the determinant of $\mathbf{P}^{-1}$) guarantees that the Fisher information matrix is Positive definite. This permits

approximating an exact feasible parameter set (20) with an ellipsoid (cf. Fig. 3 to see the similarity to an ellipsoid), and hence it permits using (9) as an adaptive threshold.

Unfortunately, the conditions of *Theorem 1* have strictly theoretical meaning as in most practical situations the FIM would be close to singular but not singular in an exact sense. This makes the process of eliminating redundant hidden neurons far more difficult and there is no really efficient algorithm that can be employed to settle this problem. Indeed, the approach presented in [10] is merely sub-optimal. On the other hand, if such an algorithm does not give satisfactory results, i.e. the FIM is still close to the singular matrix, then FIM should be regularized in the following way [1, p. 110]:

$$\mathbf{P}_{\epsilon}^{-1} = \mathbf{P}^{-1} + \epsilon \mathbf{I}, \tag{31}$$

for $\epsilon > 0$ small but large enough to permit the inversion of $\mathbf{P}_{\epsilon}^{-1}$. The main drawback of the above approach is the fact that there is no clear rules for selecting ϵ. In this paper, we propose to use a completely different approach that is more appropriate for the models (1). As has already been mentioned, from (30) it is clear that the process of maximizing the determinant of $\mathbf{P}^{-1}$ with respect to $\mathbf{u}$ is independent of the linear parameters $\mathbf{p}^l$. This means that $\mathbf{p}^l$ can be set arbitrarily. This implies that it can be employed as an elegant tool for controlling the value of the determinant (30). Thus, before the iterative algorithm for finding D-optimum design is started (e.g. the Wynn–Fedorov algorithm [1, 16] used in this work), it is necessary to select $\mathbf{p}^l$ so as to ensure that the matrix $\mathbf{P}^{-1}$ is far from singular. Contrary to the solution presented in [1, p. 110], setting $\mathbf{p}^l$ is an easy task.

4. Experimental study with the DAMADICS benchmark

DAMADICS (*Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems*) was a Research Training Network (RTN) funded by the European Commission within the 5-th Framework Programme. This multidisciplinary and complementary RTN DAMADICS was focused on drawing together wide-ranging techniques and fault diagnosis within the framework of a real application to on-line diagnosis of a 5-stage evaporisation plant of a sugar factory in Lublin, Poland. The network focuses on the diagnosis of valve (cf. Fig. 4) plant actuators and looks towards real implementation methods for new actuator systems. The sugar factory is a subcontractor (under

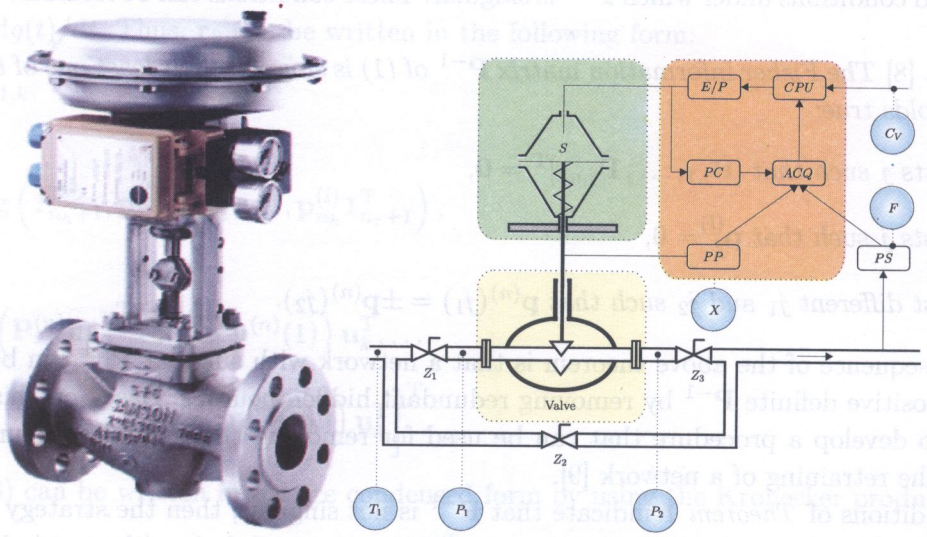

Fig. 4. Actuator and its scheme

the Technical University of Warsaw) providing real process data and the evaluation of trials of fault diagnosis methods.

The control valve is the mean used to prevent, allow and/or limit the flow of sugar juice through the control system (a detailed description of this actuator can be found in [5]. As can be seen in Fig. 4, the following process variables can be measured: CV is the control signal, $P1$ is the pressure at the inlet of the valve, $P2$ is the pressure at the outlet of the valve, $T1$ is the juice temperature at the inlet of the valve, X is the servomotor rod displacement, F is the juice flow at the outlet of the valve. In Fig. 4, additional three bypass valves (denoted by z_1, z_2, and z_3) can be seen. The state of these valves can be controlled manually by the operator. They are introduced for manual process operation, actuator maintenance and safety purposes. The data gathered from the real plant is available on the DAMADICS website [5]. Although a large amount of real data is available, these data do not cover all faulty situations. Thus, an actuator simulator was developed with MATLAB Simulink (available on [5]). The main reason for using the data from the simulator is that the achieved results can be easily compared with the results achieved with different approaches, e.g. [13, 14, 18].

The main objective of the subsequent part of this section is to develop a neural network that can be used for fault detection of an industrial valve actuator. The above task can be divided into the following steps: *Step 1:* training of a network based on the nominal data set. *Step 2:* design of the experiment based on the network obtained in *Step 1*. *Step 3:* training of a network based on the data obtained with the experimental design. Based on the experience with an industrial valve actuator, it was observed that the following subset of measured variables is sufficient for fault detection purposes: $\mathbf{u} = (CV, P1, 1)$, $y = F$.

In *Step 1* a number of experiments (the training of a neural network with the Levenberg–Marquardt algorithm [16]) were performed in order to find a suitable number of hidden neurons n_h (cf. (1)). As a result, a neural model consisting of $n_h = 12$ hidden was obtained. The main objective of *Step 2* was to utilize the above model and the Wynn–Fedorov algorithm [16] in order to obtain D-optimum experimental conditions. First, an initial experiment was generated in such a way so as to equally divide the design space of $\mathbf{u}$ (see [5] for details). Next, in order to prevent the fact that the FIM is almost a singular matrix it was appropriately transformed with the use of the linear parameters $\mathbf{p}_l$. Finally, the Wynn–Fedorov algorithm was applied. Figure 5 shows the support points ($n_e = 156$) and the variance function for the obtained D-optimum design. Moreover, Fig. 6 presents the support points and their respective weights. Based on the obtained continuous

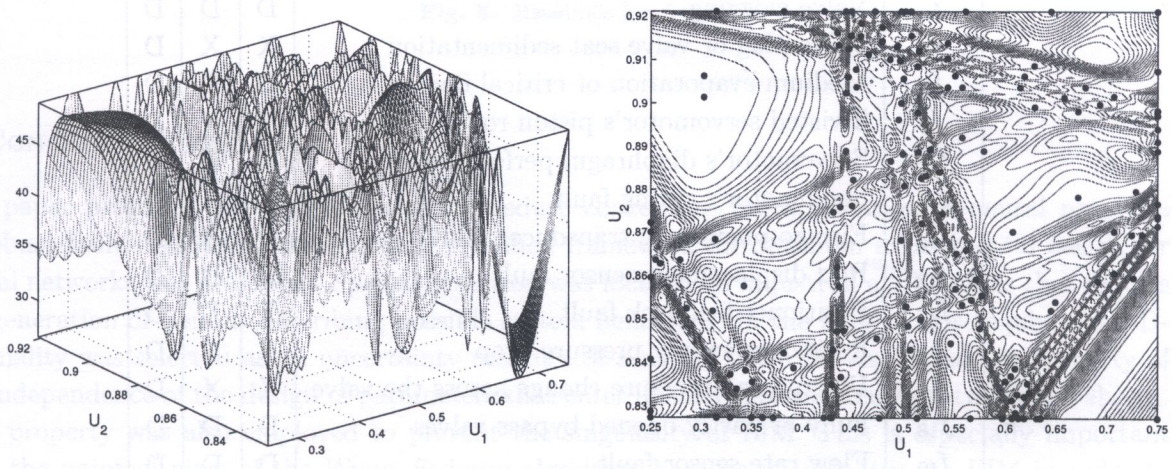

Fig. 5. Variance function and the corresponding support points

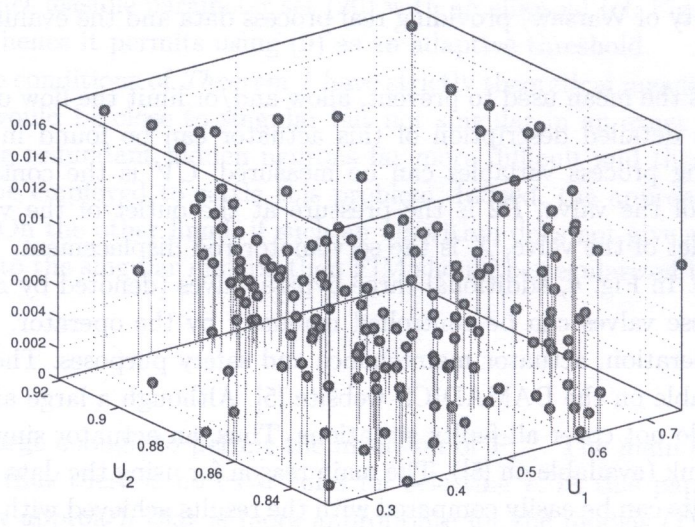

Fig. 6. Support points and their respective weights

design, a set consisting of $n_t = 500$ points was obtained and used for data generation. It should be strongly stressed that the data were collected in the steady-state of the valve because the utilized model (1) was static. Finally, the new data set was used for training the network. The objective of the subsequent part of this section is to use the obtained network for fault detection as well as to compare its performance with that of a network obtained for a nominal data set. Table 1 shows the results of fault detection for a set of faults being specified for the benchmark. Figures 7 and 8 show exemplary residuals and their respective bounds (9) obtained for the faults f_{11} and f_{13}, respectively. It can be observed that the neural network obtained with the use of D-optimum experimental design (Figs. 7a–8a) makes it possible to obtain more accurate bounds than those obtained with a neural network trained with a nominal data set (Figs. 7b–8b). This implies that the application of D-optimum design makes it possible to design a more reliable and sensitive fault detection system. As can be observed in Table 1, almost all faults specified for the benchmark can

Table 1. Results of fault detection (D – detected, N – not detected, X – not specified for the benchmark, S – small, M – medium, B – big)

Fault	Description	S	M	B
f_1	Valve clogging	D	D	D
f_2	Valve plug or valve seat sedimentation	X	X	D
f_7	Medium evaporation or critical flow	D	D	D
f_8	Twisted servomotor's piston rod	N	N	D
f_{10}	Servomotor's diaphragm perforation	D	D	D
f_{11}	Servomotor's spring fault	X	X	D
f_{12}	Electro-pneumatic transducer fault	D	D	D
f_{13}	Rod displacement sensor fault	D	D	D
f_{15}	Positioner feedback fault	X	X	D
f_{16}	Positioner supply pressure drop	N	N	D
f_{17}	Unexpected pressure change across the valve	X	X	D
f_{18}	Fully or partly opened bypass valves	D	D	D
f_{19}	Flow rate sensor fault	D	D	D

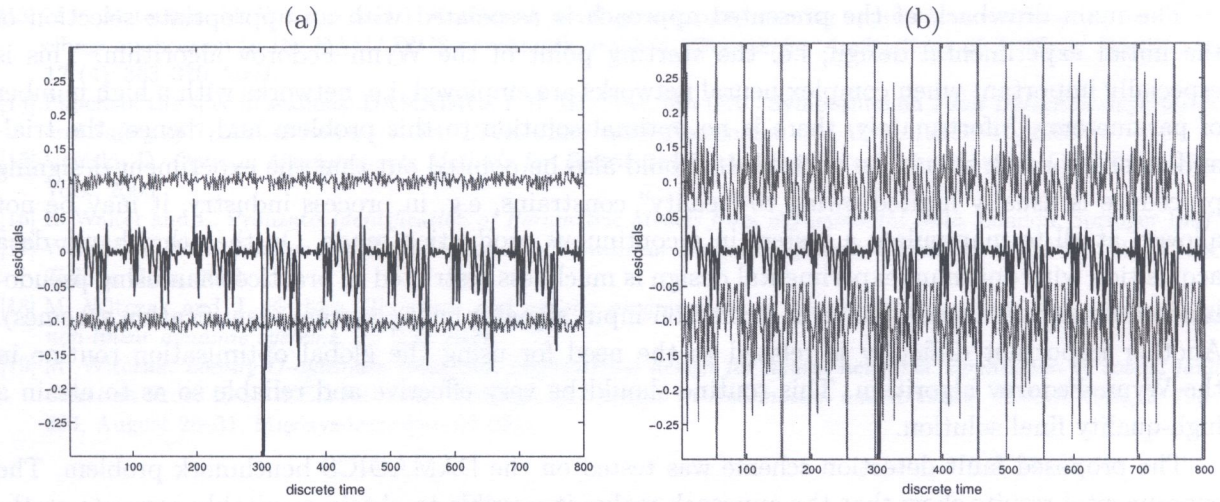

Fig. 7. Residuals for f_{11}

be detected. The main reason why the faults f_8 and f_{16} (small and medium) cannot be detected is that their effect is exactly at the same level as that of noise. However, it should be pointed out that this was the case for other techniques [14, 18] tested with the DAMADICS benchmark.

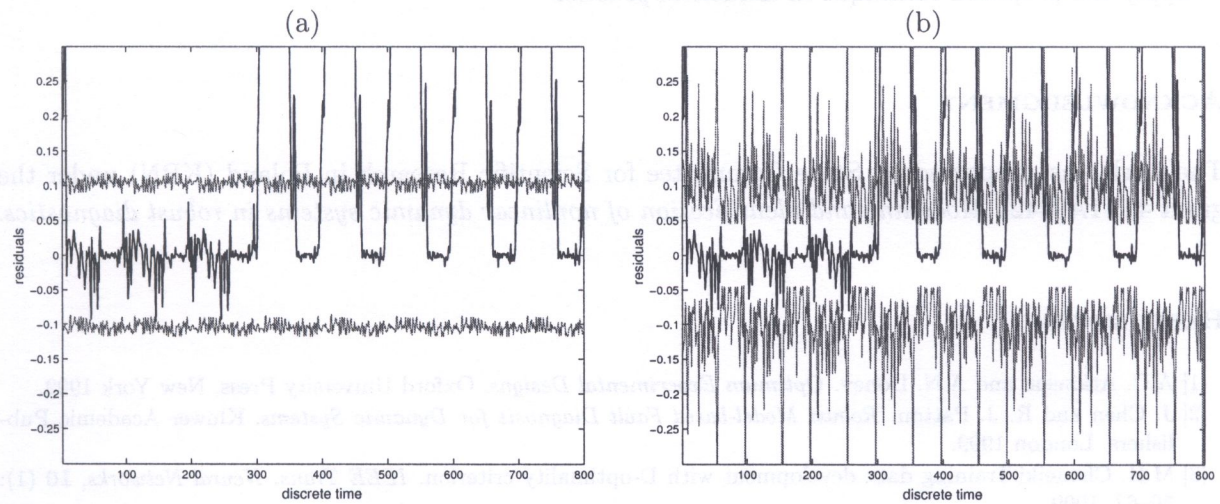

Fig. 8. Residuals for f_{13}

5. Concluding remarks

The paper proposes a complete design procedure concerning the application of neural networks to robust fault detection. In particular, a unified framework for optimum experimental design for neural networks was presented. Special attention was focused on industrial applications that require the generation of bounds describing possible system behaviour. In the light of this requirement, D-optimality was selected as an uncertainty measure of a neural network. An important property of the independence of the design of parameters that enter linearly into the neural network was shown. This property was also employed to prevent the singularity of FIM. This is especially important from the point of view of the Wynn–Fedorov algorithm because it evaluates the FIM in order to obtain the D-optimum design.

The main drawback of the presented approach is associated with an appropriate selection of the initial experimental design, i.e. the starting point of the Wynn–Fedorov algorithm. This is especially important when complex neural networks are employed, i.e. networks with a high number of parameters. Unfortunately, there is no optimal solution to this problem and, hence, the trial-and-error approach should be utilized. It should also be pointed out that the experiment designing procedure is usually limited owing to "reality" constrains, e.g. in process industry, it may be not allowed at all to manipulate a system in a continuous production mode. On the other hand, data acquisition with optimum experimental design is much less restricted in practice than using pseudo-random binary sequences (which is a common input signal in many system identification schemes). Another important difficulty is related to the need for using the global optimisation routine in the Wynn–Fedorov algorithm. This routine should be very effective and reliable so as to attain a high-quality final solution.

The proposed fault detection scheme was tested on the DAMADICS benchmark problem. The experimental results show that the approach makes it possible to obtain a suitably accurate mathematical description of the system with small modelling uncertainty. The developed model was employed to design a fault detection scheme for the valve actuator. In order to perform a comprehensive fault detection study, the MATLAB SIMULINK simulator of the valve actuator was employed. The obtained results show that almost all faults can be detected except for two small or medium ones. The authors hope that the results presented in this paper will encourage engineers to apply the proposed technique in industrial practice.

Acknowledgment

The work was supported by State Committee for Scientific Research in Poland (KBN) under the grant 4T11A01425 *Modelling and identification of nonlinear dynamic systems in robust diagnostics.*

References

[1] A.C. Atkinson and A.N. Donev. *Optimum Experimental Designs.* Oxford University Press. New York 1992.

[2] J. Chen and R. J. Patton. *Robust Model-based Fault Diagnosis for Dynamic Systems.* Kluwer Academic Publishers. London 1999.

[3] M.H. Choueiki.Training data development with D-optimality criterion. *IEEE Trans. Neural Networks*, **10** (1): 56–63, 1999.

[4] G. Chryssolouris, M. Lee and A. Ramsey. Confidence interval prediction for neural network models. *IEEE Trans. Neural Networks*, **7** (1): 229–232, 1996.

[5] DAMADICS (2004): Website of the Research Training Network *DAMADICS: Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems.* `http://diag.mchtr.pw.edu.pl/damadics/`.

[6] F. Delebecque, R. Nikoukah and H. Rubio Scola. Test signal design for failure detection: A linear programming approach. *Int. J. Appl. Math. Comput. Sci.*, **13** (4): 515–526, 2003.

[7] P.M. Frank, G. Schreier and E.A. Garcia. Nonlinear observers for fault detection and isolation. In: *New Directions in Nonlinear Observer Design* (Nijmeijer H., Fossen T.I., eds.) Berlin, Springer-Verlag 1999.

[8] K. Fukumizu. A regularity condition of the information matrix of a multi-layer perceptron network. *Neural Networks,* **9** (5): 871–879, 1996.

[9] K. Fukumizu. Active learning in multilayer perceptrons. In: D.S. Touretzky et al. eds.: *Advances in Neural Information Processing Systems*, Cambridge: MIT Press, pp. 295–301, 1996.

[10] K. Fukumizu. Statistical active learning in multilayer perceptrons. *IEEE Trans. Neural Networks,* **11** (1): 17–26, 2000.

[11] M.M. Gupta, L. Jin and N. Homma. *Static and Dynamic Neural Networks. From Fundamentals to Advanced Theory.* Wiley, New Jersey 2003.

[12] J. Korbicz, J.M. Kościelny, Z. Kowalczuk and W. Cholewa eds. *Fault Diagnosis. Models, Artificial Intelligence, Applications.* Berlin, Springer-Verlag 2004.

[13] M.F. Metenidis, M. Witczak and J. Korbicz. A novel genetic programming approach to nonlinear system modelling: application to the DAMADICS benchmark problem. *Engineering Applications of Artificial Intelligence*, **17** (4): 363–370, 2004.

[14] Papers of the special sessions: DAMADICS I, II, III. Proc. 5th IFAC Symposium on *Fault Detection Supervision and Safety of Technical Processes, SAFEPROCESS 2003*, Washington DC, USA, June 9-11, 2003.

[15] Uciński, D. *Optimal Measurements Methods for Distributed Parameter System Identification.* CRC Press New York, 2005.

[16] E. Walter and L. Pronzato *Identification of Parametric Models from Experimental Data.* London, Springer 1997.

[17] Witczak M. *Identification and Fault Detection of Non-linear Dynamic Systems.* University of Zielona Góra Press. Zielona Góra, 2003.

[18] M. Witczak and J. Korbicz *Observers and genetic programming in the identification and fault diagnosis of non-linear dynamic systems.* In: [12], 2004.

[19] M. Witczak. *Locally D-optimum sequential experimental design for neural networks: application to robust fault detection* Proc. 10-th IEEE Int. Conf. *Methods and Models in Automation and Robotics: MMAR 2004*, **2**: 798–804, August 28–31, Międzyzdroje Poland 2004.