

# The Errors-In-Variables Cost Function For Learning Neural Networks With Noisy Inputs

Jürgen Van Gorp, Johan Schoukens and Rik Pintelon

ANNIE 1998, Intelligent Engineering Systems Through Artificial Neural Networks, Volume 8, pp. 141 - 146.

**Abstract** — Currently, most learning algorithms for neural network modelling 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, but special care must be taken when training with noisy input data, or when both inputs and outputs contain noise if the derived NN parameters will be used with exact inputs, as is the case in simulations or inverting control. This paper introduces a novel cost function for learning NN with noisy inputs, based on the Errors-In-Variables cost function. A learning scheme is presented and examples are given demonstrating the improved performance in Neural Network curve fitting, at the cost of performing more calculations.

## I. INTRODUCTION

In system identification it is the goal to map measurement samples on a given model. The parameters of the model are optimized in order to fit the measurements. In this scope linear identification, Neural Networks (NN), Fuzzy Logic (FL), ... can all be seen as different models in a general identification frame. The chosen model puts some restrictions on the complexity of the identified system. In contrast to Linear Identification NN are considered as black box models, that can map (identify) nonlinear surfaces. In general, the identification is done in four steps:

- Perform input-output measurements.
- Choose the model (Linear, FL, NN, ...)
- Choose a cost function, e.g. Least Squares (LS) or Weighted Least Squares (WLS).
- Optimize the parameters so that the model fits the measurements, by minimizing the cost function. In NN this is called learning. Known optimization methods are Gradient Methods (e.g. Backpropagation) and Levenberg-Marquardt.

In the past much effort has been put in the introduction of different models and different learning schemes in the NN domain. However, little work has been done on choosing the proper cost function. Mostly the LS cost function is used. In

this paper will be shown that using LS with noisy input data causes biasing in the NN parameters.

In the past noise on the outputs was considered within the theory of Bayesian inference, introduced to Neural Networks by Green and MacKay [4] and Bishop [1]. Using Bayesian techniques on both inputs and outputs leads to a novel estimator, based on the Errors-In-Variables (EIV) cost function. The EIV cost that is presented in this paper is already known in linear identification [7] and introduced to nonlinear systems in [9]. The inherent black box structure of NN will cause the EIV technique to be prone to overfitting, for which this paper introduces an early stopping technique based on Bayesian techniques.

## II. PROBLEM STATEMENT

Consider a Single Input Single Output (SISO) linear or nonlinear system (<sup>1</sup>), as shown in Fig. 1. For this system we seek an Neural Network (NN) black box model  $y = f_{NN}(\theta, \mathbf{u})$ , with  $\theta$  the neural network parameters,  $\mathbf{u}$  the input and  $y$  the output of the system. The function  $f_{NN}$  applied on a vector, is taken elementwise.

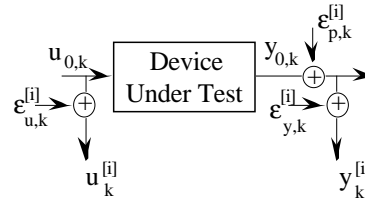


Fig. 1. Measurement setup

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. A general representation of  $\mathbf{u}$  and  $\mathbf{y}$  is

$$\mathbf{u} \in \mathfrak{R}^{N \times M}; \mathbf{u} = \begin{bmatrix} u_1^{[1]} & u_1^{[2]} & \dots & u_1^{[M]} \\ u_2^{[1]} & u_2^{[2]} & \dots & u_2^{[M]} \\ \dots & \dots & \dots & \dots \\ u_N^{[1]} & u_N^{[2]} & \dots & u_N^{[M]} \end{bmatrix} \quad (1)$$

and

<sup>1</sup>The author is with the V.U.B., Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium, E-mail: jvgorp@vub.ac.be

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

$$\mathbf{y} \in \mathfrak{R}^{N \times M}; \mathbf{y} = \begin{bmatrix} y_1^{[1]} & y_1^{[2]} & \dots & y_1^{[M]} \\ y_2^{[1]} & y_2^{[2]} & \dots & y_2^{[M]} \\ \dots & \dots & \dots & \dots \\ y_N^{[1]} & y_N^{[2]} & \dots & y_N^{[M]} \end{bmatrix} \quad (2)$$

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

$$\begin{aligned} u_k &= \langle u_k^{[i]} \rangle_i & \sigma_{u,k}^2 &= \text{Var}_i(u_k^{[i]}) \\ y_k &= \langle y_k^{[i]} \rangle_i & \sigma_{y,k}^2 &= \text{Var}_i(y_k^{[i]}) \end{aligned} \quad i = 1 \dots M \quad (3)$$

A special case is when  $M = 1$ . In this case the variance can not be calculated, but should be given by estimation, based on the confidence interval of each measurement.

*Assumption 1:* The process noise and measurement noise sources,  $\epsilon_p$  and  $\epsilon_m$ , are considered to be independent, mutually uncorrelated, zero mean Gaussian distributed random variables such that  $\sigma_{u,y,k} = 0; \forall k$ .

The measured values can then be written as

$$\begin{aligned} u_k^{[i]} &= u_{0,k} + n_{u,k}^{[i]} \\ y_k^{[i]} &= y_{0,k} + n_{y,k}^{[i]} \end{aligned} \quad (4)$$

in which the  $u_{0,k}$  and  $y_{0,k}$  are the true, but unknown, values and  $n_{u,k}^{[i]}$  and  $n_{y,k}^{[i]}$  are the noise contributions. Under assumption 1 and for  $M$  very large, the expectation of the mean input and output values converges strongly to their true values (Proof: see [8]), or

$$M \rightarrow \infty \Rightarrow \begin{cases} E_i\{u_k^{[i]}\} = u_{0,k} \\ E_i\{y_k^{[i]}\} = y_{0,k} \end{cases} \quad (5)$$

### III. ON THE OUTPUT ERROR METHOD

Most current NN programs make use of the Output Error (OE) cost function in order to perform the identification. Assume that the noise is zero (all  $u_k^{[i]} \equiv u_{0,k}$  and  $y_k^{[i]} \equiv y_{0,k}$ ) then the OE cost function can be written as

$$K_{OE} = \frac{1}{N} \sum_{k=1}^N (y_{0,k} - f_{NN}(\theta, u_{0,k}))^2 \quad (6)$$

*Assumption 2:* It is possible to find the true neural network parameters  $\theta^*$  based on the cost function (6) with

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

such that

$$y_{0,k} = f_{NN}(\theta^*, u_{0,k}) \quad (8)$$

to within any given precision. This assumption is based on the fact that neural networks are universal approximators.

Now consider the case in which only the output measurements contain noise. This will mostly be the case if a system is to be identified where the perturbation on the system is well known, and the output is measured in a noisy environment.

*Lemma 1:* The OE cost function is unbiased for output noise.

*Proof:* The cost function (6) becomes

$$K_{OE, n_y} = \frac{1}{N} \sum_{k=1}^N \frac{1}{M} \sum_{i=1}^M (y_k^{[i]} - f_{NN}(\theta, u_{0,k}))^2 \quad (9)$$

Replace  $y_k^{[i]}$  by the expression given in (4). Knowing that

$$E_i\{(n_{y,k}^{[i]})^2\} = \sigma_{y,k}^2 \quad (10)$$

the expectation value of (9) becomes

$$E\{K_{OE, n_y}\} = \frac{1}{N} \sum_{k=1}^N \left[ (y_{0,k} - f_{NN}(\theta, u_{0,k}))^2 + \sigma_{y,k}^2 \right] \quad (11)$$

in which  $\sigma_{y,k}^2$  is  $\theta$ -independent such that

$$\underset{\theta}{\text{argmin}} (K_{OE, n_y}) = \underset{\theta}{\text{argmin}} (K_{OE}) = \theta^* \quad (12)$$

which means that  $K_{OE}$  is unbiased for output noise.  $\square$

*Lemma 2:* If a neural network is used with noisy input data or in a measurement environment where both inputs and outputs contain noise, the OE estimator is severely biased.

*Proof:* The cost function will now be

$$K_{OE, n_y, n_u} = \frac{1}{N} \sum_{k=1}^N \frac{1}{M} \sum_{i=1}^M (y_k^{[i]} - f_{NN}(\theta, u_k^{[i]}))^2 \quad (13)$$

Redo the expectation calculation on  $K_{OE, n_y, n_u}$  with the inputs given in equation (4). First consider the case where the noise contributions  $n_{u,k}^{[i]}$  are very small, so that the following approximation of  $f_{NN}$  can be made:

$$f_{NN}(\theta, u_k^{[i]}) \cong f_{NN}(\theta, u_{0,k}) + n_{u,k}^{[i]} \frac{\partial(f_{NN}(\theta, u_{0,k}))}{\partial u_{0,k}} \quad (14)$$

With the property that [8]

$$\frac{1}{\sigma_u \sqrt{2\pi}} \int_{-\infty}^{+\infty} (n_{u,k})^{2r} e^{-\frac{(n_{u,k})^2}{2\sigma_u^2}} d(n_{u,k}) = \frac{\sigma_u^{2r} (2r)!}{2^r r!} \quad (15)$$

the expectation value for the OE cost function becomes

$$E\{K_{OE, n_y, n_u}\} = \frac{1}{N} \sum_{k=1}^N \left[ (y_{0,k} - f_{NN}(\theta, u_{0,k}))^2 + \sigma_{y,k}^2 + \sigma_{u,k}^2 \left( \frac{\partial(f_{NN}(\theta, u_{0,k}))}{\partial u_{0,k}} \right)^2 \right] \quad (16)$$

When the same reasoning is followed as for equation (12), it can be seen that in this case the extra terms are not  $\theta$ -independent and  $\underset{\theta}{\operatorname{argmin}}(K_{OE, n_y, n_u})$  will no more equal  $\theta^*$ . This means that the cost function is biased for input noise. In the case where the  $n_{u,k}^{[i]}$  become larger, also higher order terms of the Taylor expansion (14) must be taken into account and the bias error will only become worse.  $\square$

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

The extra knowledge on the variances on the inputs and the outputs allows the introduction of the Errors In Variables (EIV) cost function

$$K_{EIV} = \frac{1}{N} \sum_{k=1}^N \left[ \frac{1}{M} \sum_{i=1}^M \left[ \frac{(y_k^{[i]} - \hat{y}_{0,k})^2}{\sigma_{y,k}^2} + \frac{(u_k^{[i]} - \hat{u}_{0,k})^2}{\sigma_{u,k}^2} \right] + \lambda_k (\hat{y}_{0,k} - f_{NN}(\theta, \hat{u}_{0,k})) \right] \quad (17)$$

The  $\lambda_k$  are the Lagrange parameters while  $\hat{u}_{0,k}$  and  $\hat{y}_{0,k}$  are estimated values of respectively  $u_{0,k}$  and  $y_{0,k}$ . The interest lays only in the NN parameters  $\theta$  and not in the estimation of the  $y_{0,k}$ , such that it makes sense to enforce the equality

$$\hat{y}_{0,k} = f_{NN}(\theta, \hat{u}_{0,k}) \quad (18)$$

Then the cost function (17) reduces to

$$K_{EIV} = \frac{1}{N} \sum_{k=1}^N \frac{1}{M} \sum_{i=1}^M \left[ \frac{(y_k^{[i]} - f_{NN}(\theta, \hat{u}_{0,k}))^2}{\sigma_{y,k}^2} + \frac{(u_k^{[i]} - \hat{u}_{0,k})^2}{\sigma_{u,k}^2} \right] \quad (19)$$

Notice that both  $\theta$  and the  $\hat{u}_{0,k}$  form the unknown parameters, which enlarges the parameter space with  $N$  parameters.

*Lemma 3:* If  $M$  is take very large the EIV cost function is less biased than the OE cost function with respect to noise on both inputs and outputs.

*Proof:* Define the error on the estimated input values as

$$\varepsilon_{u,k} = u_{0,k} - \hat{u}_{0,k} \equiv \sum_{i=1}^M (u_k^{[i]} - \hat{u}_{0,k}) \quad (20)$$

and assume the linearized form of the NN

$$f_{NN}(\theta, \hat{u}_{0,k}) = f_{NN}(\theta, u_{0,k}) + \varepsilon_{u,k} \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \quad (21)$$

Use equations (20) and (21) in the EIV cost function (19).  $\varepsilon_{u,k}$  is calculated by demanding that  $\partial K_{EIV} / \partial \varepsilon_{u,k} = 0$ . It can then be shown that the EIV cost function reduces to

$$K_{EIV} \equiv \frac{1}{N} \sum_{k=1}^N \frac{1}{M} \sum_{i=1}^M \frac{(y_k^{[i]} - f_{NN}(\theta, u_{0,k}))^2}{\sigma_{y,k}^2 + \left( \frac{\partial f_{NN}(\theta, u_{0,k})}{\partial u_{0,k}} \right)^2 \sigma_{u,k}^2} \quad (22)$$

which is the Bayesian representation of the cost function (9) from Lemma 1. The denominator is a normalization factor that makes use of the variance of the output and the variance of the input that is fed through the linearized NN. Just as in Lemma 1 minimization with respect to  $\theta$  will give the true NN parameters. Comparing equation (22) with equation (16) shows that in the case of a first order approximation the EIV cost function is even unbiased. When larger noise levels are used, higher order terms come into effect and the EIV cost function will be biased too. However, in the case of EIV only higher order terms of the noise will lead to bias, which makes it more robust against input noise.  $\square$

In the particular case that  $M$  is small or even  $M = 1$  the property (20) no longer holds and the convergence of (19) will no more be uniform. However, in practise the extra knowledge on the variances still leads to better 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

In practise it is observed that the EIV cost function used on NN, leads to an increased possibility of being trapped in a local minimum during optimization. This was also observed in [9]. For that reason it is advisable to use the output parameters from the OE method as the starting parameters of the Errors-In-Variables method. This will not only reduce the robustness, but also decreases the number of needed learning steps. As such EIV can be regarded as a postprocessing tool to increase the accuracy of the NN parameters after OE learning. Further, two learning methods will be discussed: gradient methods for large number of measurement samples, and the more robust Levenberg-Marquardt method for smaller numbers of samples ( $N < 500$ ).

### A. Gradient Methods (Backpropagation)

Define the errors

$$\begin{aligned} e_{f,k}^{[i]} &= y_k^{[i]} - f_{NN}(\theta, \hat{u}_{0,k}) \\ e_{u,k}^{[i]} &= \hat{u}_{0,k} - u_k^{[i]} \end{aligned} \quad (23)$$

such that  $\mathbf{e}_f = [e_{f,k}^{[i]}]$  and  $\mathbf{e}_u = [e_{u,k}^{[i]}]$ . Define the error vector  $\mathbf{e}$  as  $\mathbf{e} = [\mathbf{e}_f(\cdot) \ \mathbf{e}_u(\cdot)]^T$ . Define  $\mathbf{p} = [\theta^T \ \mathbf{u}_0^T]^T$  as the vector of parameters. For backpropagation the parameter update vector (also called the learning rule in NN theory)  $\Delta\mathbf{p} = [\Delta\theta^T \ \Delta\mathbf{u}_0^T]^T$  is calculated using

$$\Delta\mathbf{p} = -\eta\mathbf{J}^T\Phi\mathbf{e} \quad (24)$$

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 [3], but these will not be discussed here. The  $\Phi$  matrix is a diagonal matrix that contains the variances

$$\Phi = \begin{bmatrix} 1/\sigma_y^2 & 0 \\ 0 & 1/\sigma_u^2 \end{bmatrix} \quad (25)$$

$\mathbf{J}$  is the Jacobian matrix, defined as  $\mathbf{J} = \partial\mathbf{e}/\partial\mathbf{p}$  and equals

$$\mathbf{J} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ 0 & \mathbf{I} \end{bmatrix} \quad (26)$$

with

$$\mathbf{A} = \frac{\partial(f_{NN}(\theta, \hat{\mathbf{u}}_0))}{\partial\theta} \text{ and } \mathbf{B} = \frac{\partial(f_{NN}(\theta, \hat{\mathbf{u}}_0))}{\partial\hat{\mathbf{u}}_0} \quad (27)$$

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  $N \times N$ . However, making use of the property that  $\mathbf{B}$  is a sparse matrix, it is possible to calculate the parameter updates as:

$$\Delta\mathbf{p} = -\eta \begin{bmatrix} \mathbf{A}^T\sigma_y^{-2}\mathbf{e}_f(\cdot) \\ \mathbf{B}^T\sigma_y^{-2}\mathbf{e}_f(\cdot) + \sigma_u^{-2}\mathbf{e}_u(\cdot) \end{bmatrix} \quad (28)$$

and finally

$$\begin{cases} \Delta\theta_j = -\eta \frac{1}{N} \sum_{k=1}^N \frac{1}{M} \sum_{i=1}^M \frac{e_{f,k}^{[i]} \partial(f_{NN}(\theta, \hat{u}_{0,k}))}{\sigma_{y,k}^2 \partial\theta_j} \\ \Delta u_{0,k} = -\eta \frac{1}{M} \sum_{i=1}^M \left[ \frac{e_{f,k}^{[i]} \partial(f_{NN}(\theta, \hat{u}_{0,k}))}{\sigma_{y,k}^2 \partial\hat{u}_{0,k}} - \frac{e_{u,k}^{[i]}}{\sigma_{u,k}^2} \right] \end{cases} \quad (29)$$

It can be seen that the calculation of the  $\mathbf{B}$  matrix is no more necessary and the calculation time is significantly reduced. This already indicates that the EIV method needs more memory than the OE method and that the method is slower, with the added gain that the method produces a more correct NN model.

### B. Levenberg-Marquardt

The optimization step is in this case defined as

$$\Delta\mathbf{p} = -(\mathbf{J}^T\Phi\mathbf{J} + \lambda\mathbf{I}) \setminus (\mathbf{J}^T\Phi\mathbf{e}) \quad (30)$$

with  $\mathbf{J}$ ,  $\Phi$  and  $\mathbf{e}$  the same as in the previous section. Remark that the terms containing second order derivatives of the Neural Network function  $f_{NN}(\theta, \hat{u}_{0,k})$  with respect to the parameters, have been neglected compared to the first order terms. In contradiction 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 on  $\mathbf{J}$ , as given in equation (26). The resulting parameter update vector

$$\Delta\mathbf{p} = -\eta \begin{bmatrix} \mathbf{A}^T\sigma_y^{-2}\mathbf{A} & \mathbf{A}^T\sigma_y^{-2}\mathbf{B} \\ \mathbf{B}^T\sigma_y^{-2}\mathbf{A} & \mathbf{B}^T\sigma_y^{-2}\mathbf{B} + \sigma_u^{-2}\mathbf{e}_u(\cdot) \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{A}^T\sigma_y^{-2}\mathbf{e}_f(\cdot) \\ \mathbf{B}^T\sigma_y^{-2}\mathbf{e}_f(\cdot) + \sigma_u^{-2}\mathbf{e}_u(\cdot) \end{bmatrix}$$

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 caused by the amount of 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, allowing for an easy, but inaccurate mapping of the data.

The effects of this overfitting are shown in Fig. 2. An arbitrary nonlinear function

$$y = \tanh(10x + 4) - \tanh(10x + 3) + \tanh(10x - 4) - \tanh(10x - 3) \quad (31)$$

is chosen and sampled 100 times with 10dB Gaussian, zero mean, noise on both inputs and outputs. The measurements were not repeated, so that  $M = 1$ . The samples are shown as crosses on the figure. The circles are

the estimated  $\hat{u}_{0,k}$ , after mapping the samples using a NN with one hidden layer of 10 tan-sigmoid perceptrons. The figure shows how the original measurements are drawn towards the approximated curve. Remark that most  $\hat{u}_{0,k}$  lay close to the EIV approximation of the original function. However, no  $\hat{u}_{0,k}$  points remain in the regions  $(u, y) = (-0.44, -0.3)$  and  $(u, y) = (0.3, 0.42)$  and the NN approximation of the curve show an overfitting in these regions. Within the uncertainty bounds of the  $u_k^{[i]}$  values, the measurements in these regions are shifted away, 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  $N_l$  measurements with:

$$\begin{aligned} u_l &= u_{0,l} + n_{u,l} \\ y_l &= y_{0,l} + n_{y,l} \end{aligned} \quad l = 1 \dots N_l \times M_l \quad (32)$$

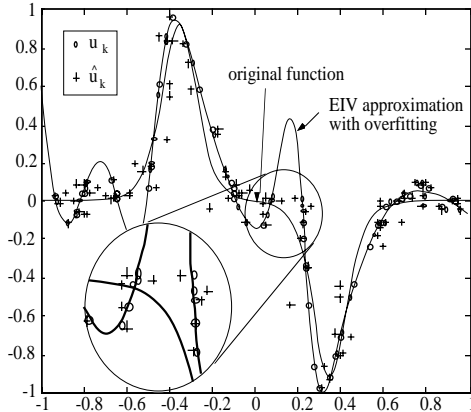


Fig. 2. The overfitting problem for EIV

Remark that repeated measurements ( $M_l > 1$ ) can be used to calculate the variances on the inputs and outputs, but aren't treated different from other measurements when evaluating the early stopping criterion. The easiest way to implement a stopping criterion is by monitoring the OE cost function (33) applied on the validation set and stop learning whenever this cost function starts raising again. However, it is reasonable to use the information that is given by the variances of the validation set, in order to become an optimized stopping criterion, by normalizing the OE cost function with the variance of the cost. To do so, the variance on the OE cost function

$$K = \sum_{l=1}^{N_l \times M_l} (y_l - f_{NN}(\theta, u_l))^2 \quad (33)$$

is calculated using a first order Taylor expansion of the Neural Network mapping, similar to the approximation of

equation (21) in Lemma 3. The early stopping criterion takes the same form as in equation (22).

*Assumption 3:* Equation (22) makes use of the derivative of the NN function, taken in the true  $u_{0,k}$  value. In practise this value is unknown, and therefore it will be assumed that

$$\frac{\partial f_{NN}(\theta, u_{0,l})}{\partial u_{0,l}} \cong \frac{\partial f_{NN}(\theta, u_l)}{\partial u_{0,l}} \quad (34)$$

The normalized cost function, used for early stopping, is then defined as

$$K_{ES} = \sum_{l=1}^{N_l \times M_l} \frac{(y_l - f_{NN}(\theta, u_l))^2}{\sigma_{y,l}^2 + \left( \frac{\partial f_{NN}(\theta, u_l)}{\partial u_{0,l}} \right)^2 \sigma_{u,l}^2} \quad (35)$$

in which the parameters  $\theta$  are the ones that have been found with the optimization routine of section V. Remark that the early stopping algorithm is applied without the need of estimating the  $\hat{u}_{0,l}$  parameters.

#### D. Learning procedure

Earlier 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 easier reached with the OE method currently used in most mathematical packages, the following procedure is used, which proved to be very robust:

a) With common OE methods, based on a Least Squares cost function, the starting values for the NN parameters are calculated using equation (13)

$$\theta_{OE} = \underset{\theta}{\operatorname{argmin}} (K_{OE, n_y, n_u}) \quad (36)$$

b) Choose the proper value for  $\lambda$  or  $\eta$ , depending on the chosen optimization routine. The initial values for  $\hat{u}$  are chosen according to (3):  $\hat{u}_{0,k} = \langle u_k^{[i]} \rangle_i$ . The initial NN parameters are  $\theta_1 = \theta_{OE}$ .

c) From these starting values, start the following iteration

- Choose a  $\Delta p$  (section V.) and calculate the new parameter vector  $p_{k+1} = p_k + \Delta p$ .
- With the new  $\theta_{k+1}$  values, check the cost function (35) on the validation data. If the cost starts raising, stop the iteration process.

## VI. SIMULATION RESULTS

The given method was simulated on the function  $y = \sin(9 \sin^3(u + 1.8) - 1)$  with  $u$  taken in the region  $u \in [e^0 \ e^1] - 1.8$ . 5dB noise was added to become a training set with 300 simulation pairs  $(u_k^{[1]}, y_k^{[1]})$ .  $M$  was

chosen to be 1. The validation set was built with a second set of 300 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. 3 with  $\theta = [W_1^T B_1^T W_2^T B_2^T]^T$

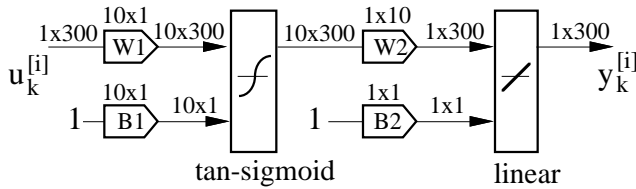


Fig. 3. Neural Network topology

The simulations were repeated 338 times following the procedure of the previous section and using the Levenberg-Marquardt optimization step. From these simulations 288 simulations were taken that had less than 20% residual least squares error after performing the OE optimization, i.e. 50 simulations were considered to have a bad convergence. The mean result is shown in Fig. 4

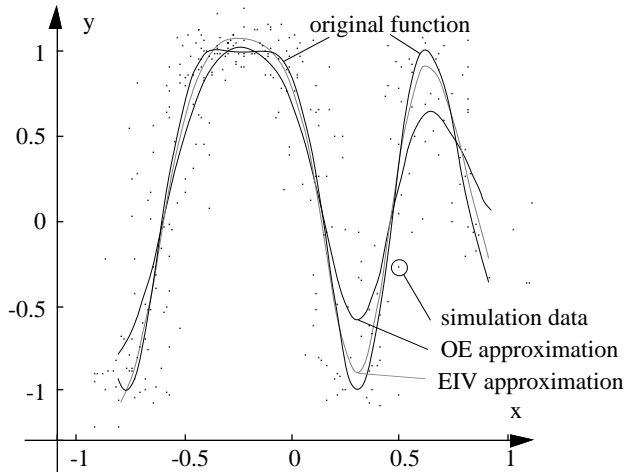


Fig. 4. Simulation results of EIV compared to OE

The mean number of iterations for OE to reach convergence in this example, was 22. The computing time for one simulation was approximately 19 seconds. The mean number of epoch needed for EIV, starting from the OE result, was 3.7. This took another 105 seconds of computing time. The Least Squares output error compared to the original function dropped from 8.8% for OE to 4% for EIV when using the NN with noiseless input data.

## VII. 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 can not be avoided, regardless of which type of NN is used, and regardless which learning scheme is used to learn the NN parameters.

The Errors-In-Variables (EIV) cost function is presented to perform neural network identification. It is shown that this cost function will reduce biasing effects on the NN parameters. The extra cost to perform the identification is the knowledge on the variances of the measured data. The gain of the algorithm lays in the improved performance of the identification when both the input data and output data contain noise or when only the input data contains 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 with the EIV cost function are meant for use with noiseless input data, as is the case for inverting control and simulation purposes.

A method is given how the EIV estimator can be used as a postprocessing tool for the mostly used Output Error estimator, and a dedicated early stopping algorithm is presented.

## 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 scientific responsibility rests with its authors.

## References

- [1] Christopher M. Bishop, "Neural Networks for Pattern Recognition", Oxford : Clarendon Press, 1995, ISBN 0-19-853864-2
- [2] R. Fletcher, "Practical Methods of Optimization", John Wiley & Sons Ltd., 1987-'91, ISBN 0-471-91547-5
- [3] James A. Freeman, David M. Skapura, "Neural Networks, Algorithms, Applications and Programming Techniques", Addison-Wesley Publishing Co., 1991, ISBN 0-201-51376-5
- [4] Andrew G. Green and David J.C. MacKay, "Bayesian analysis of linear phased-array radar", 1997, Cavendish Laboratory, Cambridge, CB3 0HE. U.K.
- [5] Lukacs, "Stochastic Convergence", Academic Press, New York, 1975
- [6] Radford M. Neal, "Bayesian Training of Backpropagation Networks by the Hybrid Monte Carlo Method", 1992, Technical Report CRG-TR-92-1, Dep. of Computer Science, University of Toronto
- [7] Johan Schoukens and Rik Pintelon, "Identification of Linear Systems", Pergamon Press - London, 1991, ISBN 0-08-040734-X
- [8] Alan Stuart & J. Keith Ord, "Kendall's Advanced Theory of Statistics, Distribution Theory", Vol. 1, Charles Griffin & Co., 1987
- [9] Gerd 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
- [10] Jürgen Van Gorp, "Control of a Static Nonlinear Plant Using a Neural Network Linearization", submitted for IEEE on Neural Networks, August 1997