JAISCR, 2011, Vol.1, No.2, pp. 103–114

OPTIMAL TRAINING STRATEGIES FOR LOCALLY RECURRENT NEURAL NETWORKS

Krzysztof Patan and Maciej Patan
Institute of Control and Computation Engineering,
University of Zielona Góra,
ul. Podgórna 50, 65-246 Zielona Góra, Poland
e-mail:{k.patan, m.patan}issi.uz.zgora.pl

Abstract

The problem of determining an optimal training schedule for locally recurrent neural network is discussed. Specifically, the proper choice of the most informative measurement data guaranteeing the reliable prediction of neural network response is considered. Based on a scalar measure of performance defined on the Fisher information matrix related to the network parameters, the problem was formulated in terms of optimal experimental design. Then, its solution can be readily achieved via adaptation of effective numerical algorithms based on the convex optimization theory. Finally, some illustrative experiments are provided to verify the presented approach.

1 Introduction

A training of neural network, being the dynamic data-driven process requires a proper selection of measurement data to provide satisfactory representation of the modelled system behaviour [7, 12]. In practice, this is equivalent to determination of a limited number of observational units obtained from the experimental environment in such a way as to obtain the best quality of the system responses.

The importance of input data selection has already been recognized in many application domains [28]. One of the most stimulating practical examples is Fault Detection and Identification (FDI) of industrial systems [11, 15]. A crucial issue among the fundamental tasks of failure protection systems is to provide reliable diagnosis of the expected system state. But to produce such a forecast, an accurate model is necessary and its calibration requires parameter estimation. Preparation of experimental conditions in order to gather informative measurements can be very expensive or even impossible (e.g. for the faulty system states). On the other hand, the data form real-world system may be very noisy and using all the available data may lead to significant systematic modelling errors. In result, we are faced with the problem of optimal choice of the available training data in order to obtain the most accurate model.

Although it is well known that the training quality for neural networks heavily depends on the choice of input sequences, surprisingly, there have been relatively few contributions to experimental design for those systems and, in addition, they focus mainly on the multi-layer perceptron class of networks [2, 5, 30, 9] or radial basis function networks [24, 4]. The applicability of such a static type of networks for the modelling of dynamic systems is rather limited. Recently, the problem of optimal selection of input sequences in the context of dynamic neural networks has been discussed by the authors in [16, 15], where the problem is formulated in spirit of optimum experimental design theory for lumped systems [6, 31, 3]. However, the simulation results presented therein concern the training of the single dynamic neuron only. The contribution of this work is to extend this approach to the locally recurrent neural network with one hidden layer which

can be applied in real-world systems. Moreover, to illustrate the delineated approach some experiments are performed using real process data.

2 Dynamic Neural Networks

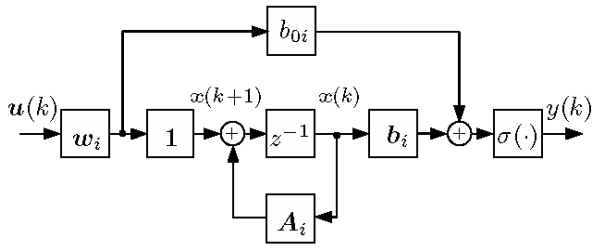

Figure 1. State-space form of the i-th neuron with IIR filter.

The topology of the neural network considered is analogous to that of the multi-layered feedforward one and the dynamics are reproduced by the so-called dynamic neuron models. Such neural networks are called locally recurrent globally feedforward [25, 13, 15]. Dynamic properties of the model are achieved by introducing an Infinite Impulse Response (IIR) filter into a neuron structure. As a consequence of incorporating an IIR filter between input weights and an activation function, the neuron can reproduce its own past inputs and activations using two signals: the input $u(k)$ and the output $y(k)$. The state-space representation of the neuron is shown in Figure 1. In this paper a discrete-time dynamic network with n time varying inputs and m outputs is discussed. The description of such kind of a dynamic network with v hidden dynamic neurons, each containing an r-th order IIR filter, is given by the following nonlinear system:

$$\begin{cases} x(k+1) = Ax(k) + Wu(k) \\ y(k) = C\sigma(Bx(k) + Du(k) - g)^T \end{cases}, \quad (1)$$

where $N = v \times r$ represents the number of model states, $x \in \mathbb{R}^N$ is the state vector, $u \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ are input and output vectors, respectively, $A \in \mathbb{R}^{N\times N}$ is the block diagonal state matrix ($\mathrm{diag}(A) = [A_1, \ldots, A_v]$), $W \in \mathbb{R}^{N\times n}$ ($W = [w_1 1^T, \ldots, w_v 1^T]^T$, where w_i is the input weight vector of the i-th hidden neuron), and $C \in \mathbb{R}^{m\times v}$ are the input and output matrices, respectively, $B \in \mathbb{R}^{v\times N}$ is a block diagonal matrix of feedforward filter parameters ($\mathrm{diag}(B) = [b_1, \ldots, b_v]$), $D \in \mathbb{R}^{v\times n}$ is the transfer matrix ($D = [b_{0_1} w_1^T, \ldots b_{0_v} w_v^T]^T$), $g = [g_1 \ldots g_v]^T$ denotes the vector of biases, and $\sigma : \mathbb{R}^v \to \mathbb{R}^v$ is the nonlinear vector-valued function. The presented structure can be viewed as a network with a single hidden layer containing v dynamic neurons as processing elements and an output layer with linear static elements. For structural details, the interested reader is referred to [15, 14, 17].

3 Optimal Sequence Selection Problem

3.1 Statistical Model

Let $y^j = y(u^j; \theta) = \{y(k;\theta)\}_{k=0}^{L_j}$ denote the sequence of network responses for the sequence of inputs $u^j = \{u(k)\}_{k=0}^{L_j}$ related to the consecutive time instants $k = 0, \ldots, L_j < \infty$ and selected from among an *a priori* given set of input sequences $\mathcal{U} = \{u^1, \ldots, u^P\}$. Here θ represents a p-dimensional unknown network parameter vector which must be estimated using observations of the system (i.e. filter parameters, weights, slope and bias coefficients).

From the statistical point of view, the sequences of observations related to P input sequences may be considered as

$$z^j(k) = y^j(k;\theta) + \varepsilon^j(k),\ k = 0, \ldots, L_j,\ j = 1, \ldots, P, \quad (2)$$

where $z^j(k)$ is the output and $\varepsilon^j(k)$ denotes the measurement noise. It is customary to assume that the measurement noise is zero-mean, Gaussian and white, i.e.

$$E[\varepsilon^i(k)\varepsilon^j(k')] = v^2 \delta_{ij}\delta_{kk'}, \quad (3)$$

where $v > 0$ is the standard deviation of the measurement noise, δ_{ij} and $\delta_{kk'}$ standing for the Kronecker delta functions.

An additional substantial assumption is that the training of the neural network, equivalent to the estimation of the unknown parameter vector θ, is performed via the minimization of the least-squares criterion

$$\hat{\theta} = \arg\min_{\theta\in\Theta_{\mathrm{ad}}} \sum_{j=1}^{P} \sum_{k=0}^{L_j} \|z^j(k) - y^j(k;\theta)\|^2, \quad (4)$$

where Θ_{ad} is the set of admissible parameters. It becomes clear that since $y^j(k;\theta)$ strongly depends on the input sequences u^j it is possible to improve the training process through appropriate selection of input sequences.

3.2 Sequence Quality Measure and Experimental Design

In order to properly choose the input sequences which will be most informative for the training of the dynamic network, a quantitative measure of the goodness of parameter identification is required. A reasonable approach is to choose a performance measure defined on the Fisher Information Matrix (FIM), which is commonly used in optimum experimental design theory [1, 3, 29, 27].

Sequences which guarantee the best accuracy of the least-squares estimates of θ are then found by choosing u^j, $j = 1,\ldots,P$ so as to minimize some scalar measure of performance Ψ defined on the *average Fisher information matrix* given by [19]:

$$M = \frac{1}{PL_j}\sum_{j=1}^{P}\sum_{k=0}^{L_j} H(u^j,k)H^T(u^j,k), \qquad (5)$$

where

$$H(u,k) = \left(\frac{\partial y(u,k;\theta)}{\partial\theta}\right)_{\theta=\theta^0} \qquad (6)$$

stands for the so-called *sensitivity matrix*, θ^0 being a prior estimate to the unknown parameter vector θ which can be obtained from previous experiments or alternatively some known nominal values can be used [26, 23, 19, 27].

Such a formulation is generally accepted in optimum experimental design for nonlinear dynamic systems, since the inverse of the FIM constitutes, up to a constant multiplier, the Cramér-Rao lower bound on the covariance matrix of any unbiased estimator of θ [29, 1], i.e.

$$\operatorname{cov}\hat{\theta} \succeq M^{-1}. \qquad (7)$$

Since the class of dynamic networks considered in this paper follows the universal approximation property [15], a proper representation of the system is only a matter of a network structure. Because a structure optimization is far beyond the scope of this work, we can assume in the following that the network has ability to properly represent the dynamics of the process considered and the parameter estimates are unbiased. If it is not a case we have to incorporate the bias terms into the (7), cf. [22].

Moreover, under somewhat mild assumptions [23, 27], it is legitimate to assume that our estimator is *efficient* in the sense that the parameter covariance matrix achieves the lower bound.

As for criterion Ψ, various choices are proposed in the literature [29, 3, 1], but the most popular choice is so-called D-optimality (determinant) criterion:

$$\Psi(M) = -\log\det M; \qquad (8)$$

which minimizes the volume of the uncertainty ellipsoid for the parameter estimates. The introduction of an optimality criterion renders it possible to formulate the sensor location problem as an optimization problem:

$$\Psi\left[M(u^1,\ldots,u^P)\right] \longrightarrow \min \qquad (9)$$

with respect to u^j, $j = 1,\ldots,P$ belonging to the admissible set $\mathcal{U}$.

The direct consequence of the assumption (3) is that we admit replicated input sequences, i.e. some u^js may appear several times in the optimal solution (because independent observations guarantee that every replication provides additional information). Consequently, it is sensible to reformulate the problem so as to operate only on the distinct sequences $u^1,\ldots,u^S$ instead of $u^1,\ldots,u^P$ by relabelling them suitably. To this end, we introduce $r_1,\ldots,r_S$ as the numbers of replicated measurements corresponding to the sequences $u^1,\ldots,u^S$. In this formulation, the u^is are said to be the *design* or *support* points, and $p_1,\ldots,p_S$ are called their weights. The collection of variables

$$\xi_P = \begin{Bmatrix} u^1, & u^2, & \ldots, & u^S \\ p_1, & p_2, & \ldots, & p_S \end{Bmatrix}, \qquad (10)$$

where $p_i = r_i/P$, $P = \sum_{i=1}^{S} r_i$, is called the *exact design* of the experiment. The proportion p_i of observations performed for u^i can be considered as the percentage of experimental effort spent at that sequence. Hence, we are able to rewrite the FIM in the form

$$M(\xi_P) = \sum_{i=1}^{S} p_i \frac{1}{L_i}\sum_{k=0}^{L_i} H^T(u^i,k)H(u^i,k). \qquad (11)$$

Here the p_is are rational numbers, since both r_is and P are integers. This leads to a discrete numerical analysis problem whose solution is difficult for standard optimization techniques, particularly when P is large. A potential remedy for this problem is to extend the definition of the design. This is achieved through the relaxation of constraints on weights, allowing the p_is to be considered as real

numbers in the interval $[0,1]$. This assumption will be also made in what follows. Obviously, we must have $\sum_{i=1}^{S} p_i = 1$, so we may think of the designs as probability distributions on $\mathcal{U}$. This leads to the so-called *continuous* designs, which constitute the basis of the modern theory of optimal experiments [3, 1]. It turns out that such an approach drastically simplifies the design, and the existing rounding techniques [3]justify such an extension. Thus, we shall operate on designs of the form

$$\xi = \left\{ \begin{matrix} u^1, & u^2, & \ldots, & u^S \\ p_1, & p_2, & \ldots, & p_S \end{matrix}; \quad \sum_{i=1}^{S} p_i = 1 \right\}, \quad (12)$$

which concentrates Pp_1 observational sequences for u^1 (so we repeat approximately Pp_1 times the presentation of this sequence during the training of the network), Pp_2 for u_2, and so on. Then we may redefine optimal design as a solution to the optimization problem

$$\xi^\star = \arg \min_{\xi \in \Xi(\mathcal{U})} \Psi[M(\xi)], \quad (13)$$

where $\Xi(\mathcal{U})$ denotes the set of all probability distributions on $\mathcal{U}$.

3.3 Characterization of Optimal Solutions

In the remainder of this chapter we shall assume that $H \in C(\mathcal{U};\mathbb{R}^p)$. The following characterizations of the optimal design $\xi^\star$ can be derived in a rather straightforward manner from the general results given in [19, 20] or [27].

Theorem 1 *An optimal design exists comprising no more than $p(p+1)/2$ support sequences. Moreover, the set of optimal designs is convex.*

The practical importance of this property cannot be underestimated since we can restrict our attention to the designs with limited number of sequences what significantly reduces the complexity of resulting optimization problem. But the next theorem is essential for the approach considered and provides a tool for checking the optimality of designs. It is usually called an *equivalence theorem* [10].

Theorem 2 *Equivalence theorem The following conditions are equivalent:*

(i) the design $\xi^\star$ minimizes $\Psi(M) = -\ln\det M(\xi)$,

(ii) the design $\xi^\star$ minimizes $\max_{u^i \in \mathcal{U}} \phi(u^i,\xi)$, and

(iii) $\max_{u^i \in \mathcal{U}} \phi(u^i,\xi) = p$,

and the so-called *sensitivity function*

$$\phi(u^i,\xi) = \text{trace}\left(\frac{1}{L_i} \sum_{k=0}^{L_i} H^T(u^i,k) M^{-1} H(u^i,k) \right)$$

is of paramount importance here as it can be interpreted in terms of average variance of the estimated system response being the natural measure for the quality of the training process. From the result above it comes immediately that suppressing the maximal level of the prediction variance is equivalent to the optimization of the D-optimality criterion. This paves the way to almost direct application of numerous efficient algorithms known from experimental design theory to the discussed problem. Since analytical determination of optimal designs is difficult or impossible even for very simple network structures, some iterative design procedures will be required. A dedicated computational scheme for that purpose is given in the next section.

4 Selection of Training Sequences

It is clear that design problem (13) is highly nontrivial due to its complexity. Moreover, some additional design factors as sequences lengths, sampling time or initial conditions also can influence the quality of training and may be a part of optimization process. In order to somewhat simplify the problem and reduce its complexity, in what follows, we assume that the set of admissible sequences $\mathcal{U}$ is finite.

In such a framework, a particularly simple and efficient computational algorithm can be derived based on the mapping $\mathcal{T} : \Xi(\mathcal{U}) \to \Xi(\mathcal{U})$ defined by

$$\mathcal{T}\xi = \left\{ \begin{matrix} u^1, & \ldots, & u^S \\ p_1\phi(u^1,\xi)/p, & \ldots, & p_S\phi(u^S,\xi)/p \end{matrix} \right\}. \quad (14)$$

From Theorem 2 it follows that a design $\xi^\star$ is D-optimal if it is a fixed point of the mapping $\mathcal{T}$, i.e.

$$\mathcal{T}\xi^\star = \xi^\star. \quad (15)$$

Therefore, the following algorithm can be used as a generalization of that proposed in [21, p.139] for the classical optimum experimental design problem

consisting in iterative computation of a D-optimum design on a finite set:

Step 1. Guess a discrete starting design $\xi^{(0)}$ such that $p_i^{(0)} > 0$ for $i = 1,\ldots,S$. Choose some positive tolerance $\eta \ll 1$. Set $\ell = 0$.

Step 2. If the condition

$$\frac{\phi(u^i,\xi^{(\ell)})}{p} < 1+\eta, \quad i = 1,\ldots,S$$

is satisfied, then *STOP*.

Step 3. Construct the next design $\xi^{(k+1)}$ by determining its weights according to the rule

$$p_i^{(\ell+1)} = p_i^{(\ell)} \frac{\phi(u^i,\xi^{(\ell)})}{m}, \quad i = 1,\ldots,S,$$

increment k by one and go to Step 2.

Its important to include all admissible input sequences as a support The convergence result of this scheme can be found in [27].

5 Examples

5.1 DC motor

Simulation setting.

Experiments were carried out using the AMIRA DR300 laboratory system. This laboratory system is used to control the rotational speed of a DC motor with a changing load. This is the single input single output system.

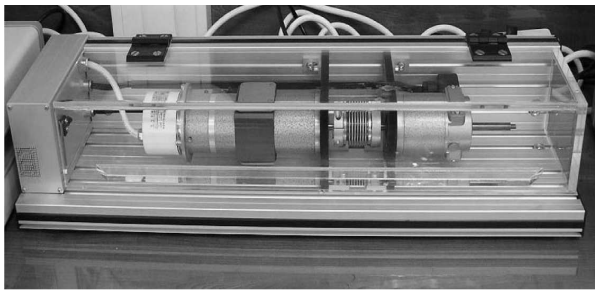

Figure 2. Amira DR300 laboratory stand.

A separately excited DC motor was modelled by using the dynamic neural network presented briefly in Section 2. The output signal was the rotational speed (T) measured by an analog tachometer. The input signal (U) was selected as a sum of sinusoids:

$$\begin{aligned} U(k) =& 3\sin(2\pi 1.7k) + 3\sin(2\pi 1.1k - \pi/7) \\ &+ 3\sin(2\pi 0.3k + \pi/3) \end{aligned} \tag{16}$$

The structure of the neural network model (1) was selected arbitrarily and had the following structure: one input, three IIR neurons with second order filters and hyperbolic tangent activation functions, and one linear output neuron. Taking into account that a neural network is a redundant system, some of its parameters are not identifiable. In order to apply optimum experimental design to the neuron training, certain assumptions should be made. So, without loss of generality, let us assume that the feedforward filter parameter b_0 for each hidden neuron is fixed to the value of 1. This reduces the dimensionality of estimation and assures the identifiability of the rest of the parameters (i.e. it assures that the related FIM is non-singular).

At the beginning, the network was preliminarily trained in order to obtain the initial parameters estimates. Feeding the laboratory system with signal (16), a learning set containing 500 samples was formed, and then the training process was carried out off-line for 2000 steps using the Extended Dynamic Back-Propagation (EDBP) algorithm [16]. At the second stage of the training process the learning data were split into 20 time sequences, containing 150 consecutive samples each. The design purpose was to choose from this set of all learning patterns the most informative sequences (in the sense of D-optimality) and their presentation frequency (i.e. how often they should be repeated during the training). To determine the optimal design, a numerical routine from Section 4 was implemented in the form of the MATLAB program. All the admissible learning sequences taken with equal weights formed the initial design. The accuracy of the design algorithm was set to $\eta = 10^{-2}$.

Results

The network was preliminarily trained and the initial network parameters estimates are presented in the second column of Table 1. In this case Sum of Squared Errors (SSE) calculated using the training set was equal to 5.7001. After that, the training of the network was continued in two ways. The first way was to use the optimal training sets selected during the optimum experimental design phase. The second way was to use random sequences as the training ones. The purpose of these experiments is to check the quality of parameter estimation. In the case considered here the opti-

Table 1. Sample mean and the standard deviation of parameter estimates.

para-	initial	sample mean		standard deviation	
meter	value	random design	optimal design	random design	optimal design
w_1	0.3232	0.2867	0.2894	0.0104	0.0028
w_2	0.9000	0.9105	0.9082	0.0034	0.0009
w_3	0.0758	0.0898	0.0789	0.0194	0.0027
b_{11}	0.8328	0.8187	0.8195	0.0040	0.0011
b_{21}	-0.6316	-0.6053	-0.6072	0.0078	0.0019
b_{31}	0.8558	0.8616	0.8581	0.0079	0.0011
b_{12}	0.7892	0.7742	0.7747	0.0042	0.0011
b_{22}	0.0631	0.0910	0.0897	0.0082	0.0019
b_{32}	0.5745	0.5808	0.5812	0.0076	0.0011
a_{11}	0.1258	0.1302	0.1301	0.0012	0.0003
a_{21}	0.0853	0.0807	0.0812	0.0015	0.0004
a_{31}	-0.4171	-0.4196	-0.4170	0.0055	0.0015
a_{12}	0.1656	0.1703	0.1703	0.0012	0.0003
a_{22}	0.0266	0.0217	0.0221	0.0016	0.0004
a_{32}	-0.5566	-0.5587	-0.5562	0.0052	0.0015
g_1	-0.3794	-0.4057	-0.4024	0.0132	0.0055
g_2	-0.3978	-0.3599	-0.3673	0.0206	0.0089
g_3	0.3187	0.3040	0.3136	0.0189	0.0008
c_1	-0.4908	-0.4905	-0.4893	0.0081	0.0032
c_2	0.7773	0.7708	0.7716	0.0078	0.0035
c_3	0.4540	0.4438	0.4408	0.0075	0.0006

mal design consists of the sequences $5, 6, 8$ and 16. For a selected design, each distinct sequence was replicated proportionally to its weight in the design with total number of replications assumed to be $P = 10$. For example, if the optimal design consists of the four aforementioned sequences with the weights 0.3, 0.1, 0.3 and 0.3, respectively, then during the training the 5-th, 8-th and 16-th sequences were used three times each, and the 6-th sequence only once (in random order). The training procedure was repeated 10 times using different measurement noise affecting the output of the system. The statistics are presented in Table 1. As we can see, the application of training sets selected according to the optimal design leads to the better accuracies of parameter estimates than in the case of randomly selected training sets. It is observed that the standard deviation of each network parameter has lower value in the case of the optimal design what means the more reliable estimate of a given parameter.

The uncertainty of the network response prediction is examined based on the parameter estimates determined using the optimal and random designs. The testing phase of each of 10 realizations of locally recurrent network was performed using 1000 samples different from the training ones. The results of testing are presented in Table 2.

Table 2. Results of network testing.

Network realization	Random design	Optimal design
1	29.7367	31.0998
2	27.4287	26.5564
3	42.4463	26.4758
4	85.8052	26.6182
5	99.5833	26.4214
6	82.9475	26.4577
7	35.3615	26.6521
8	29.6130	26.5550
9	26.8438	26.5030
10	26.2403	26.2885

Looking at these results one can state that using random design it is possible to obtain a good generalization of the network, e.g. networks 9 and 10, but the results of training are not repetitive as in the

case of optimal design when 9 of 10 training run give the similar good results. This fact, in connection with the plot of response prediction variance (Figure 2) clearly shows that training based on optimal learning sequences leads to greater reliability of the network response as the maximal variance level can be significantly reduced.

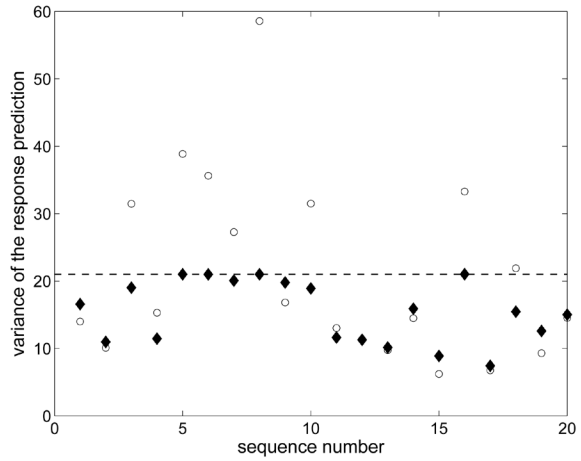

Figure 3. Variances of the model response prediction for the optimum design (diamonds) and random design (circles)

5.2 Tunnel Furnace

Simulation setting.

As an experimental testbed for the second example a laboratory model of tunnel furnace has been used (Figure 4).

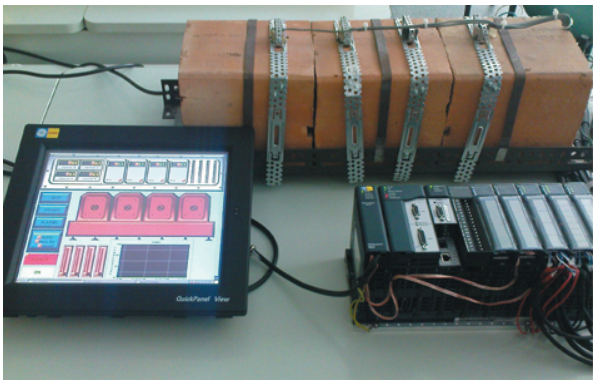

Figure 4. A laboratory system of tunnel furnace with the RX3i controllers and operational panel

This is the multi input multi output system. It contains four electric heaters which are controlled with the continuous input signals and four resistance detectors (RTD) measuring temperature gradient along the furnace chamber. The control system is based on the industrial programmable logic controllers PACSYSTEMS RX3i produced by GE FANUC Intelligent Platforms and supplemented with touchpad operational panel QUICKPANEL CE working under Microsoft Windows CE .NET 5. As the input signals the random step functions were selected in order to provide the persistent excitation of the object.

The system to be modelled has three inputs (the fourth input is reserved for the diagnostic purposes) and four outputs. In order to model the tunnel furnace the MIMO representation was decomposed into four MISO models. The structure of each neural network model (1) was selected arbitrarily and had the following structure: three inputs, three IIR neurons with second order filters and hyperbolic tangent activation functions, and one linear output neuron. Once again, let us assume that the feedforward filter parameter b_0 for each hidden neuron is fixed to the value of 1. Firstly, each network is trained in a classical way. Training set contains 500 samples and the training process was carried out off-line for 100 steps using the Levenberg-Marquardt (LM) algorithm. The LM training can be easily implemented based on the EDBP algorithm (see [8]). At the second stage of the training process the learning data were split into 30 time sequences, containing 100 consecutive samples each. The design purpose was to choose from this set of all learning patterns the most informative sequences (in the sense of D-optimality) and their presentation frequency.

Results

The modelling quality in the form of SSE calculated for each initially trained neural model using the testing set containing 3000 samples are presented in the first column of Table 3. It is obvious that modelling results are not satisfactory with exception of the neural network modelling 4th output of the system. In order to achieve more reliable models, the training is continued in two ways. The first way is to use the optimal training sets selected during the optimum experimental design phase. The second way is to use random sequences as the training ones (a method in-between cross-validation and bootstrap techniques). The purpose of these experiments is to check how different methods of input data presentation influence on the qual-

ity of the models. The optimally selected training sequences with weights assigned to them are presented in Figure 5. In each case considered, the design algorithm selected 9 or 10 sequences out of 30 as the most informative. Analyzing the distribution of the sequences one can say that models for each system output require different set of training sequences although some sequences are common for all models (16th, 22th, 28th, 30th), however, have different impact on training taking into account presentation frequency. In the case considered here, for a selected design, each distinct sequence is replicated proportionally to its weight in the design with total number of replications assumed to be $P = 20$. The training procedure was repeated 10 times and the modelling quality, in the form of SSE calculated using 3000 testing samples, for the best achieved models are presented in the second column of Table 3. Outputs of the tunnel furnace and coresponding model outputs are shown in Figure 6. As one can see there neural models mimic the behaviour of the system pretty well, however some problems are observed in the case of the second model (the worst modelling quality). Summarizing, once again the application of training sets selected according to the optimal design leads to the better network generalization than in the case of classical training.

Table 3. Quality of neural models

Model	Initial model	Optimal design	Random design
output 1	34.44	1.93	2.83
output 2	11.79	2.6	3.69
output 3	34.78	0.8658	5.96
output 4	0.37	0.064	0.093

Results achieved using the optimal experimental design are compared with the random designs. In this case the training sequences were selected randomly with total number of presentations equal to $P = 20$. The training procedure was repeated 10 times and the modelling quality, in the form of SSE calculated using 3000 testing samples, for the best achieved models are presented in the third column of Table 3. Using random designs, a better modelling quality was obtained than using classical training. It is caused by the fact that such a way of training sequences presentation is something in-between the cross-validation and bootstrap techniques. However, comparing results achieved using random designs with those achieved using optimal designs one can see that the better results are obtained using the latter method, especially in the case of the third output of the system.

6 Conclusions

We have addressed the problem of selecting optimal training sequences in view of accurate modelling of responses for locally recurrent neural networks. Although the problem of qualitative choice of learning data is well recognized and has been approached from various angles since the mid-1990s, there are still few systematic and versatile methods for its solution. In this work we started from the formulation, in which the training data comprises finite number of time sequences and the aim is to select the frequencies of their presentations as to maximize the quality of the training process. The problem formulated in terms of the minimization of the variance of the network response prediction can be translated to maximizing the determinant of the Fisher information matrix associated with the estimated network parameters.

Obtained results show that some well-known methods of optimum experimental design theory for linear regression models can be easily extended to the setting of the optimal training sequence selection problem for dynamic neural networks. The clear advantage of the proposed approach is that the quality of the training process measured in terms of the uncertainty of network response prediction can be significantly improved with the same effort spent on training or, alternatively, training process complexity can be reduced without degrading network performance.

The proposed approach was also tested using other network structures. Experiments were carried out for a locally recurrent network with two hidden neurons as well as for a network with five hidden neurons. In each case considered, the results are similar taking into account reliability of the parameters estimates.

It is important to express that delineated approach is dedicated to situation where it is possible to replicate some process trials (e.g., as for the quality tests or system identification experiments) for the same input data or we have redundant sensors

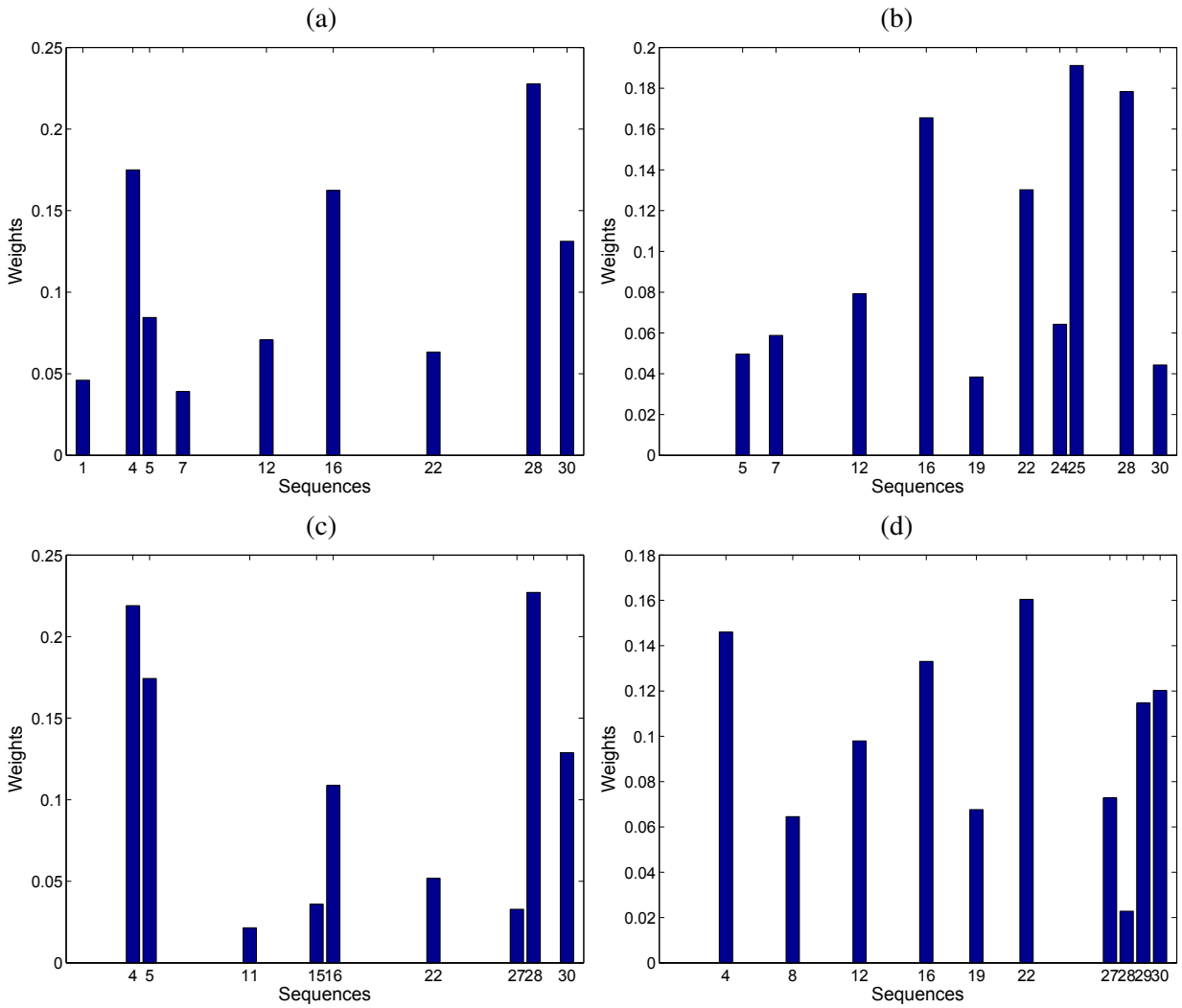

Figure 5. Optimal sequences weights histograms: the first output (a), second output (b), third output (c), fourth output (d)

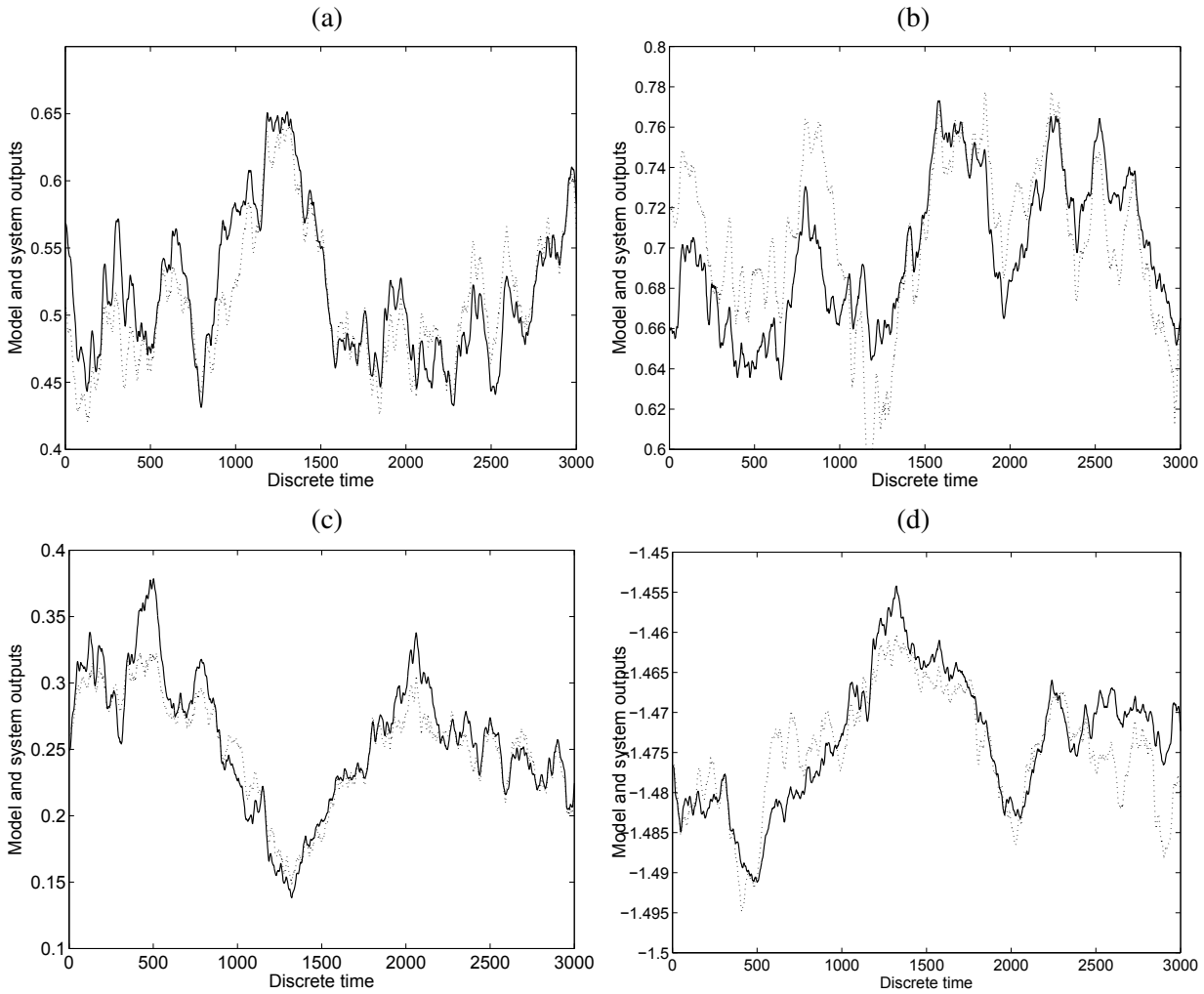

Figure 6. Responses of the tunnel furnace (solid) and model (dotted): the first output (a), second output (b), third output (c), fourth output (d)

providing additional observations. From practical point of view, often it may be very difficult to provide such experimental conditions. In such a case we have to revert to other algorithmic techniques to imply some additional constraints on the experimental design. For example an application of exchange type algorithms for determination of informative training sequences based on the concept of directly constrained design measures provides very promising results [18]. In such an approach the frequencies of presentation are fixed what significantly simplifies the implementation of training and broaden the practical application area.

Obviously, there is still necessity for further refinements and theoretical developments. We can mention the following points which are the matter of our current research:

- Sequential design techniques will be investigated in order to provide more robust design of training sequences with respect to network parameter misspecification. This should pave the way towards on-line training schemes.
- Based on the optimal control techniques it is possible to state the problem in terms of a proper dynamic design of input time sequences which provide the best conditions of training process. This, however, is a very challenging problem as it quickly leads to non-convex nonlinear optimization tasks with nonlinear constraints.

Acknowledgments

The first Author was supported in part by the Ministry of Science and Higher Education in Poland under the grant N N514 2305 40. The second Author was supported by the Ministry of Science and Higher Education in Poland under the grant N N514 2305 37.

References

[1] A. C. Atkinson, A. N. Donev and R. Tobias, *Optimum Experimental Design, with SAS*, Oxford University Press, Oxford, 2007.

[2] M. H. Choueiki and C. A. Mount-Campbell, *Training data development with the doptimality criterion,* IEEE Transactions on Neural Networks, 10(1):56–63, 1999.

[3] V. V. Fedorov and P. Hackl, *Model-Oriented Design of Experiments, Lecture Notes in Statistics*, Springer-Verlag, New York, 1997.

[4] E. Fokoué and P. Goel, *An optimal experimental design perspective on radial basis function regression,* Communications in Statistics - Theory and Methods, 40(7):1184–1195, 2011.

[5] K. Fukumizu, *Statistical active learning in multilayer perceptrons,* IEEE Transactions on Neural Networks, 11:17–26, 2000.

[6] G. C. Goodwin and R. L. Payne, *Dynamic system identification, Experiment design and data analysis, Mathematics in Science and Engineering,* Academic Press, New York, 1977.

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

[8] M. T. Hagan and M. B. Menhaj, *Training feedforward networks with the Marquardt algorithm,* IEEE Transactions on Neural Networks, 5:989–993, 1994.

[9] S. Issanchou and J. P. Gauchi, *Computer-aided optimal designs for improving neural networks generalization,* Neural Networks, 21:945–950, 2008.

[10] J. Kiefer and J. Wolfowitz, *Optimum designs in regression problems,* The Annals of Mathematical Statistics, 30:271–294, 1959.

[11] J. Korbicz and J.M. Kościelny, editors. *Modeling, Diagnosis and Process control, Implementation in the Diaster System.* Springer-Verlag, Berlin Heidelberg, 2010.

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

[13] T. Marcu, L. Mirea, and P. M. Frank, *Development of dynamical neural networks with application to observer based fault detection and isolation,* International Journal of Applied Mathematics and Computer Science, 9(3):547–570, 1999.

[14] K. Patan, *Stability analysis and the stabilization of a class of discrete-time dynamic neural networks,* IEEE Transactions on Neural Networks, 18:660–673, 2007.

[15] K. Patan, *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes. Lecture Notes in Control and Information Sciences,* SpringerVerlag, Berlin, 2008.

[16] K. Patan and M. Patan, *Selection of training sequences for locally recurrent neural network training,* In K. Malinowski and L. Rutkowski, editors, Recent Advances in Control and Automation, pages 252–262, Academic Publishing House, EXIT, Warsaw, 2008.

[17] K. Patan and M. Patan, *Corrigendum to stability analysis and the stabilization of a class of discrete-time dynamic neural networks,* IEEE Transactions on Neural Networks, 20:547–548, 2009.

[18] K. Patan and M. Patan, *Selection of training data for locally recurrent neural network,* In Proc. 20th Int. Conference on Artificial Neural Networks, ICANN 2010, Thesaloniki, Greece, 2010, Published on CD-ROM.

[19] M. Patan, *Optimal Observation Strategies for Parameter Estimation of Distributed Systems,* volume 5 of Lecture Notes in Control and Computer Science, Zielona Góra University Press, Zielona Góra, Poland, 2004.

[20] M. Patan and K. Patan, *Optimal observation strategies for model-based fault detection in distributed systems,* International Journal of Control, 78(18):1497–1510, 2005.

[21] A. Pázman, *Foundations of Optimum Experimental Design, Mathematics and Its Applications,* D. Reidel Publishing Company, Dordrecht, 1986.

[22] A. Pázman, *Nonlinear Statistical Models*, Kluwer, Dordrecht, 1993.

[23] E. Rafajowicz, *Optimum choice of moving sensor trajectories for distributed parameter system identification,* International Journal of Control, 43(5):1441–1451, 1986.

[24] K. Sung and P. Niyogi, *Active learning the weights of a RBF network,* In Proc. IEEE Workshop on Neural Networks for Signal Processing, Cambridge, MA, USA, 40–47, 1995.

[25] A. Ch. Tsoi and A. D. Back, L*ocally recurrent globally feedforward networks: A critical review of architectures,* IEEE Transactions on Neural Networks, 5:229–239, 1994.

[26] D. Ucinski, O*ptimal selection of measurement locations for parameter estimation in distributed processes,* International Journal of Applied Mathematics and Computer Science, 10(2):357–379, 2000.

[27] D. Uciński, *Optimal Measurement Methods for Distributed Parameter System Identification,* CRC Press, Boca Raton, 2005.

[28] M. van de Wal and B. de Jager, *A review of methods for input/output selection,* Automatica, 37:487–510, 2001.

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

[30] M. Witczak, *Toward the training of feedforward neural networks with the D-optimum input sequence,* IEEE Transactions on Neural Networks, 17:357–373, 2006.

[31] M. B. Zarrop and G. C. Goodwin, *Comments on optimal inputs for system identification,* IEEE Transactions on Automatic Control, 20(2):299–300, 1975.