Iterative Neural Network Model Inversion

Annamária R. Várkonyi-Kóczy, András Rövid

Dept. of Measurement and Information Systems, Budapest University of Technology and Economics
Budapest, Hungary
Integrated Intelligent Systems Japanese-Hungarian Laboratory
e-mail: koczy@mit.bme.hu
e-mail: rovid@mit.bme.hu

Abstract: Recently model based techniques have become wide spread in solving measurement, control, identification, etc. problems. For measurement data evaluation and for controller design also the so called inverse models are of considerable interest. In this paper a technique to perform neural network inversion is introduced. For discrete time inputs the proposed method provides good performance if the iterative inversion is fast enough compared to system variations, i.e. the iteration is convergent within the sampling period applied. The proposed method can be considered also as a simple nonlinear state observer, which reconstructs the selected inputs of the neural network from its outputs.

1 Introduction

In measurement and information processing systems the model based schemes play very important role. The basically linear approaches to fault diagnosis [1], optimal state estimation [2] and controller design are well understood and successfully combined with adaptive techniques (see. e.g. [3]) to provide optimum performance. There is a wide variety of possible models to be applied based on both classical methods [4] and recent advances in handling [5] information. Nonlinear techniques, however, are far from this maturity or still are not well understood. Furthermore, in a lot of cases, the exact mathematical model is not available or is too complex to be handled. Even if we can build usable models, in most of the cases they can be used only with limitations and are not universal enough to solve a larger family of nonlinear problems. In such cases soft computing based modeling can very effectively be used in different problems. The efforts on the field of fuzzy and neural network (NN) based modeling and control, however, seem to result in a real breakthrough also in this respect.

Using model based techniques in identification, measurement, and control also the inverse models play an important role [3]. Serious research has been done e.g. in fuzzy model inversion, however in most of the cases inverse models can be derived only with direct limitations on the models applied (see e.g. [6] [7]). The inversion technique reported in [2] [8] follows a different approach and is based on the quite general concept of state observation widely used in measurement and filtering applications. The key element of this concept is to force a model of a physical system to "copy" the behavior of the system to be observed (see Fig. 1). This scheme is the so called observer structure which is a common structural representation for the majority of iterative data and signal processing algorithms. Traditionally the observer is a device to measure the states of dynamic systems having state variable representation. These states, however, can be regarded as unknown inputs, and therefore their "copy" within the observer as the result of model inversion.

Concerning neural networks we can find just a few attempts for NN inversion (see e.g. [9] [10] [11]). Furthermore, we would like to note that most of the reported methods do not cover any universal and systematic approach, but can (very advantageously) be used to invert concrete neural network models. For example, such a method is the inversion of the NN based underwater acoustic model [12] and the proposed method can invert just the concrete kind of NNs. If we have other types of problems we have to design different inversion techniques for those types of NNs.

In this paper a new, more universal neural network iteration technique is proposed which is very similar to the methods described in [2] [8]. Is also uses the observer based concept except that it is able to invert MIMO NN models. The paper is organized as follows: The possible role of inverse neural network models and the main features of the explicit inversion methods are described in Section 2. Section 3 presents the observer based iterative inversion technique. A simple example illustrating the proposed method is given in Section 4, while Section 5 provides the conclusions.

2 Inverse Neural Network Models

Nowadays solving computer based measurement and control problems involves model-integrated computing. This integration means that the available knowledge finds a proper form of representation and becomes an active component of the computer program to be executed during the operation of the measuring and control devices. The role of the inverse models in measurements is obvious: observations are mappings from the measured quantity. This mapping is performed by a measuring channel the inverse model of which is inherent in the data/signal processing phase of the measurement. In control applications inverse plant models are to be applied as controllers in feedforward (open-loop) systems, as well as in various alternative control schemes. Additionally, there are very successful control structures incorporating both forward and inverse plant models (see e.g. [3]).

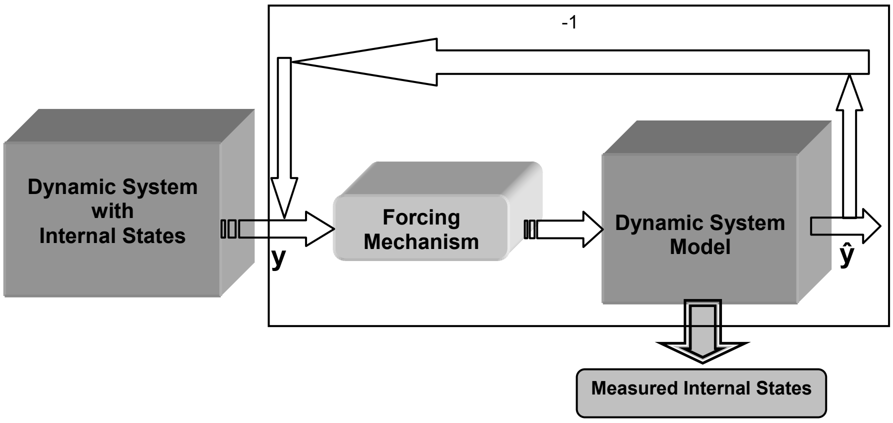

Figure 1
The observer concept

When we refer to “inversion” of a neural network for the acquisition of certain input parameters, we are actually referring to a constrained inversion. That is to say, given the functional relationship of the NN input to the output, we have the forward and want to have the inverse relationships (see Fig. 2), where $x_1, x_2, \ldots, x_n$ are the input of the forward neural network model and $\mathbf{y}$ is the output vector of the model. Fig. 2b illustrates the inverse NN model, where output vector $\mathbf{x}^s$ is composed of a subset of the input variables of the forward model. Given the forward relationship (Eq. 2) described in a neural network form, we wish to find an inverse mapping (Eq. 3). It is obvious from these figures that if the forward neural network model is available only the inverse (nonlinear static) mapping must be derived. There are different alternatives to perform such a derivation. One alternative is to invert the neural network model using the classical regression technique based on input-output data. To solve this regression problem iterative algorithms can also be considered. The result of such a procedure is an approximation of the inverse and the accuracy of this approximation depends on the efficiency of the model fitting applied.

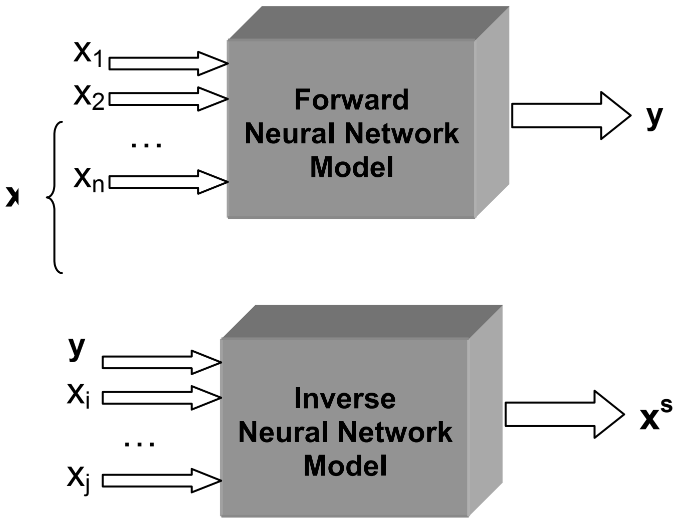

Figure 2
Forward and inverse neural network models.

$$\mathbf{x} = \{x_1, \ldots, x_n\},\ \mathbf{x}^s \subseteq \mathbf{x} \equiv \{x_1, \ldots, x_n\} \setminus \{x_i, \ldots, x_j\}$$

3 Concept of the Observer Based Iterative Inversion

The general concept of the observer is represented by Fig. l. The physical system produces output **y** and we suppose that its behavior can be described by a dynamic system model. This system description becomes the inherent part of the measurement procedure and is forced to behave similarly to the physical system. If the correction (forcing) mechanism is appropriate the observer will converge to the required state and will produce the estimate of the unknown input. The strength of this approach is that this iterative evaluation is easy to implement, e.g., using standard digital signal processors. The complete system can be embedded into a real-time environment, the necessary number of iterations to get the inverse can be performed within one sampling time slot of the measurement or control application. For the correction several techniques can be proposed based on the vast literature of numerical methods (see e.g. [13]) since the proposed iterative solution is nothing else than the numerical solution of a single (or multi) variable nonlinear equation (system).

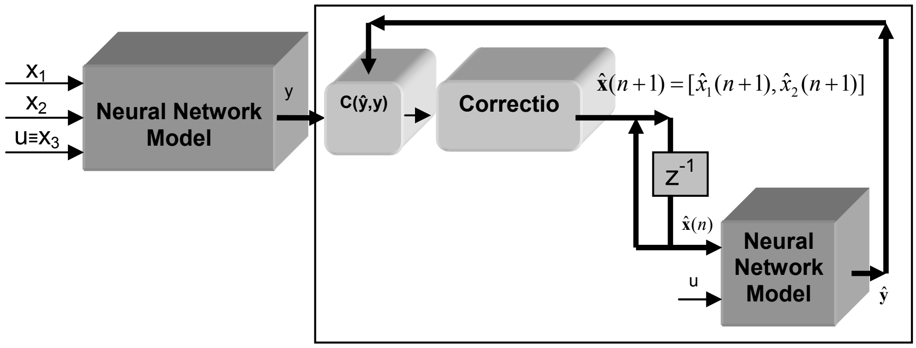

Figure 3
Block diagram of the iterative inversion scheme for the three input two output example, where $\mathbf{y} = (y_1, y_2)$ and $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2)$ are the output vectors.

The iteration is based on the following general formula:

$$\hat{\mathbf{x}}(n+1) = \hat{\mathbf{x}}(n) + correction[C(\hat{\mathbf{y}}, \mathbf{y}), \hat{\mathbf{x}}(n), f(.), \mu] \quad (1)$$

where *f(.)* stands for nonlinear function to be inverted, *C* is some kind of error function between the "real" and estimated output, and μ denotes the step size. When we refer to "inversion" of a NN implementing **y**=*f*(**x**) for the acquisition of certain input parameters, we are not merely trying to find $\mathbf{x}=f^{-1}(\mathbf{y})$, rather having the forward relationship (2) in an NN form we are looking for the inverse relationship (3).

$$\mathbf{y} = f(\mathbf{x}, \mathbf{u}) \quad (2)$$

$$\mathbf{x} = g(\mathbf{y}, \mathbf{u}) \quad (3)$$

Here **x** stands for the unknown environmental parameters we wish to obtain, and **u** denotes a vector of the known environmental and system parameters. In Fig. 3 a three input two output example can be followed.

In the simplest case, as iteration method we can use the Newton iteration, which has the form of:

$$\mathbf{x}(n+1) = \mathbf{x}(n) - \mu \nabla C(\mathbf{x}(n)) \quad (4)$$

where **x** denotes the for input vector of the unknowns and *C* stands for mean square error function used by the iteration process:

$$C(\hat{\mathbf{y}}, \mathbf{y}) = \sum_i (\hat{y}_i - y_i)^2 . \quad (5)$$

Since usually we do not know the exact mathematical description of the cost function, the partial derivatives of function *C* must be evaluated locally using simple numerical techniques:

$$\frac{\partial C}{\partial x_1} = \frac{C(x_1 + \Delta, x_2) - C(x_1 - \Delta, x_2)}{2\Delta} \tag{6}$$

$$\frac{\partial C}{\partial x_2} = \frac{C(x_1, x_2 + \Delta) - C(x_1, x_2 - \Delta)}{2\Delta} \tag{7}$$

The computational complexity of this iterative procedure depends mainly on the complexity of the forward neural network model itself. It is anticipated, however, that after the first convergence if the input of this observer changes relatively smoothly then relatively few iterations will be required to achieve an acceptable inverted value.

Unfortunately, the simple Newton iteration may fail if the error function *C* has multiple minima. This is because gradient-based techniques work using locally available information. If multiple minima may occur, global search techniques are to be applied. For decreasing the computational complexity of the iteration we can also use the combined technique described in [8], which means that the iteration is based on the simple Newton technique but if it fails it is switched to a global e.g. genetic algorithm based method. Our examinations show that the computational complexity of the method is in the range of the complexity of the simple Newton iteration, thus the introduction of the combined technique results in significant improvement in the convergence property with only a tolerable increase of the computational complexity.

4 Examples

In this section, to illustrate the proposed iterative inversion method, a simple example with a three input two output forward NN model is presented (see Fig. 3). The purpose of this model is to describe the dependency of the 3D point coordinates and its 2D projection. This mapping is very important at the 3D object reconstruction from digital images [14].

The 3D point **M** projects to the image point **m** (see Fig. 4a). The orthogonal projection of camera position **C** onto an image plane **Ω** is the principal point **O** and axis **z** corresponding to this projection line is called principal axis.

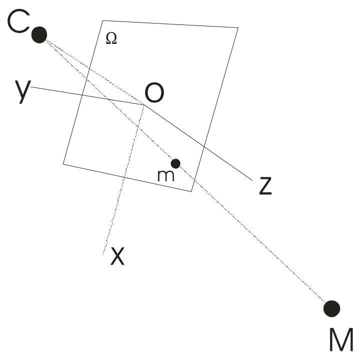

Figure 4a
Illustration of 3D point **M** and its projection **m** in the image plane **Ω**. The position of the camera is at the origin of the coordinate system

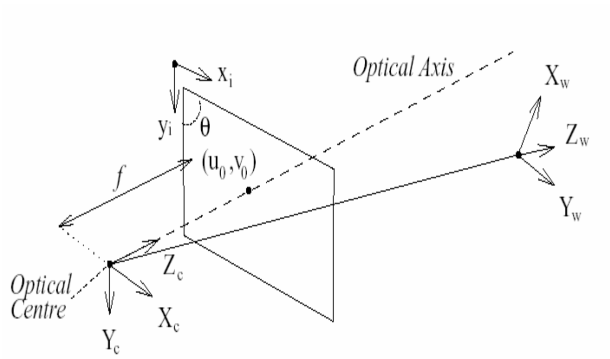

Figure 4b
Illustration of the camera movement, which causes the change of the elements of the projection matrix.

(*f*-focal length, X_c, Y_c, Z_c – camera position and orientation, X_w, Y_w, Z_w – world coordinate-system)

As we know, the projection of **M** onto an image plane is unambiguous. Inversely it is not valid. To an image point a line corresponds in the 3D space, not a single point, thus this mapping is not unambiguous and therefore to invert this mapping it is necessary to perform some modifications. To solve this problem one from the 3D point coordinates should be given. That means, if we have the image point coordinates of the projection of a 3D point **M** and also one coordinate component of the point **M** is given, then it is possible to perform unambiguous mapping from 2D to 3D. If the center of projection is at the origin **C** of the 3D reference frame of the space and the image plane is parallel to the **xy** plane and displaced a distance *f* (focal length) along the **z** axis from the origin, then the projection of a 3D point **M**=[X,Y,Z,1] with homogeneous coordinates can be calculated as:

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} . \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{8}$$

where f is the focal length, s stands for the scale factor and x, y denote the coordinates of the projection **m** in the image plane. In real images, the origin of the image coordinates is not the principal point and the scaling along each image axis is different, so the image coordinates undergo a further transformations as rotation and shifting (see Fig. 4b). In this case the projection matrix can be described by the following matrix [14]:

$$\begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix} . \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{9}$$

where a, b,…,l represent the entries of the perspective projection matrix.

As an example, we implemented the projection and its inverse operation by NNs. We used a system where the camera is not at the center of the world coordiante system (see Fig. 4b). In the illustrated example the projection matrix was chose to [1 1 2 0; 3 1 2 0; 1 4 1 0]. The Z component of the 3D point coordinates was chosen for 10. For the training of our feedforward backpropagation neural network we used sample pairs in which the input are the 3D point coordinates while the output their projections. For simplicity, the focal length is chosen for 1. The network has 2 hidden layers with 15 neurons. The searched X, Y values, which are the output of the inverse NN correspond to the minimum of the error surface (the simplest case is illustrated in Fig. 5 while in Fig. 6 a more general problem is shown). The searching procedure is performed according to the iterative searching algorithm described above. Figs. 7, 8 represent the output of the forward NN, while in Figs. 9,10 the output of the inverse NN can be followed. Fig 11 illustrates the mean square error (MSE) of the inversion, i.e. the MSE of the obtained 3D point (by the inversion) compared to the original one which was projected to the plain.

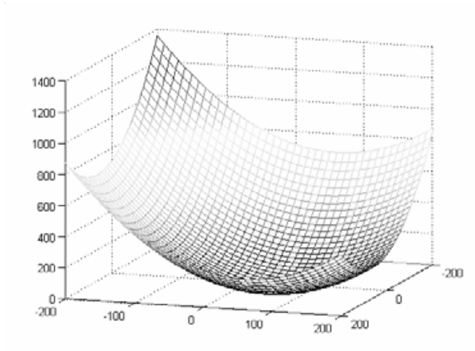

Figure 5
The surface of the error function *C* in the case when the position of the camera is at the origin of the world coordinate system.

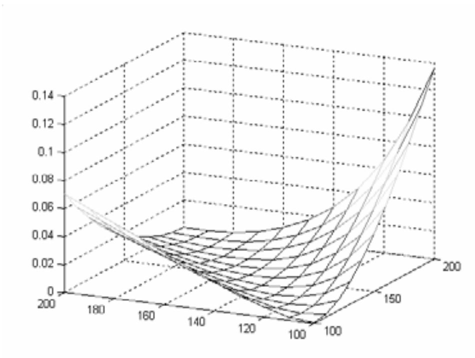

Figure 6
The surface of the error function *C* in the case when the position of the camera is not at the origin of the world coordinate system. (Projection matrix: [1 1 2 0; 3 1 2 0; 1 4 1 0]).

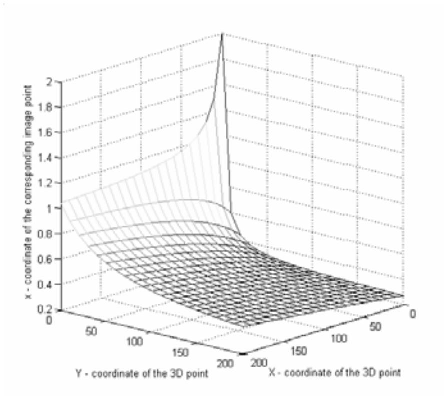

Figure 7
The *x* component of the output of the forward neural network model.
(Projection matrix: [1 1 2 0; 3 1 2 0; 1 4 1 0]).

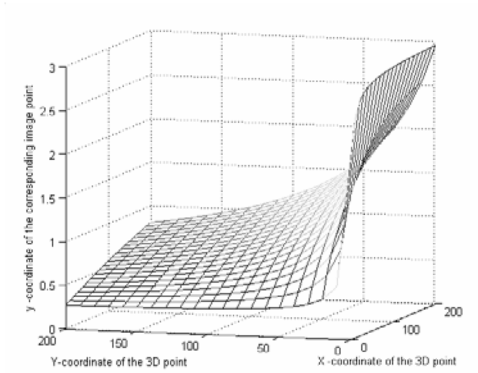

Figure 8
The *y* component of the output of the forward neural network model.
(Projection matrix: [1 1 2 0; 3 1 2 0; 1 4 1 0]).

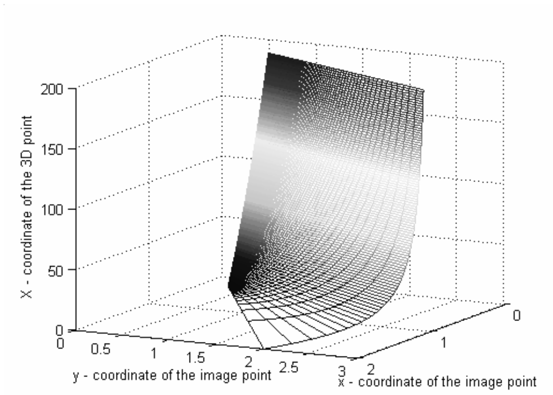

Figure 9
The output *X* values of the inverse NN

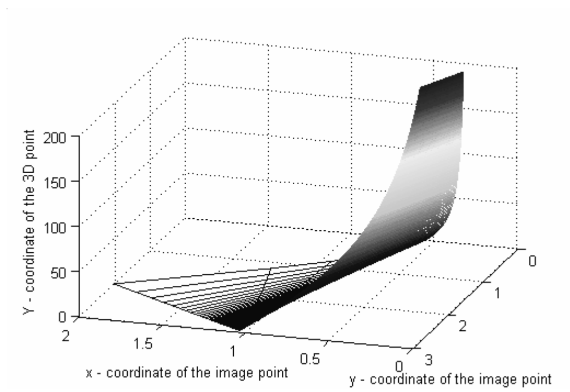

Figure 10
The output *Y* values of the inverse NN

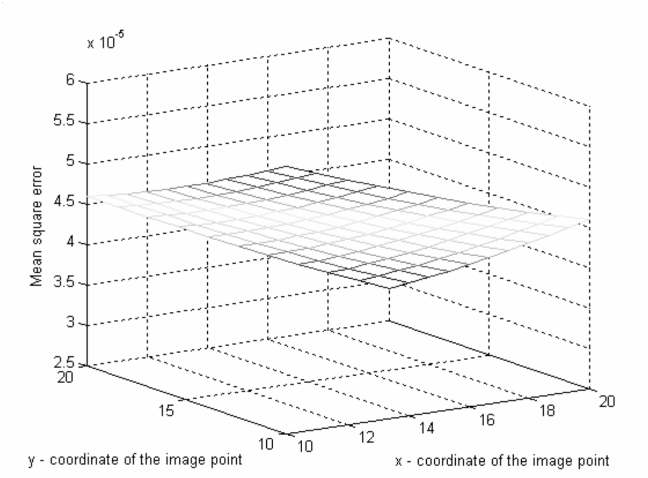

Figure 11
The mean square error of the inversion

Conclusions

In this paper an iterative method has been presented to solve the inversion of neural network models for identification, measurement, and control, etc. applications. The derivation of this iterative technique is related to the state observer concept. The proposed technique can effectively be used for different kind of feedforward MIMO NN models.

References

[1] B. D. O. Anderson, J. B. Moore, Optimal Filtering, Prentice-Hall, Inc., Englewood Cliffs, N. J. 1979

[2] R. Patton, P. Frank and R. Clark, Fault Diagnosis in Dynamic Systems, Prentice Hall International (UK) Ltd., 1989

[3] B. Wildrow, E. Walach, Adaptive Inverse Control, Prentice Hall, 1996

[4] S. A. Billings, 'Identification of Nonlinear Systems – A Survey,“ IEEE Proc. Vol. 127, Pt. D, No. 6, Nov. 1980, pp. 272-284

[5] G. J. Klir, T. A. Folger, Fuzzy Sets, Uncertainty, and Information, Prentice-Hall International, Inc., 1988

[6] P. Baranyi, P. Korondi, H. Hashimoto and M. Wada, Fuzzy Inversion and Rule Base Reduction, INES'97, pp. 301-306

[7] C. Batur, A. Shrinivasan and Chieng-Chung Chan. Inverse Fuzzy Model Controllers, Proc. of the American Control Conf., San Francisco, California, pp. 772-774, June 1993

[8] Várkonyi-Kóczy, A. R., A. Álmos, and T. Kovácsházy, "Genetic Algorithms in Fuzzy Model Inversion." In Proc. of the 8th IEEE Int. Conference on Fuzzy Systems, FUZZ-IEEE'99, Seoul, Korea, Aug. 22-25, 1999, Vol. 3, pp. 1421-1426

[9] Jensen, C. A.; Reed, R. D.; Marks, R. J., II; El- Sharkawi, M. A.; Jae-Byung Jung; Miyamoto, R. T.; Anderson, G. M.; Eggen, C. J., "Inversion of feedforward neural networks: algorithms and applications", Proceedings of the IEEE, Volume: 87 9, Sept. 1999, pp. 1536 –1549

[10] Benson, J., Chapman, N. R., and Antoniou, A., "Geoacoustic Inversion with Artificial Neural Networks," OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century, Volume: 1, 1999, pp. 446-451

[11] A. Linden and J. Kindermann: Inversion of multilayer nets, Proc. IJCNN, pp. 425-430, 1989

[12] Thompson, B. B. Marks, R. J., II El-Sharkawi, M. A. Fox, W. J. Miyamoto, R. T., "Inversion of neural network underwater acoustic model for estimation of bottom parameters using modified particle swarm optimizers," In Proceedings of the International Joint Conference on Neural Networks, 20-24 July 2003, vol. 2, pp. 1301- 1306

[13] F. S. Acton, Numerical Methods Thut Work, The Mathematical Association of America, Washington D.C., 1990

[14] Rövid, A., Várkonyi-Kóczy, A. R., Várlaki, P., "3D Model Estimation from Multiple Images," In Proc. of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'2004, Budapest, Hungary, July 25-29, 2004, pp. 1661-1666