

# TNN A180 Rev.: Learning Neural Networks With Noisy Inputs Using The Errors-In-Variables Approach

Jürgen Van Gorp<sup>\*</sup>, Johan Schoukens<sup>\*\*</sup> and Rik Pintelon<sup>\*\*\*</sup>

**Abstract** — Currently, most learning algorithms for neural network modeling are based on the Output Error approach, using a least squares cost function. This method provides good results when the network is trained with noisy output data and known inputs. Special care must be taken, however, when training the network with noisy input data, or when both inputs and outputs contain noise. This paper proposes a novel cost function for learning NN with noisy inputs, based on the Errors-In-Variables stochastic framework. A learning scheme is presented and examples are given demonstrating the improved performance in Neural Network curve fitting, at the cost of increased computation time.

**Keywords** — Neural Network Identification, Errors-In-Variables, Nonlinear identification

## I. INTRODUCTION

The goal of system identification is to estimate the parameters of a given model from a set of measured data. In this context linear identification and nonlinear identification of systems such as Neural Networks (NN), Fuzzy Logic (FL), etc. can all be seen as different models in the same general identification frame. In general, the identification is carried out as follows (also see Fig. 1):

- 1) Perform input-output measurements  $(u_k, y_k)$ . In this paper we will consider noisy observations of the true, unknown values  $(u_{0,k}, y_{0,k})$ . A basic decision is the choice of persistent measurements that fully cover the wanted model behaviour.
- 2) Choose the model. This can be either a linear or a nonlinear model, based on specific properties associated with the model. Fuzzy models are well known to map local linearities, while NN are optimal for soft nonlinear mappings. Another basic decision is the choice of either white or black box models.
- 3) Choose a cost function. Although this is usually neglected, choosing a wrong cost function can degrade the final plant model completely. Mostly the Least Squares (LS) cost function is used. If information on

the measurement variances is known, the Weighted Least Squares (Bayesian) cost function is applicable. Regularization [15] issues are implemented by adding extra terms in the cost function.

These three steps determine the basic properties of the model which are obtained by minimizing the cost function. This minimization is usually a nonlinear problem that is solved numerically. This leads to a series of fundamental choices, resulting in different learning schemes:

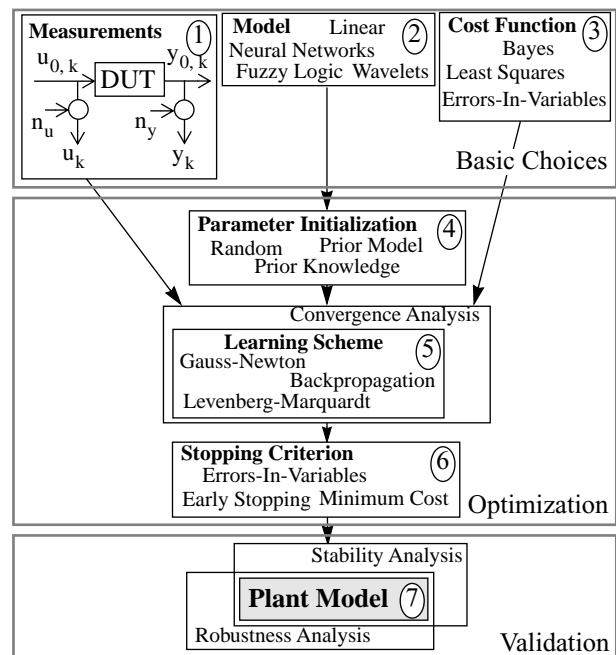


Fig. 1: A standard scheme for modeling

- 4) Initialize the model parameters. Fuzzy Logic involves much effort in the choice of optimal fuzzy sets. Most optimization methods start from random initial model parameters. In this paper the results of a prior modeling with the OE cost function are used as initial parameters for the EIV cost function.
- 5) The optimization of the model parameters can be achieved via various methods, starting from simple gradient methods to Genetic Algorithms. The selection of the different learning schemes is based on the speed of optimization and the possibility of avoiding local

The authors are with the Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium, E-mail: \*Jurgen.Van.Gorp@vub.ac.be \*\*Johan.Schoukens@vub.ac.be \*\*\*Rik.Pintelon@vub.ac.be

minima.

- 6) When does the optimization stop ? This is not always at the minimum of the cost function. When the data are sparse or the measurements are noisy, early stopping [16] is recommended.
- 7) Eventually, the identified model can be analysed for robustness and stability. Note that stability can be imposed during the optimization part of the modeling scheme (constrained optimization). This is usually done by adding extra terms in the cost function [4].

To date, much has been done to introduce different models and different learning schemes in the NN domain. However, more work is required on choosing the proper cost function which in turn determines completely the stochastic properties (noise sensitivity) of the NN. In most cases the LS cost function is used. In this paper it will be shown that using LS with noisy input data causes biasing in the NN parameters. An attempt to reduce this bias is made by replacing the output error cost function by an Errors-In-Variables (EIV) cost, which deals with the input noise.

The contribution of this paper to nonlinear modeling is shown in detail in Fig. 2: The Errors-In-Variables cost function is introduced to NN modeling and two learning schemes are implemented for the minimization of the cost function. Noise on the outputs has already been considered within the theory of Bayesian inference, introduced to NN by Green and MacKay [8] and Bishop [2]. However, using Bayesian techniques on both inputs and outputs leads to a novel estimator, called the EIV estimator. The EIV cost function that is presented, is already in use for the identification of linear [3] [14] [19] and nonlinear [22] models. When applied to NN, the inherent black box structure of NN will cause the EIV technique to be prone to overfitting. To overcome this, we introduce an early stopping criterion, based on the EIV cost function and Bayesian techniques.

This paper is organized as follows. Section II begins with a description of a measurement setup with noisy input and output measurements. Basic assumptions are made on the noise and notations are introduced. Section III describes how the generally used Least Squares cost function is usable in the presence of output noise, but leads to biased NN parameters when a NN is trained with noisy input samples. The EIV cost function is introduced in Section IV, and it is proven that this new cost function is less prone to bias effects when training with noisy inputs. Section IV also describes how a small number of repeated measurements are sufficient for obtaining the sample variances needed for the EIV cost function. The choice of the cost function directly influences the optimization of the model parameters. To resolve this, Section V describes the training of the NN when using the EIV cost function. More specifically, the learning steps are

derived for training a NN with the backpropagation and the Levenberg-Marquardt learning schemes. The section also describes an early stopping criterion that can be used with both learning schemes, and gives a detailed modeling scheme. Section VI gives an implementation of the scheme for different practical examples and compares the EIV cost function with the OE cost function. Section VII shows that the OE cost function is actually a special case of the EIV cost function and describes the use of the EIV cost function for noiseless outputs.

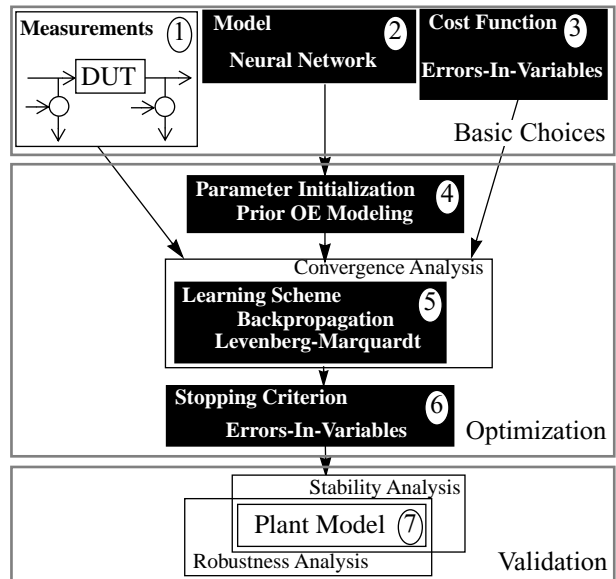


Fig. 2: Contributions to nonlinear modeling that are brought in this paper

## II. PROBLEM STATEMENT

Consider a Single Input Single Output (SISO) linear or nonlinear system  $(\mathcal{L})$ , as shown in Fig. 3. For this system we seek a Neural Network (NN) black box model  $y_k^{[i]} = f_{NN}(\theta, u_k^{[i]})$ , with  $\theta$  the neural network parameters,  $u_k^{[i]}$  the input and  $y_k^{[i]}$  the output of the system.

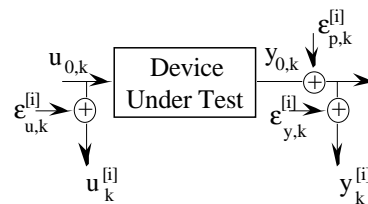


Fig. 3: Measurement setup

The NN is trained with a set of  $N \times M$  input-output samples, in which each measurement  $k$ ,  $k = 1 \dots N$  is repeated  $M$  times. The input and output matrices are

1. The theory also applies for MIMO systems. Here only SISO systems are regarded for simplicity.

denoted by  $\mathbf{u}$  and  $\mathbf{y}$ , and defined as

$$\mathbf{u} \in \mathbb{R}^{N \times M}; \mathbf{u} = \begin{bmatrix} \mathbf{u}_1^{[1]} & \mathbf{u}_1^{[2]} & \dots & \mathbf{u}_1^{[M]} \\ \mathbf{u}_2^{[1]} & \mathbf{u}_2^{[2]} & \dots & \mathbf{u}_2^{[M]} \\ \dots & \dots & \dots & \dots \\ \mathbf{u}_N^{[1]} & \mathbf{u}_N^{[2]} & \dots & \mathbf{u}_N^{[M]} \end{bmatrix} \quad (1)$$

and

$$\mathbf{y} \in \mathbb{R}^{N \times M}; \mathbf{y} = \begin{bmatrix} \mathbf{y}_1^{[1]} & \mathbf{y}_1^{[2]} & \dots & \mathbf{y}_1^{[M]} \\ \mathbf{y}_2^{[1]} & \mathbf{y}_2^{[2]} & \dots & \mathbf{y}_2^{[M]} \\ \dots & \dots & \dots & \dots \\ \mathbf{y}_N^{[1]} & \mathbf{y}_N^{[2]} & \dots & \mathbf{y}_N^{[M]} \end{bmatrix}. \quad (2)$$

The measurement pairs  $(\mathbf{u}_k^{[i]}, \mathbf{y}_k^{[i]})$  will further be called the training set. For each measurement  $k$  the sample variances of the inputs,  $\hat{\sigma}_{\mathbf{u},k}^2$ , and outputs,  $\hat{\sigma}_{\mathbf{y},k}^2$ , and the mean values  $\hat{\mathbf{u}}_k$  and  $\hat{\mathbf{y}}_k$  are determined as

$$\begin{aligned} \hat{\mathbf{u}}_k &= \frac{1}{M} \sum_{i=1}^M \mathbf{u}_k^{[i]} & \hat{\sigma}_{\mathbf{u},k}^2 &= \frac{1}{M-1} \sum_{i=1}^M (\mathbf{u}_k^{[i]} - \hat{\mathbf{u}}_k)^2 \\ \hat{\mathbf{y}}_k &= \frac{1}{M} \sum_{i=1}^M \mathbf{y}_k^{[i]} & \hat{\sigma}_{\mathbf{y},k}^2 &= \frac{1}{M-1} \sum_{i=1}^M (\mathbf{y}_k^{[i]} - \hat{\mathbf{y}}_k)^2 \end{aligned} \quad (3)$$

where  $\hat{\bullet}$  denotes the estimated values. A special case is when  $M = 1$  for which the variance cannot be calculated, but should be given a priori on the basis of the error interval of the measurement device, for example.

*Assumption 1:* The process noise and measurement noise sources  $\epsilon_{p,k}^{[i]}$ ,  $\epsilon_{\mathbf{u},k}^{[i]}$  and  $\epsilon_{\mathbf{y},k}^{[i]}$  are assumed to be stationary, independent, mutually uncorrelated and zero mean random variables with finite fourth order moments, and  $\sigma_{\mathbf{u},k} = 0; \forall k$ .  $\square$

The measured values can then be written as

$$\begin{aligned} \hat{\mathbf{u}}_k &= \mathbf{u}_{0,k} + \mathbf{n}_{\mathbf{u},k} & \mathbf{n}_{\mathbf{u},k} &= \frac{1}{M} \sum_{i=1}^M \mathbf{n}_{\mathbf{u},k}^{[i]} \\ \hat{\mathbf{y}}_k &= \mathbf{y}_{0,k} + \mathbf{n}_{\mathbf{y},k} & \mathbf{n}_{\mathbf{y},k} &= \frac{1}{M} \sum_{i=1}^M \mathbf{n}_{\mathbf{y},k}^{[i]} \end{aligned} \quad (4)$$

in which the  $\mathbf{u}_{0,k}$  and  $\mathbf{y}_{0,k}$  are the true, but unknown, values and  $\mathbf{n}_{\mathbf{u},k} = \epsilon_{\mathbf{u},k}$  and  $\mathbf{n}_{\mathbf{y},k} = \epsilon_{\mathbf{y},k} + \epsilon_{p,k}$  are the noise contributions. Since by assumption the repeated measurements  $i = 1, \dots, M$  are independent and the noise is stationary, the sample mean and sample variances converge strongly to their true values [20], viz:

$$\begin{aligned} \text{a.s.} \lim_{M \rightarrow \infty} (\hat{\mathbf{u}}_k) &= \mathbf{u}_{0,k} & \text{a.s.} \lim_{M \rightarrow \infty} (\hat{\mathbf{y}}_k) &= \mathbf{y}_{0,k} \\ \text{a.s.} \lim_{M \rightarrow \infty} (\hat{\sigma}_{\mathbf{u},k}^2) &= \sigma_{\mathbf{u},k}^2 & \text{a.s.} \lim_{M \rightarrow \infty} (\hat{\sigma}_{\mathbf{y},k}^2) &= \sigma_{\mathbf{y},k}^2 \end{aligned} \quad (5)$$

in which a.s. lim stands for *almost sure limit* [11], the limit with probability one.

### III. ON THE OUTPUT ERROR METHOD

Most current NN programs make use of the Output Error (OE) cost function to fit the model to the data (also called mapping in the theory of NN). In this section it will first be shown that the OE cost function is a good choice if a NN is only trained with noisy output and exact input samples. Next it will be shown that training the NN with noisy inputs leads to faulty NN parameters.

Assume that the noise is zero (all  $\mathbf{u}_k^{[i]} \equiv \mathbf{u}_{0,k}$  and  $\mathbf{y}_k^{[i]} \equiv \mathbf{y}_{0,k}$ ) then the OE cost function can be written as

$$K_{\text{OE}} = \frac{1}{N} \sum_{k=1}^N (\mathbf{y}_{0,k} - f_{\text{NN}}(\theta, \mathbf{u}_{0,k}))^2. \quad (6)$$

*Assumption 2:* The input is persistently exciting the system, such that the cost function (6) has a unique global minimum

$$\theta^* = \underset{\theta}{\text{argmin}} (K_{\text{OE}}) \quad (7)$$

satisfying

$$\mathbf{y}_{0,k} = f_{\text{NN}}(\theta^*, \mathbf{u}_{0,k}) \quad (8)$$

to within any given precision and for any  $N$ , including infinity. This assumption is based on the fact that neural networks are universal approximators [5] [7] [9].  $\square$

In practise it is possible that more global minima exist, e.g. by swapping two neurons within a network layer. If all the connected neurons in other layers are swapped with those neurons, the overall mapping of the NN remains the same [21]. In general this non-uniqueness problem can be solved by the use of parameter sets that lead to the same input-output relation, e.g. it is possible to demand that all neurons in a layer are sorted according to the weight and neurons in the other layers are swapped accordingly. This would lead to a unique solution of the NN parameters, without affecting the overall mapping. For simplicity we will not deal with the problem where the global minimum is not unique, since this does not contribute directly to the idea that is presented.

Consider the case where only the output measurements contain noise, for example if a known excitation is applied to a system and its noisy output is observed. The cost function (6) is then written as

$$K_{OE, n_y} = \frac{1}{N} \sum_{k=1}^N (\hat{y}_k - f_{NN}(\theta, u_{0,k}))^2. \quad (9)$$

A necessary condition for consistency of the estimates is that the true NN parameters  $\theta^*$  minimize the cost function (9) in the presence of noisy measurement samples and for  $N \rightarrow \infty$  [14]. This will first be proven for the expected value of the cost function in the following lemma.

*Lemma 1:* If a NN is trained with noisy outputs and exact inputs, the minimizer of the expected value of the cost function (9) equals the true NN parameters  $\theta^*$  under assumptions 1 and 2, or

$$\theta^* = \underset{\theta}{\operatorname{argmin}} (E\{K_{OE, n_y}\}) \quad (10)$$

*Proof:* See appendix.  $\square$

*Theorem 1:* If a NN is trained with exact input and noisy output measurements using the OE cost function (9), the estimated NN parameters strongly converge to the true NN parameters  $\theta^*$  under assumptions 1 and 2, or:

$$\text{a.s.} \lim_{N \rightarrow \infty} [\underset{\theta}{\operatorname{argmin}} (K_{OE, n_y})] = \theta^* \quad (11)$$

*Proof:* See appendix.  $\square$

*Theorem 2:* If a NN is trained with noisy input measurements using the OE cost function (9), the estimated NN parameters are in general inconsistent, or

$$\text{a.s.} \lim_{N \rightarrow \infty} [\underset{\theta}{\operatorname{argmin}} (K_{OE, n_u})] \neq \theta^* \quad (12)$$

*Proof:* See appendix.  $\square$

#### IV. THE ERRORS-IN-VARIABLES COST FUNCTION

The extra knowledge of the variances on the inputs and the outputs allows the use of the EIV cost function (13). It will first be shown that the EIV estimates are consistent with the linearized NN, such that only higher order noise terms will contribute to the inconsistency. Next it is shown that for Gaussian errors the strong consistency for the linearized NN remains if the sample variances  $\hat{\sigma}_{u,k}^2$  and  $\hat{\sigma}_{y,k}^2$  are used instead of the true variances, where at least six repeated measurements are made.

*Definition:* The EIV cost function for NN is defined as

$$EIV = \frac{1}{N} \sum_{k=1}^N \left[ M \left( \frac{(\hat{y}_k - y_k)^2}{\sigma_{y,k}^2} + \frac{(\hat{u}_k - u_k)^2}{\sigma_{u,k}^2} \right) + \lambda_k (y_k - f_{NN}(\theta, u_k)) \right] \quad (13)$$

with  $\lambda_k$  the Lagrange parameters.  $u_k$  and  $y_k$  parametrize the true but unknown input and output values which must be estimated.  $\square$

Expressing the stationarity of the Lagrange parameters  $\lambda_k$  gives

$$\frac{\partial K_{EIV}}{\partial \lambda_k} = 0 \Rightarrow y_k = f_{NN}(\theta, u_k) \quad (14)$$

and the cost function (13) reduces to

$$K_{EIV} = \frac{M}{N} \sum_{k=1}^N \left[ \frac{(\hat{y}_k - f_{NN}(\theta, u_k))^2}{\sigma_{y,k}^2} + \frac{(\hat{u}_k - u_k)^2}{\sigma_{u,k}^2} \right]. \quad (15)$$

Note that both  $\theta$  and the  $u_k$  form the unknown parameters, thus enlarging the parameter space with  $N$  additional parameters.

*Definition:* The linearized form of the EIV cost function (15) is denoted as

$$\bar{K}_{EIV} = \frac{M}{N} \sum_{k=1}^N \left[ \frac{\left( \hat{y}_k - f_{NN}(\theta, u_{0,k}) + \varepsilon_{u,k} \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \right)^2}{\sigma_{y,k}^2} + \frac{(\hat{u}_k - u_k)^2}{\sigma_{u,k}^2} \right] \quad (16)$$

in which the error on the estimated input values is defined as

$$\varepsilon_{u,k} = u_k - u_{0,k} \quad (17)$$

and the NN is replaced by its first order approximation

$$f_{NN}(\theta, u_k) \cong f_{NN}(\theta, u_{0,k}) + \varepsilon_{u,k} \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}}. \quad (18)$$

$\square$

*Lemma 2:* If a NN is trained with noisy input and output measurements using the linearized EIV cost function (16), the estimated NN parameters strongly converge to the true NN parameters  $\theta^*$ , or

$$\theta^* = \underset{\theta}{\operatorname{argmin}} (E\{\bar{K}_{EIV}\}). \quad (19)$$

*Proof:* See appendix.  $\square$

*Theorem 3:* If a NN is trained with noisy input and output measurements using the linearized EIV cost function (16), the estimated NN parameters strongly converge to the true NN parameters  $\theta^*$ , or

$$\text{a.s.} \lim_{N \rightarrow \infty} [\underset{\theta}{\operatorname{argmin}} (\bar{K}_{EIV})] = \theta^* \quad (20)$$

*Proof:* Following the same route of the proof of Theorem 1, Theorem 3 is a direct consequence of Lemma 2.  $\square$

In practise it is not possible to use the linearized EIV cost function (16) because it needs the true input values  $u_{0,k}$  which are not known. Consequently the EIV cost function (15) must be used. The effects on the NN parameters is described in the following theorem.

*Theorem 4:* If a NN is trained with noisy input and output measurements and for sufficiently large S/N ratio's, the NN parameters based on the EIV cost function (15) have a smaller bias than the NN parameters based on the OE cost function.

*Proof:* When training with noisy inputs Theorem 2 shows the inconsistency for the linearized OE estimator while Theorem 3 proves the consistency of the linearized EIV estimator. If larger noise levels are used, higher order terms come into effect and both estimators become inconsistent. For the EIV, however, only higher order moments of the noise will contribute to the inconsistency, which makes EIV more robust against input noise.  $\square$

We can conclude that the generally used Least Squares cost function leads to biased NN parameters when training with noisy input measurements. Therefore it is better to learn the NN parameters with the EIV cost function, thus reducing the bias on the parameters. Note that the bias doesn't completely disappear. Moreover, in [13] it is shown that for the linearized case and for sufficiently large signal-to-noise ratios, the EIV cost function approaches the Cramér-Rao bound [12] asymptotically when modeling with noisy inputs. This is not the case for the OE cost function. As a result the EIV qualifies as an efficient estimator for the linearized NN. The variance on the NN parameters compared to the true parameters will be less when using the EIV cost function than when using the OE cost function. Also the RMS error of the resulting NN is lower compared with the RMS error of the NN that ensue when the OE cost function is used. The examples given will further show that the advantage of the EIV is mainly in improving mapping of the details of a transfer function, and a smaller RMS error when the NN model is compared with the true model.

### Condition relaxation on the variances

For very large data sets with many repeated measurements ( $M$  large) the variances can be well estimated and equation (5) applies. In practise, the true variances of the input and output measurements are usually not known, in which case the sample variances (3) must be used.

If the noise is Gaussian distributed, [14] proves that even for small data sets ( $M \geq 6$ )

$$E\{\hat{K}_{EIV}\} = \frac{M-1}{M-3}E\{K_{EIV}\} \quad (21)$$

with

$$\hat{K}_{EIV} = \frac{M}{N} \sum_{k=1}^N \left( \frac{(\hat{y}_k - f_{NN}(\theta, u_k))^2}{\hat{\sigma}_{y,k}^2} + \frac{(\hat{u}_k - u_k)^2}{\hat{\sigma}_{u,k}^2} \right). \quad (22)$$

Thus, even for a small number of repeated measurements it is possible to replace the true variances and means by the sample variances and sample means. This has no effect on the minimization of the cost function with respect to  $\theta$ .

In the particular case that  $M = 1$  and the user can give good estimates for the variances, the EIV estimator still proves to be useful. The extra knowledge on the variances allows for a better learning of the NN parameters, and outliers due to noise on the inputs will be moved towards the NN mapping.

## V. LEARNING ALGORITHMS FOR THE EIV COST FUNCTION

It is observed that the EIV cost function used on NN, leads to an increased risk of being trapped in a local minimum during optimization, as was also observed in [22]. It is therefore advisable to use the output parameters from the OE method as the starting values for the Errors-In-Variables method. This increases the robustness, and decreases the number of needed learning steps. EIV can be regarded as a postprocessing tool to improve the accuracy of the NN parameters after OE learning and can be used on a 'no cure, no pay' basis. In the sequel two learning methods are discussed: gradient methods for large number of measurement samples, and the more robust Levenberg-Marquardt method for smaller numbers of samples.

### A. Gradient Methods (Backpropagation)

Define the errors

$$\begin{aligned} e_{f,k} &= \hat{y}_k - f_{NN}(\theta, u_k) \\ e_{u,k} &= \hat{u}_k - u_k \end{aligned} \quad (23)$$

such that  $\mathbf{e}_f = [e_{f,k}]$  and  $\mathbf{e}_u = [e_{u,k}]$  are column vectors with length  $N$ . Define the error vector  $\mathbf{e}$  and the vector of parameters  $\mathbf{p}$  as

$$\mathbf{e} = [\mathbf{e}_f^T \quad \mathbf{e}_u^T]^T \quad (24)$$

$$\mathbf{p} = [\theta^T \quad \mathbf{u}^T]^T. \quad (25)$$

For backpropagation the parameter update vector (also known as the learning rule in NN theory)  $\Delta \mathbf{p} = [\Delta \theta^T \quad \Delta \mathbf{u}^T]^T$  is calculated using

$$\Delta \mathbf{p} = -\eta \mathbf{J}^T \Phi \mathbf{e} \quad (26)$$

in which  $\eta$  is a small positive arbitrary value, called the learning rate. Methods exist to make  $\eta$  adaptive or to include a momentum term [6], but these will not be discussed here. The  $\Phi$  matrix is a diagonal matrix of size  $2N \times 2N$  that contains the variances

$$\Phi = \begin{bmatrix} \Phi_y & 0 \\ 0 & \Phi_u \end{bmatrix} \quad (27)$$

with

$$\Phi_y = \begin{bmatrix} 1/\hat{\sigma}_{y,1}^2 & 0 \\ 0 & 1/\hat{\sigma}_{y,N}^2 \end{bmatrix} \text{ and } \Phi_u = \begin{bmatrix} 1/\hat{\sigma}_{u,1}^2 & 0 \\ 0 & 1/\hat{\sigma}_{u,N}^2 \end{bmatrix}. \quad (28)$$

In (26)  $\mathbf{J}$  is the Jacobian matrix, defined as  $J_{ij} = \partial e_i / \partial p_j$  which will be denoted as  $\mathbf{J} = \partial \mathbf{e} / \partial \mathbf{p}$ . The Jacobian equals

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \quad (29)$$

with

$$\mathbf{A} = \frac{\partial f_{NN}(\theta, \mathbf{u})}{\partial \theta} \text{ and } \mathbf{B} = \frac{\partial f_{NN}(\theta, \mathbf{u})}{\partial \mathbf{u}}. \quad (30)$$

The derivation of the Jacobian matrix poses a special problem. While for the OE method only  $\mathbf{A}$  must be calculated, EIV also demands for calculation of the  $\mathbf{B}$  matrix, which has a size of  $N \times N$ . Making use of the property that  $\mathbf{B}$  is a sparse matrix, it is possible to calculate the parameter updates as:

$$\Delta \mathbf{p} = -\eta \begin{bmatrix} \mathbf{A}^T \Phi_y \mathbf{e}_f \\ \mathbf{B}^T \Phi_y \mathbf{e}_f + \Phi_u \mathbf{e}_u \end{bmatrix} \quad (31)$$

and finally

$$\begin{cases} \Delta \theta_j = -\eta \frac{1}{N} \sum_{k=1}^N \frac{e_{f,k}}{\hat{\sigma}_{y,k}^2} \frac{\partial f_{NN}(\theta, u_k)}{\partial \theta_j} \\ \Delta u_k = -\eta \left[ \frac{e_{f,k}}{\hat{\sigma}_{y,k}^2} \frac{\partial f_{NN}(\theta, u_k)}{\partial u_k} - \frac{e_{u,k}}{\hat{\sigma}_{u,k}^2} \right]. \end{cases} \quad (32)$$

It can be seen that it is no longer necessary to store the  $\mathbf{B}$  matrix in memory, although it still needs to be calculated. This indicates that the EIV method needs more calculations than the OE method and that the method is slower, with the added advantage that the method produces a more correct NN model.

## B. Levenberg-Marquardt

Here, the optimization step is defined as

$$\Delta \mathbf{p} = -(\mathbf{J}^T \Phi \mathbf{J} + \mu \mathbf{I})^{-1} (\mathbf{J}^T \Phi \mathbf{e}) \quad (33)$$

with  $\mathbf{J}$ ,  $\Phi$  and  $\mathbf{e}$  as in the previous section. Note that the terms containing second order derivatives of the Neural Network function  $f_{NN}(\theta, u_k)$  with respect to the parameters are negligent compared to the first order terms. In contrast to gradient methods, Levenberg-Marquardt (LM) optimization demands that the full Jacobian is calculated. Moreover a very large matrix must be inverted in order to calculate the optimization step. It is possible to reduce the number of calculations, based on the knowledge of  $\mathbf{J}$ , as given in equation (29). The resulting parameter update vector

$$\Delta \mathbf{p} = -\eta \begin{bmatrix} \mathbf{A}^T \Phi_y \mathbf{A} & \mathbf{A}^T \Phi_y \mathbf{B} \\ \mathbf{B}^T \Phi_y \mathbf{A} & \mathbf{B}^T \Phi_y \mathbf{B} + \Phi_u \mathbf{e}_u \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{A}^T \Phi_y \mathbf{e}_f \\ \mathbf{B}^T \Phi_y \mathbf{e}_f + \Phi_u \mathbf{e}_u \end{bmatrix} \quad (34)$$

is still very large. In practise, however, LM proves to be very robust, and convergence is reached faster than with gradient methods. The only restriction is limited available memory.

## C. Early stopping

EIV identification shapes both inputs and outputs according to the confidence level, indicated by the variances  $\sigma_u^2$  and  $\sigma_y^2$ . As a result EIV is very prone to overfitting. In the case of large variances, the measurements are typically grouped together related to the horizontal axis, allowing for an easy, but inaccurate mapping of the data. The effects of this overfitting are shown in Fig. 4:

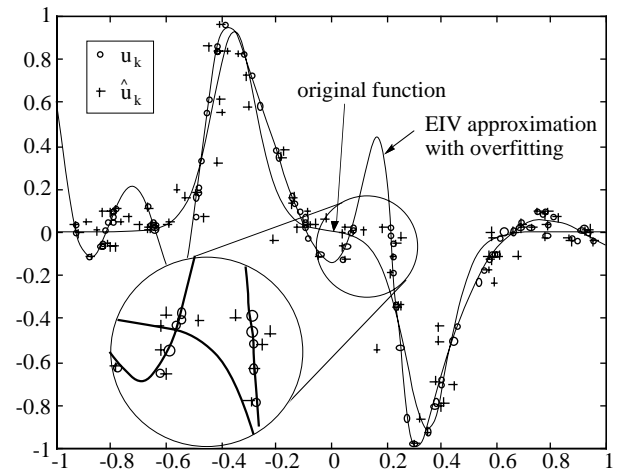


Fig. 4: The overfitting problem for EIV

An arbitrary nonlinear function

$$y = \tanh(10u + 4) - \tanh(10u + 3) + \tanh(10u - 4) - \tanh(10u - 3) \quad u \in [-1, 1] \quad (35)$$

is chosen and sampled 100 times. Gaussian, zero mean, noise is added to both inputs and outputs with  $\sigma_{u,k}^2 = \sigma_u^2 = 13\text{dB}$  and  $\sigma_{y,k}^2 = \sigma_y^2 = 13\text{dB}$ . The measurements were not repeated ( $M = 1$ ). The samples are shown as crosses on the figure. The circles are the estimated  $u_k$ , after mapping the samples using a NN with one hidden layer of 10 tan-sigmoid perceptrons. The figure shows how the estimates are drawn towards the approximated curve: most  $u_k$  lie close to the EIV approximation of the original function. However, no  $u_k$  points remain in the region  $u \in [0.081, 0.217]$ , for example, (enlarged section) and the NN approximation of the curve shows an overfitting in this region. Within the uncertainty bounds of the  $u_k^{[1]}$  values, the measurements in this region are shifted away over the horizontal axis, such that the neural network is allowed to take any arbitrary form in the resulting gap.

In the case of EIV, this overfitting can successfully be prevented by the use of a dedicated early-stopping algorithm. This is done with a validation set which is built in the same way as the training set and consists of  $N_v$  measurements with:

$$\begin{aligned} u_1 &= u_{0,1} + n_{u,1} \\ y_1 &= y_{0,1} + n_{y,1} \end{aligned} \quad l = 1 \dots N_v \times M_v \quad (36)$$

in which the index  $\bullet_v$  denotes the validation set. Note that repeated measurements ( $M_v > 1$ ) can be used to calculate the variances on the inputs and outputs, but aren't treated differently from other measurements when evaluating the early stopping criterion.

The easiest way to implement a stopping criterion is by monitoring the OE cost function

$$K = \sum_{l=1}^{N_v \times M_v} (y_l - f_{\text{NN}}(\theta, u_l))^2 \quad (37)$$

applied on the validation set and stops learning whenever this cost function starts increasing again. In the case of EIV the cost function (22) should be used with fixed NN parameters  $\theta$ , in which case the true input values  $u_{0,1}$  must be estimated prior to the evaluation of the cost function. This calls for an extra learning sequence each time validation is performed. It is evident that this results in a very slow learning process. It is possible to normalize (37) using a linearized form of the NN function, similar to the approximation used in Lemma 2. The early stopping

criterion then takes the same form as equation (59).

*Assumption 3:* Equation (59) makes use of the derivative of the NN function for the true  $u_{0,k}$  values. In practise these values are unknown. Therefore the following approximation is made

$$\frac{\partial f_{\text{NN}}(\theta, u_{0,1})}{\partial u_{0,1}} \approx \frac{\partial f_{\text{NN}}(\theta, u_1)}{\partial u_1} \quad (38)$$

This means that within the uncertainty bounds of the measurements, the NN is expected to have about the same gradient, i.e. the NN mapping should be smooth.  $\square$

If this assumption cannot be met, the only solution is the use of the OE cost function for early stopping. If the assumption is met the normalized cost function used for early stopping, is defined as

$$K_{\text{ES}} = \sum_{l=1}^{N_v \times M_v} \frac{(y_l - f_{\text{NN}}(\theta, u_l))^2}{\hat{\sigma}_{y,1}^2 + \left( \frac{\partial f_{\text{NN}}(\theta, u_l)}{\partial u_l} \right)^2 \hat{\sigma}_{u,1}^2} \quad (39)$$

in which the parameters  $\theta$  have been determined by one of the previous optimization routines. Note that the early stopping criterion is applied to the measured values only, without estimating the true  $u_{0,1}$  values.

#### D. Modeling scheme

Earlier it was stated that EIV is a rather slow method. Moreover, simulations showed that EIV demands good starting values in order to converge. Since convergence is easily reached with the OE method currently used in most mathematical packages, the following robust procedure is used:

a) With common OE methods, based on a Least Squares cost function, the starting values for the NN parameters are calculated using equation (51) and

$$\theta_{\text{OE}} = \underset{\theta}{\text{argmin}} (K_{\text{OE}, n_y, n_u}). \quad (40)$$

b) Choose the proper value for  $\mu$  or  $\eta$ , depending on the chosen optimization routine. The initial values for  $\hat{\mathbf{u}}$  and  $\hat{\sigma}^2$  are chosen according to (3). The initial NN parameters are  $\theta_1 = \theta_{\text{OE}}$  and the starting value for  $\mathbf{u}$  is taken as  $\mathbf{u}_1 = \hat{\mathbf{u}}$ .

c) From these initial values, start the following iteration

- 1) Choose a  $\Delta \mathbf{p}$  (section V.) and calculate the new parameter vector  $\mathbf{p}_{k+1} = \mathbf{p}_k + \Delta \mathbf{p}$ .
- 2) With the new  $\theta_{k+1}$  values, check the cost function (39) on the validation data. If the cost starts rising, stop the

iteration process.

## VI. SIMULATION RESULTS

### A. Curve fitting

The given method was simulated on the function  $y_k = \sin(9 \sin^3(u_k + 1.8) - 1)$  with 500 measurements  $u_k$  taken in the region  $u_k \in [e^0 \ e^1] - 1.8$ .  $M$  was chosen to be 30 and  $-5.2\text{dB}$  noise was added for a training set with  $500 \times 30$  simulation pairs  $(u_k^{[i]}, y_k^{[i]})$ . The validation set was built with a second set of 15,000 pairs. The used neural network was a two layer perceptron with 10 neurons with a hyperbolic tangent transfer function and bias. The topology of the network is shown in Fig. 5 in which the NN

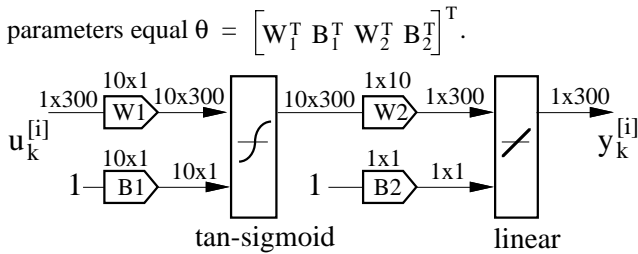


Fig. 5: Neural Network topology

The simulations were repeated 4,000 times following the procedure of the previous section and using the Levenberg-Marquardt optimization step. From these simulations we selected 3,930 simulations that had less than 20% residual least squares error between the data and the NN mapping after performing the OE optimization. I.e. 70 simulations were considered to have a bad convergence. The mean result is shown in Fig. 6. Note that different  $\hat{u}_k$  data points are taken for each simulation, dependent on the added noise. The sample means shown in the figure are only those used in the last simulation.

The mean number of iterations for OE to reach convergence in this example (using early stopping), was 22. The mean computing time to reach this convergence was approximately 6.2 seconds. The mean number of epoch needed for EIV, starting from the OE result, was 3.6. This took another 75 seconds of computing time. In 89% of the cases EIV needed less than 5 epoch to reach the new convergence point for the parameters.

The Least Squares output error on the mean of the 3,930 results, compared to the original function, dropped from 6.2% for OE to 0.8% for EIV when testing the Neural Network mappings with noiseless input data. The NN parameters were also compared using a test set of 2000 input-output pairs in which noise was added to the inputs with  $\sigma_u^2 = 13 \text{ dB}$ . The Least Squares output error, compared with the true outputs, dropped from 14% in the

OE case to 7.3% in the EIV case.

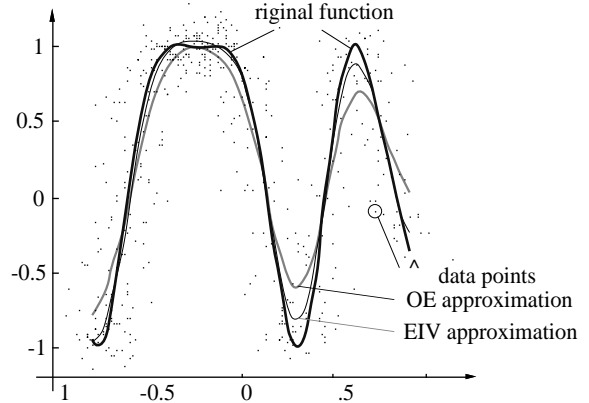


Fig. 6: Simulation results of EIV compared to OE

Fig. 7: shows a histogram of the 4,000 RMS errors of the NN mappings compared to the true functional, based on the noiseless data points. From the figure it follows that more simulations had a low RMS error after using the EIV cost function than when using the LS cost function: in average the RMS error drops from 7.5% to 5.1%. In this example the error of the EIV mapping was lower than the error of the OE mapping in 81% of the cases. Hence, the EIV mapping isn't guaranteed to perform better in all cases.

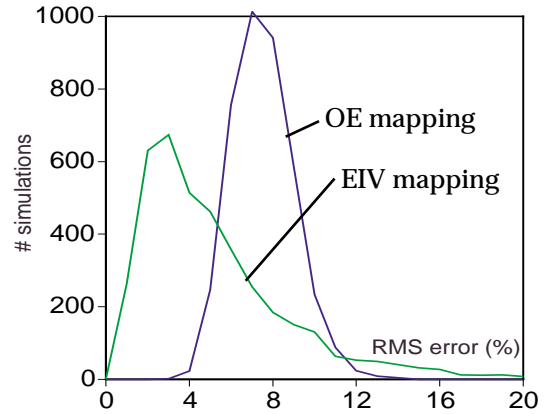


Fig. 7: Histogram of RMS errors

### B. Circle fitting with sparse data

To demonstrate the biasing effects of input noise when performing a nonlinear mapping, Amemiya [1] proposes the fitting of a circle. For the simulations a SIMO (Single-Input Multiple-Output) version of the EIV costfunction is used. One output maps the upper part of a circle shaped waveform, another output maps the lower part. The optimization of the NN parameters was done using backpropagation with an adaptive learning rate  $\eta$ .

In the first example the circle is sampled 12 times at 16 points with  $-7\text{dB}$  noise at the inputs and  $-40\text{dB}$  noise at the outputs. The data are then split into a learning set and a validation set, with the learning set of size  $16 \times 6$  with

$M = 6$  and the validation set containing 96 samples. Note that the choice for  $M$  is the minimal value required by equation (22) with sample means and sample variances. In order to compare OE with EIV, both are trained with the 16 sample means, and both use the same validation set for early stopping. Both methods use a two-layer perceptron NN architecture with 4 neurons in the hidden layer.

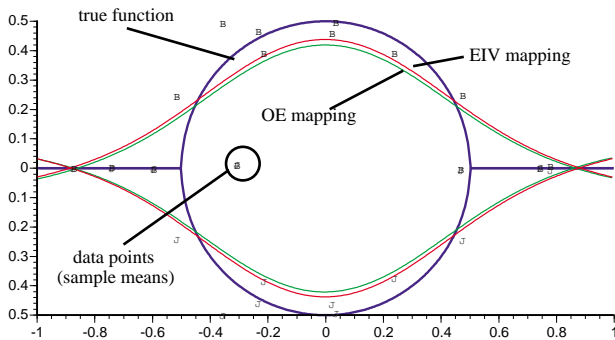


Fig. 8: Circle mapping with large bias and low number of samples

The simulations were repeated 5000 times. 70 simulations had more than 20 % residual error after training with the OE cost function, and were skipped. The mean epoch for OE to reach the point of early stopping was 5, which took about 0.4 seconds. The postprocessing with EIV needed 33 epoch on the average, which took another 2.6 seconds for each simulation.

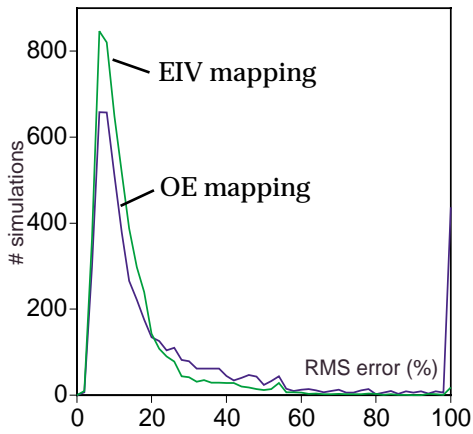


Fig. 9: Histogram of RMS errors

The mappings of both NN are compared with the original circular signal. The LS error on the mappings drops from 9.8% in the OE case to 8.9% in the EIV case. The mean results of the 5000 simulations are shown in Fig. 8: It is clear that the large noise levels have caused the mappings to be severely biased. From the figure it can be seen that the EIV mapping is slightly less biased than the OE mapping.

The EIV fit on the data outperforms the OE mapping in 69% of the 5,000 simulations and in the mean the mapping

resembles the true plant characteristics better. Fig. 9: shows a histogram of the LS errors of both mappings compared with the true function. The figure shows that there are more EIV mappings with a lower LS error, while more OE mappings have an RMS error that is above 20%.

C. Circle fitting with large data set

A second circle mapping used 3200 samples with  $M = 1$ . -15dB noise was put on the inputs and the data was split into two sets of 1600 samples. Since more samples were available, a two-layer perceptron network with 20 neurons was chosen both for OE and for EIV. The OE cost function was minimized with a Levenberg-Marquardt optimization step. On a total of 30 simulations the mean epoch was 7.7, which took 2 minutes.

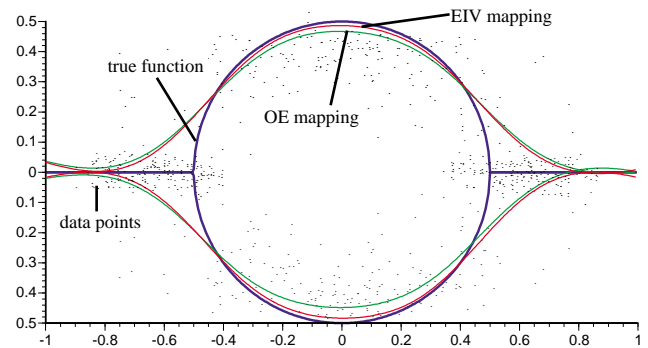


Fig. 10: Circle mapping with large bias and high number of samples

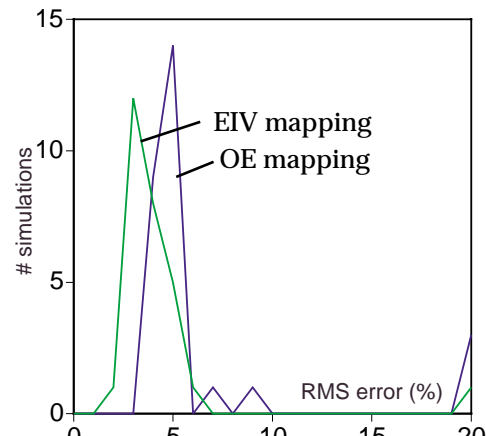


Fig. 11: Histogram of RMS errors

For the EIV postprocessing a backpropagation scheme with adaptive learning rate was needed due to the large number of samples. The mean epoch to reach the new NN parameters was 1308, needing approximately 6 hours. The error of the mappings, compared with the true function dropped from 10.3% to 4.4%. The mean result of the simulations is shown in Fig. 10:

The larger number of samples results in a significant

improvement during the postprocessing. The EIV fit on the data outperforms the OE in 93% of the cases. It is very likely that due to the large number of samples the EIV mapping is more accurate than the OE mapping. The histogram of all simulations (Fig. 11:) again shows that in general the EIV mappings have a lower RMS error when compared with the true function.

*D. Example 4: plant modeling [10]*

The quality of the steel in a steel converter is directly dependent on the end temperature of the steel in the melt. This temperature is controlled by the amount of oxygen that is blown into the melt where 20 input parameters are expected to influence this temperature. These parameters are given in table 1.

Parameter	[Min/Max]	$\sigma_u$	Unit	(I)input/ (O)output
Converter age	[1/3000]	0	days	(I)
Measure time	[1/4]	1	min	(I)
Batch type	[30/40]	0		(I)
Raw iron	[95/120]	0.36	ton	(I)
Added steel 1	[15/45]	1.22	ton	(I)
Added steel 2	[0/19]	0.99	ton	(I)
Added steel 3	[0/11]	0.57	ton	(I)
Start value C	[4/4.7]	2%	%	(I)
Goal temp.	[1650/1690]	0	° K	(I)
Oxygen	[7000/8500]	0.5%	ton	(I)
Feed type 1	[800/1800]	10	mm	(I)
Feed type 2	[500/2000]	10	mm	(I)
Feed type 3	[-150/150]	10	mm	(I)
Mn	[0.4/1.4]	1%	%	(I)
Si	[0.3/1.7]	2%	%	(I)
N <sub>2</sub>	[0/40]	0.5%	ton	(I)
Ar	[0/60]	0.5%	ton	(I)
CaO <sub>2</sub>	[5/12]	0.05	ton	(I)
Additive 1	[0/500]	10	kg	(I)
Phase 1 temp.	[1503/1683]	4	° K	(I)
Steel temperature	[1913/1983]	4	° K	(O)

Table 1: Main steel converter parameters

Some of the parameters have a nonlinear relationship to the steel temperature, and a white box identification of the plant is hardly possible. In a first stage, the goal of the NN mapping is to provide a plant model to predict the steel temperature of the plant. In a second stage an inversion of the black box model is used to predict the needed amount of pure oxygen. Since no repeated measurements are available,  $M = 1$ . The variances on the measured inputs are given by the steel company, but during simulations the dynamic range between the largest and the smallest variances was limited to 20dB.

2091 measurements were available for the mapping. From these measurements, one fifth were used for the validation set, and one seventh as a test set. The other samples were used for learning the NN parameters. The choice of the set sizes was arbitrary, keeping in mind that a maximum of samples should be reserved for learning. The NN topology used has 20 inputs, one output and 5 sigmoidal neurons in the hidden layer. The OE cost function is minimized with a Levenberg-Marquardt optimization step, while the EIV postprocessing is achieved by backpropagation with an adaptive learning rate. The simulations were repeated 5,000 times, picking different learning, validation and test sets out of the measurements on a random basis. The performance of the NN mapping is measured by the prediction of the output steel temperature. A hit is recorded if the prediction is within an interval [-10,+15] degrees Kelvin from the true end temperature. To date the hit rate of the steel plant operators is 64%.

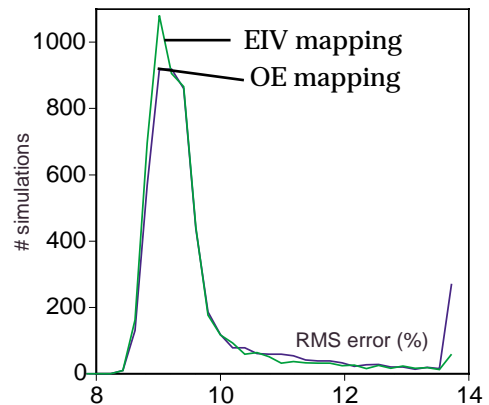


Fig. 12: Histogram of RMS errors

After the OE mappings all NN that had a hit rate below 50% were considered to have a bad convergence, such that 80 NN mappings were skipped. The mean hit rate of the OE mappings on the test set was 68.9%. The minimum on the validation set was reached in a mean of 15 iterations, which took about 12 seconds. The EIV postprocessing took another 35 seconds to reach the new minimum on the validation set in 121 iterations, using backpropagation. The mean hit rate for the EIV mapping was 69.7%. One could argue that the gain in performance (0.8%) is marginal, but it should be kept in mind that the test set is built from noisy data. In the first example it was shown that the performance decreases when using noisy samples, while the gain in performance of the EIV mapping is mainly due to a better modeling of the plant. The first example in this section also shows how the difference between OE and EIV becomes less with very low signal-to-noise ratios. Yet, even in this example the limited effort of using the EIV postprocessing leads to an improvement of the steel quality. For the sake of completeness, the histogram of the RMS errors is given in

Fig. 12: One should keep in mind that this histogram is not a comparison of the model with the true plant, though based on noisy measurements. Yet it can be seen that there is a slight improvement of the RMS error when using EIV. The EIV postprocessing leads to a better hit rate of the NN model in 71% of the simulations.

*E. Example 5: Plane mapping*

The goal of this example is the control of a flexible manipulator. Two motors control the length of two pulling cables that bend a spring into a far nonlinear state. The spring is then used as a highly flexible robot arm. The measurement of the end tip of the spring, within 4 square feet, is carried out with a magnetic inductor, which suffers from induced noise. A white box identification of the robot setup was proposed but proved to be non-inversable [24]. The goal of this example is to examine the usefulness of a NN mapping for the plant. The NN should give the position  $z$  of each motor for every given position of the end tip of the manipulator in a plane  $(x, y)$ . In order to test the method, this example makes use of an arbitrary, but well known surface, rather than the robot data.

An nonlinear surface is built on the transfer function  $z(x, y) = \sin(-3x)\cos(-5y)$  and sampled at 820 instances with  $-20\text{dB}$  noise on both inputs  $x$  and  $y$  and the output  $z$ .  $M$  is chosen as one, and the EIV cost function was trained with the true variances (20 dB). The used neural network was a two layer perceptron with 10 neurons in the hidden layer. Early stopping was used, based on a second set of 820 samples. The OE cost function is minimized with Levenberg-Marquardt, while the EIV cost function uses backpropagation.

The simulations were repeated 700 times and 634 samples had a residual error of less than 20% after the OE mapping. The mean result is shown in Fig. 14. The error compared with the true function dropped from 4.8% when using the OE cost function (15 seconds for 71 epoch) to

3.5% when using EIV (99 seconds for 447 epoch). A histogram of the RMS error over all 700 simulations is shown in Fig. 13: The figure shows that more EIV mappings have a lower RMS error. In 84% of the simulations the EIV mapping performed better than the OE mapping.

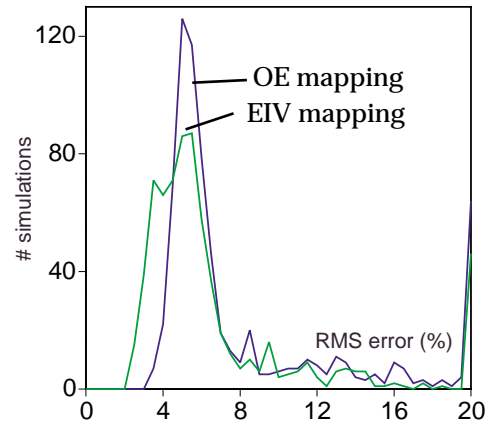


Fig. 13: Histogram of RMS errors

VII. REDUCED MODELS

Consider the case where only the output measurements contain noise. This means that all inputs are well known, or  $u_k^{[i]} \equiv u_k \equiv u_{0,k}$ , thus eliminating the second term in the cost function (15). The EIV algorithm will then be reduced to Bayesian neural network learning [2]. When the variances are not known, it is possible to set all  $\sigma_{y,k}^2 \equiv 1$ . In this case the EIV cost function is reduced to the OE cost function. The generally used OE method is, therefore, a special case of the EIV method.

Another possibility is when only the input measurements contain noise. An example where this can happen is when the data are used to invert a plant transfer function, based on a number of measurements [24]. The optimization reduces to a constrained nonlinear minimization based on a cost

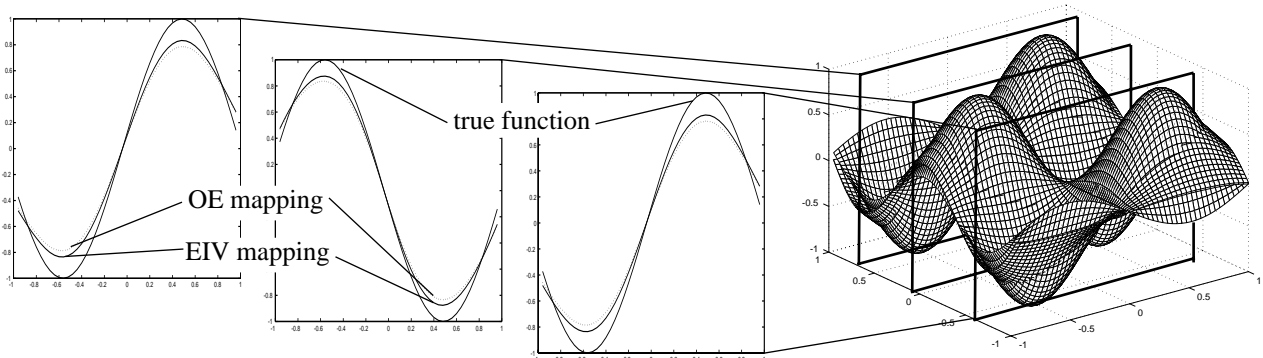


Fig. 14. NN mapping of a plane

$$\mathbf{K}' = \frac{1}{N} \sum_{k=1}^N \frac{(\hat{u}_k - u_k)^2}{\hat{\sigma}_{u,k}^2} \quad (41)$$

with constraints

$$y_k = f_{NN}(\theta, u_k). \quad (42)$$

Instead of seeking a new optimization routine, the existing routines can be used by setting the  $\hat{\sigma}_{y,k}^2$  in equation (22) to a very small value.

## VIII.CONCLUSION

This paper shows that the generally used Least Squares cost function on Neural Networks can cause severe biasing effects on the network parameters, if noisy input data is used to train the network. This bias cannot be avoided, regardless of the type of NN used, and regardless of the learning scheme used to learn the NN parameters.

The Errors-In-Variables cost function is introduced to neural network identification and it is shown that this cost function will reduce biasing effects on the NN parameters. The extra cost to perform the identification is the required knowledge of the measured data variances. The gain of the algorithm is the improved performance of the identification when both the input data and output data contain noise or when only the input data contain noise. The drawback of the method is an increased demand for computing time and memory needs. It must be stressed that the derived NN parameters obtained from noisy input and output measurements with the EIV cost function are meant for use with noiseless input data, as is the case for inverting control and simulation purposes. However, simulations also indicate an improved performance when using the EIV neural network parameters with noisy input data.

It has been demonstrated how the EIV estimator can be used as a postprocessing tool for the most popular Output Error estimator. Also a dedicated early stopping algorithm has been presented. This paper also presented an optimization scheme on the implementation of EIV with Backpropagation and Levenberg-Marquardt. Moreover, different examples have been given to demonstrate the optimization scheme.

## IX.APPENDIX

### *Proof of Lemma 1*

Replace  $\hat{y}_k$  in the cost function (9) by the expression given in (4). This results in the cost function

$$\mathbf{K}_{OE, n_y} = \frac{1}{N} \sum_{k=1}^N (y_{0,k} + n_{y,k} - f_{NN}(\theta, u_{0,k}))^2. \quad (43)$$

Knowing that

August 18, 1999

$$E\{(n_{y,k})^2\} = \sigma_{y,k}^2/M \quad (44)$$

and

$$E\{n_{y,k}(y_{0,k} - f_{NN}(\theta, u_{0,k}))\} = 0 \quad (45)$$

the expectation value of (43) becomes

$$E\{\mathbf{K}_{OE, n_y}\} = \frac{1}{N} \sum_{k=1}^N \left[ (y_{0,k} - f_{NN}(\theta, u_{0,k}))^2 + \frac{\sigma_{y,k}^2}{M} \right]. \quad (46)$$

The first term on the right hand side of this equation equals the cost function (6). The variances on the output,  $\sigma_{y,k}^2$ , are  $\theta$ -independent such that

$$\arg \min_{\theta} (E\{\mathbf{K}_{OE, n_y}\}) = \arg \min_{\theta} (\mathbf{K}_{OE}) = \theta^* \quad (47)$$

which proves the lemma.  $\square$

### *Proof of Theorem 1*

In [14] it is proven that under assumption 1

$$\text{a.s.} \lim_{N \rightarrow \infty} [\mathbf{K}_{OE, n_y} - E\{\mathbf{K}_{OE, n_y}\}] = 0 \quad (48)$$

uniformly in  $\theta$ . Thus

$$\text{a.s.} \lim_{N \rightarrow \infty} [\arg \min_{\theta} (\mathbf{K}_{OE, n_y}) - \arg \min_{\theta} (E\{\mathbf{K}_{OE, n_y}\})] = 0. \quad (49)$$

With Lemma 1 this implies that

$$\text{a.s.} \lim_{N \rightarrow \infty} [\arg \min_{\theta} (\mathbf{K}_{OE, n_y}) - \theta^*] = 0 \quad (50)$$

which proves the strong consistency of the NN parameters [18].  $\square$

### *Proof of Theorem 2*

When using noisy inputs and outputs, the cost function is written as

$$\mathbf{K}_{OE, n_y, n_u} = \frac{1}{N} \sum_{k=1}^N (\hat{y}_k - f_{NN}(\theta, \hat{u}_k))^2. \quad (51)$$

Redo the expectation calculation on  $\mathbf{K}_{OE, n_y, n_u}$  with the inputs given in equation (4). First consider the case where the noise contributions  $n_{u,k}$  are small, so that the following first order Taylor series approximation of  $f_{NN}$  can be made:

$$f_{NN}(\theta, \hat{u}_k) \cong f_{NN}(\theta, u_{0,k}) + n_{u,k} \frac{\partial (f_{NN}(\theta, u_{0,k}))}{\partial u_{0,k}}. \quad (52)$$

Note that this approximation differs from the approximation made in (18) where here, the true input parameters are compared to the input measurements instead of the

estimated parameters. Denote  $\bar{K}_{OE, n_y, n_u}$  as the linearized OE cost function. Putting (52) in (51) the expectation value for this cost function becomes

$$E\{\bar{K}_{OE, n_y, n_u}\} = \frac{1}{N} \sum_{k=1}^N \left[ (y_{0,k} - f_{NN}(\theta, u_{0,k}))^2 + \frac{\sigma_{y,k}^2}{M} + \frac{\sigma_{u,k}^2}{M} \left( \frac{\partial(f_{NN}(\theta, u_{0,k}))}{\partial u_{0,k}} \right)^2 \right]. \quad (53)$$

When the same reasoning is followed as for equation (46), it can be seen that in this case the term

$$\frac{\sigma_{u,k}^2}{M} \left( \frac{\partial(f_{NN}(\theta, u_{0,k}))}{\partial u_{0,k}} \right)^2$$

is not  $\theta$ -independent. Therefore, the minimizer  $\arg \min_{\theta} (E\{\bar{K}_{OE, n_y, n_u}\})$  is not guaranteed to equal  $\theta^*$ . This means that the OE estimates are inconsistent in the presence of input noise. In the case where the  $n_{u,k}$  become larger, higher order terms of the Taylor expansion (52) must be taken into account. These terms are also  $\theta$ -dependent and the errors deteriorate.  $\square$

### Proof of Lemma 2

The true input values  $u_k$  are found by minimizing the linearized EIV cost function (16) w.r.t.  $u_k$ , or by demanding that  $\partial \bar{K}_{EIV} / \partial u_k = 0$ . This gives (after some calculations)

$$u_k = \frac{1}{\sigma_{y,k}^2 + \left( \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \right)^2 \sigma_{u,k}^2} \left[ \sigma_{u,k}^2 \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \left( f_{NN}(\theta, u_{0,k}) + u_{0,k} \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} - \hat{y}_k \right) + \hat{u}_k \sigma_{y,k}^2 \right]. \quad (54)$$

With this expression for  $u_k$  the cost function (16) can be written as

$$\bar{K}_{EIV} = \frac{M}{N} \sum_{k=1}^N \frac{1}{\sigma_{y,k}^2 + \left( \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \right)^2 \sigma_{u,k}^2} \left( \hat{y}_k - f_{NN}(\theta, u_{0,k}) + (u_{0,k} - \hat{u}_k) \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \right)^2. \quad (55)$$

In order to assess the behaviour of this cost function for large  $N$  the expected value  $E\{\bar{K}_{EIV}\}$  is calculated. For this the  $\hat{u}_k$  and  $\hat{y}_k$  are replaced by the values given in (4). Since

$$E\left\{ n_{y,k} \left( y_{0,k} - f_{NN}(\theta, u_{0,k}) + (u_{0,k} - \hat{u}_k) \frac{\partial f_{NN}}{\partial u_{0,k}} \right) \right\} = 0, \quad (56)$$

$$E\left\{ n_{u,k} \frac{\partial f_{NN}}{\partial u_{0,k}} (y_{0,k} - f_{NN}(\theta, u_{0,k})) \right\} = 0 \quad (57)$$

and

$$E\left\{ n_{y,k}^2 + n_{u,k}^2 \left( \frac{\partial f_{NN}}{\partial u_{0,k}} \right)^2 \right\} = \sigma_{y,k}^2 + \sigma_{u,k}^2 \left( \frac{\partial f_{NN}}{\partial u_{0,k}} \right)^2, \quad (58)$$

(55) reduces to

$$E\{\bar{K}_{EIV}\} = M + \frac{M}{N} \sum_{k=1}^N \frac{(y_{0,k} - f_{NN}(\theta, u_{0,k}))^2}{\sigma_{y,k}^2 + \left( \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \right)^2 \sigma_{u,k}^2}. \quad (59)$$

It is possible to follow the same reasoning again as for lemma 1 and theorem 2. Note that (59) is minimal in the true  $\theta^*$  parameters, or

$$\arg \min_{\theta} (E\{\bar{K}_{EIV}\}) = \arg \min_{\theta} (K_{OE}) \quad (60)$$

which proves the lemma for the linearized NN.  $\square$

### Acknowledgments

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction (IUAP 4/2), initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture; and the Flemish Government (GOA, IMMI2). The research is supported by the Fund for Scientific Research - Flanders (FWO).

Parts of this work were done in cooperation with the Dept. of Measurement and Information Systems of the Technical University of Budapest, Hungary.

### References

- [1] Y. Amemiya, W. A. Fuller, "Estimation for the Nonlinear Functional Relationship", *The Annals of Statistics*, Vol. 16, No. 1, 1988, pp. 147-160.
- [2] C. M. Bishop, "Neural Networks for Pattern Recognition", Oxford: Clarendon Press, 1995.
- [3] M. Deistler, "Linear Dynamic Errors-In-Variables Models", *Journal of Applied Probability*, Vol. 23, 1986, pp. 23 - 39.
- [4] R. Fletcher, "Practical Methods of Optimization", John Wiley & Sons Ltd., 1987-'91.
- [5] K.-I. Funahashi, "On the Approximate Realization of Continuous Mappings by Neural Networks", *Neural Networks*, Vol. 2, 1989, pp. 183-192.
- [6] J. A. Freeman, D. M. Skapura, "Neural Networks, Algorithms, Applications and Programming Techniques", Addison-Wesley Publishing Co., 1991.
- [7] S. S. Ge, "Robust Adaptive Control of Robots Based on Static Neural Networks", 13th Triennial World Congress of the IFAC, San Francisco, USA, 1996, pp. 139-144.

- [8] A. G. Green and David J.C. MacKay, "Bayesian analysis of linear phased-array radar", 1997, Cavendish Laboratory, Cambridge, CB3 0HE. U.K.
- [9] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks", *Neural Networks*, Vol. 4, 1991, pp. 251-257.
- [10] G. Horváth, B. Pataki and Gy. Strausz, "Black-box Modeling of a Complex Industrial Process", Proc. of the 1999 IEEE Conference and workshop on Engineering of Computer Based Systems, March 1999, Nashville, TN. pp. 60-66.
- [11] E. Lukacs, "Stochastic Convergence", Academic Press, New York, 1975.
- [12] J. P. Norton, "An Introduction to Identification", Academic Press Inc., London, 1986.
- [13] R. Pintelon, P. Guillaume, Y. Rolain and F. Verbeyst, "Identification of Linear Systems Captured in a Feedback Loop", *IEEE Trans. Instrum. Meas.*, Vol. IM-41, No 6, 1992, pp. 747-754.
- [14] J. Schoukens, R. Pintelon and Y. Rolain, "Maximum Likelihood Estimation of Errors-In-Variables Models Using a Sample Covariance Matrix Obtained from Small Data Sets", *Recent Advances in Total Least Squares Techniques and Errors-In-Variables Modeling*, 1997, pp. 59-68.
- [15] J. Sjöberg, "Non-Linear System Identification with Neural Networks", Linköping Studies in Science and Technology Dissertation No. 381, Linköping University, Sweden, 1995.
- [16] J. Sjöberg, L. Ljung, "Overtraining, Regularization and Searching for a Minimum with Application to Neural Networks", *International Journal on Control*, Vol. 62, 1995, pp. 1391 - 1407.
- [17] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B Delyon, P. Glorennec, H. Hjalmarsson and A. Juditsky, "Nonlinear Black-Box modeling in System Identification: A Unified Overview", *Automatica*, Vol. 31, No. 12, 1995, pp. 1691 - 1724.
- [18] T. Söderström, "Convergence Properties of the Generalised Least Squares Identification Method", *Automatica*, Vol. 10, 1974, pp. 617-626.
- [19] T. Söderström, "Identification of Stochastic Linear Systems in Presence of Input Noise", *Automatica*, Vol. 17, No. 5, 1981, pp. 713 - 725.
- [20] A. Stuart & J. K. Ord, "Kendall's Advanced Theory of Statistics, Distribution Theory", Vol. 1, Charles Griffin & Co., 1987.
- [21] H. J. Sussmann, "Uniqueness of the Weights for Minimal Feedforward Nets With a Given Input-Output Map", *Neural Networks*, Vol. 5, 1992, pp. 589-593.
- [22] G. Vandersteen, "Identification of Linear and Nonlinear Systems in an Errors-in-Variables Least Squares and Total Least Squares Framework", PhD thesis, April 1997, Vrije Universiteit Brussel, 1050 Brussel, Belgium.
- [23] G. Vandersteen, Yves Rolain, Johan Schoukens, Rik Pintelon, "On the Use of System Identification for Accurate Parametric Modeling of Nonlinear Systems Using Noisy Measurements", *IEEE Transactions on Instrumentation and Measurement*, Vol. 45, No. 2, April 1996, pp.605-609.
- [24] J. Van Gorp, "Control of a Static Nonlinear Plant Using a Neural Network Linearization", *IJCNN International Joint Conference on Neural Networks 1998*, Anchorage, Alaska, pp. 2136 - 2141.
- [25] J. Van Gorp, J. Schoukens, R. Pintelon, "Adding Input Noise to Increase the Generalization of Neural Networks is a Bad Idea", *ANNIE 1998, Intelligent Engineering Systems Through Artificial Neural Networks*, Vol. 8, pp. 127 - 132.