COBEM 99
XV Congresso Brasileiro de Engenharia Mecânica
15th Brazilian Congress of Mechanical Engineering
22 - 26 de Novembro de 1999 / November 22 - 26, 1999 Águas de Lindóia, São Paulo,

A NEURAL NETWORK BASED CONTROL SCHEME FOR MULTIVARIABLE SYSTEMS

Marcos de Oliveira Fonseca, M.Sc.
ATAN Sistemas de Automação e Otimização.
Rua Pernambuco 353 16º Andar, 30.130-150 Belo Horizonte, MG, Brazil.
Phone: +55 31 261-8890. Fax +55 31 261-8900.
Email: oliveira@atan.com.br

José Maria Gálvez, Ph.D.
Universidade Federal de Minas Gerais, Departmento de Engenharia Mecânica.
Av. Antônio Carlos 6627, Pampulha, 31.270-901 Belo Horizonte, MG, Brazil.
Phone: +55 31 499-5236 Fax: +55 31 443-3783.
Email: jmgm@dedalus.lcc.ufmg.br.

ABSTRACT. *A major difficulty in multivariable control design is the cross-coupling between inputs and outputs which obscures the effects of a specific controller on the overall behavior of the system. This paper considers the application of neural networks in decoupling multivariable output feedback controllers. Simulation results are presented to show the feasibility of the proposed technique.*

Keywords: *Neural Networks, System Decoupling, Control Systems.*

1. INTRODUCTION

One of the main issues in the control design of multivariable systems is the cross-interaction between inputs and outputs. This interaction perturbs the effects of a specific loop controller on the corresponding output. And in some cases, it may cause system instability. Traditionally, frequency domain methods have been used to improve the system decoupling and to allow SISO linear control design approaches. In all these cases, a good knowledge of the plant dynamics is usually required to permit the design of a robust control system.

In general, MIMO frequency domain design techniques have to be performed in two steps. The first one consists of designing a precompensator to decouple the system and the second one corresponds to the controller design itself to achieve performance. Classical control techniques for SISO systems can be used in this second step since the system is already decoupled.

Several contributions can be found in the literature addressing the input-output cross-coupling in multivariable systems. It can be cited the papers of Bristol (1996), Wood and Berry (1973), Patel and Munro (1982), McAvoy (1983) and Deshpande (1989) among others. Also, several techniques have been proposed by the scientific community to address the robust performance and robust stability problems, among which can be cited the works of Athans (1966) in Linear Quadratic Optimal Control, Doyle (1982) in the concept of Structured Singular Values in the Frequency Domain and Francis (1987) in H-Infinity Optimal Control.

The techniques mentioned above are based upon solid theoretical concepts. In practice, however, the unidirectional and saturation characteristics of the power source as well as the plant nonlinear behavior may cause undesirable controller behavior. Also, model uncertainties may cause poor performance and even instability. Frequently, even the construction specifications work against the control performance. In some cases, optimizing the profile smoothness of the plant outputs requires increased interaction of the sub-plants that may cause the undesirable cross-coupling between inputs and outputs.

Neural Networks have been recently proposed as a solution for the control problem of some ill-conditioned processes. Some relevant contributions are the papers from Guez, Eilbert and Kam (1988); Hunt, Sbarbaro, Zbikowski and Gawthrop (1992); Khalid and Omatu (1992); Tai, Wangi and Ashenayi (1992); and Yamada and Yabuta (1993).

This paper presents an alternative approach to system decoupling and control: the Neural Network Dynamic Decoupling (NNDD) approach. From the control viewpoint, the NNDD scheme can be seen as an evolution of the direct control type schemes already presented in the literature. In this case, the NNDD scheme is basically a neural network structure whose inputs are properly chosen and which is trained to eliminate the undesirable input-output cross-coupling in multivariable plants.

2. NEURAL NETWORK DYNAMIC DECOUPLING

This section presents the Neural Network Dynamic Decoupling (NNDD) scheme. Without loss of generality, only second order transfer functions in 2x2 MIMO setups are considered. The two inputs are named u_1 and u_2, and the two outputs m_1 and m_2. Figure 1 shows a block diagram of the proposed scheme.

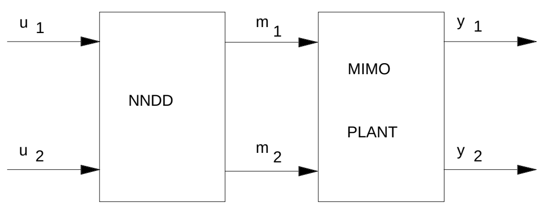

Figure 1. The NNDD scheme.

A general form for a second order discrete-time system is given by:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \qquad (1)$$

in this case

$$y(k) = b_0\, u(k) \ + \ v(k) \qquad (2)$$

where

$$v(k) = b_1 u(k-1) + b_2 u(k-2) - a_1 y(k-1) - a_2 y(k-2) \qquad (3)$$

Thus, to implement a neural-network-based dynamic model for this SISO system case, the network structure will have an input layer with n_i = 5 neurons, a hidden layer with some

n_h neurons and an output layer with $n_o = 1$ neurons. Figure 2 shows the network implementation for the general second order scalar case.

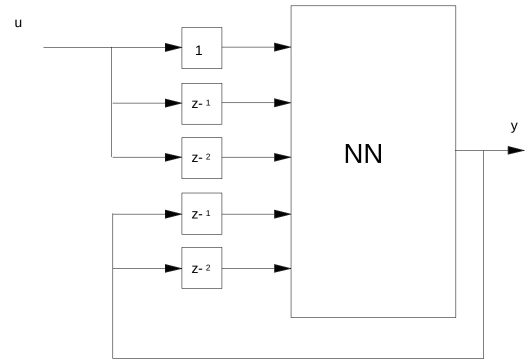

Figure 2. The NNDD SISO Structure.

In this case, the inputs and outputs of this NN structure are defined according to the current neural network operation mode, as it will be seen later. In the 2x2 MIMO case, the system matrix transfer function has the following form:

$$Y(z) = G(z)\ U(z) = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} U(z) \tag{4}$$

where

$$G_{ij}(z) = \frac{b_{ij0} + b_{ij1}z^{-1} + b_{ij2}z^{-2}}{1 + a_{ij1}z^{-1} + a_{ij2}z^{-2}} \tag{5}$$

Then, in this case the neural network structure can be built with an input layer with $n_i = 10$ neurons, a hidden layer with n_h neurons and an output layer with $n_o = 2$ neurons. Also, in this case two data vectors become available:

$$Y_k = \begin{bmatrix} y_1(k) \\ y_1(k-1) \\ y_1(k-2) \\ y_2(k) \\ y_2(k-1) \\ y_2(k-2) \end{bmatrix} \qquad U_k = \begin{bmatrix} u_1(k) \\ u_1(k-1) \\ u_1(k-2) \\ u_2(k) \\ u_2(k-1) \\ u_2(k-2) \end{bmatrix} \tag{6}$$

Clearly, any combination of data sets from Y_k and U_k can be properly selected to train the network as long as stability holds.

Two operation modes are generally considered: the training or learning mode and the real-time or controller mode.

In learning mode, the input information vector consists of the complete time history of the plant outputs and the prior plant inputs such that the following data vector and desired output vector are defined as

$$x_k = \begin{bmatrix} y_1(k) \\ y_1(k-1) \\ y_1(k-2) \\ y_2(k) \\ y_2(k-1) \\ y_2(k-2) \\ u_1(k-1) \\ u_1(k-2) \\ u_2(k-1) \\ u_2(k-2) \end{bmatrix} \qquad d_k = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{7}$$

then, the neural network output vector is given by:

$$m_k = \begin{bmatrix} m_1(k) \\ m_2(k) \end{bmatrix} = \begin{bmatrix} \tilde{u}_1(k) \\ \tilde{u}_2(k) \end{bmatrix} \tag{8}$$

finally, an error vector can now be defined as

$$e_k = d_k - m_k \tag{9}$$

Notice that the previous selection of data sets actually means that the neural network was trained to approximate, as accurate as possible to the plant inverse dynamics. In this operation mode, a recursive LMS algorithm was used to update the network weights. For the neural network training purposes the data sequences act as independent stationary numbers and can be presented to the network inputs with any time shift. This can be very useful when one is dealing with long pure delay time systems. In these cases the network algorithm can be taught to have a predictive behavior to compensate for the dead time. Figure 3 gives the NNDD block diagram for the MIMO second order transfer function structure in learning mode.

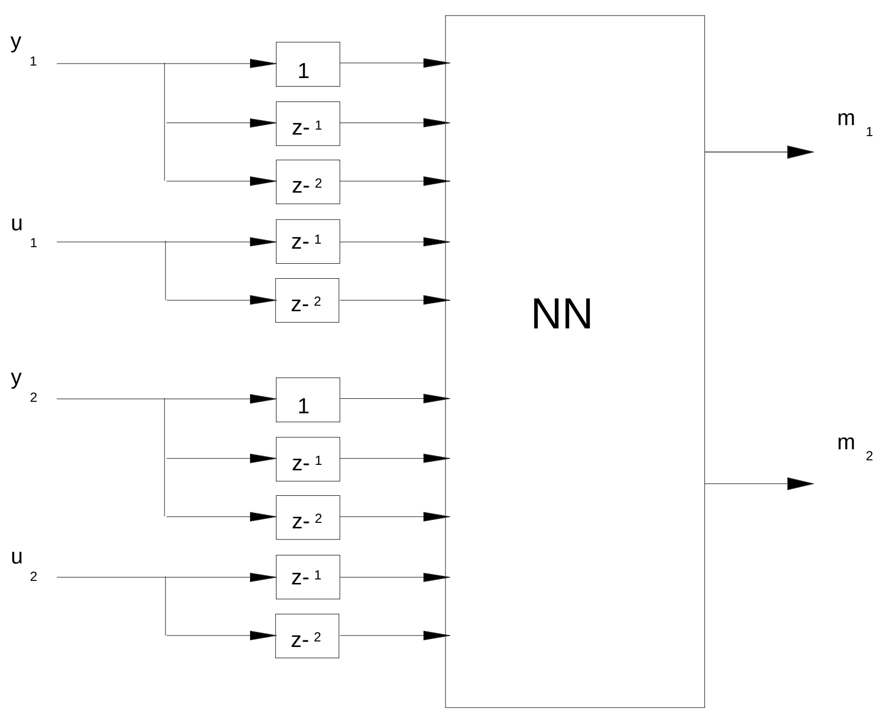

Figure 3. The NNDD Learning Mode.

In real-time mode the neural network input and output vectors are defined as

$$x_k = \begin{bmatrix} u_1(k) \\ u_1(k-1) \\ u_1(k-2) \\ u_2(k) \\ u_2(k-1) \\ u_2(k-2) \\ y_1(k-1) \\ y_1(k-2) \\ y_2(k-1) \\ y_2(k-2) \end{bmatrix} \qquad m_k = \begin{bmatrix} m_1(k) \\ m_2(k) \end{bmatrix} \tag{10}$$

In this case, the NNDD structure can be seen as a neural network implementation of IIR-type transfer functions. Thus, the proper choice of the network topology steps through a modeling problem and so depends on the dynamic characteristics of the plant. Particular cases might require different NNDD scheme configurations. In the case in which the plant model structure is known (including transfer function orders and time delay values) an improved NNDD design method can be applied as shown in Fonseca et al (1998).

Figure 4 presents the block diagram of the NNDD scheme in real time mode. Notice that the real-time NNDD implementation actually is a recursive network structure which includes feedback from the neural network outputs.

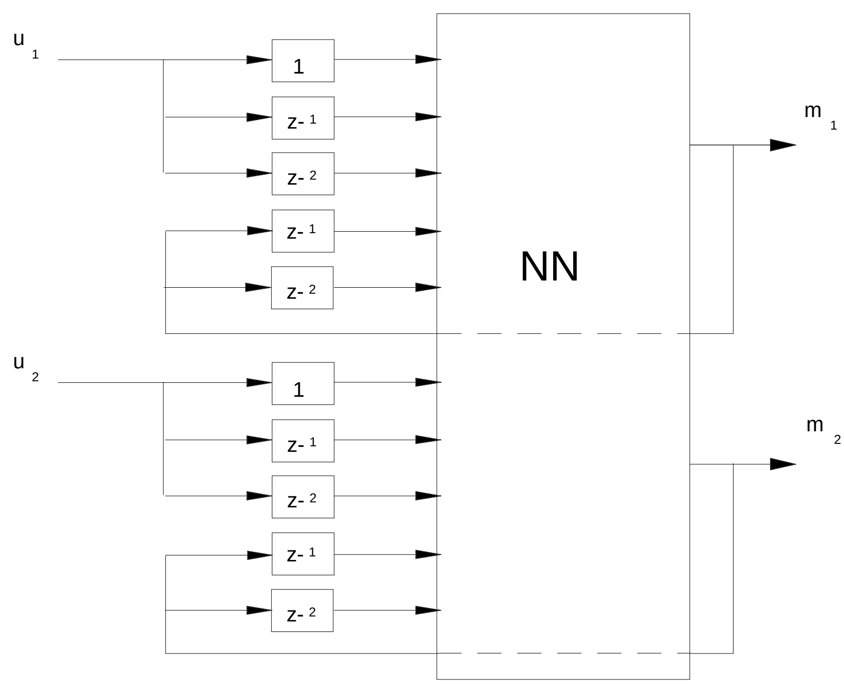

Figure 4. The NNDD Real Time Mode.

3. SIMULATION RESULTS

Figure 5 shows the complete setup for the NNDD scheme training.

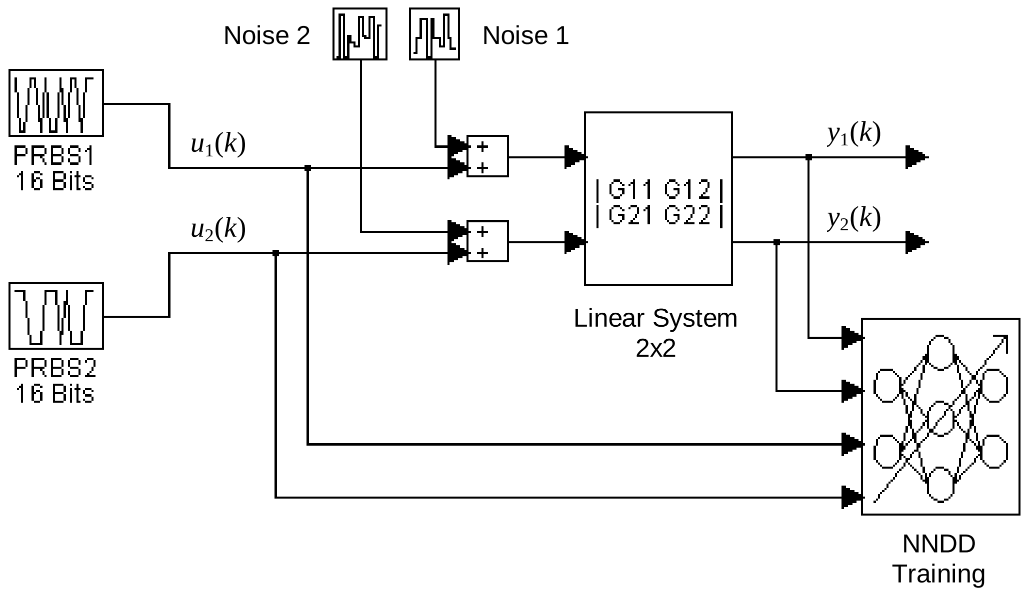

Figure 5. NNDD Training Scheme

To verify the decoupling performance of the proposed NNDD scheme, two experiments were set up (using 2x2 MIMO systems in both). In the first case the network topology was determined following a general approach without considering the system structure. In the second case, an improved method Fonseca et al (1998) was used to find the NNDD topology, the data vectors and the training technique.

<u>Case I:</u>

In this case, the neural network was trained using the standard gradient method. A linear activating function was used in the hidden and output layers. A pulse random binary signal (PRBS) was used to train the neural network (applied to the plant and network inputs). The sampling period was T=0.005 seconds and the algorithm took approximately 300.000 interactions to converge.

The plant was chosen as:

$$\begin{bmatrix} y_1(z) \\ y_2(z) \end{bmatrix} = \begin{bmatrix} \frac{0.6321}{z-0.3679} & \frac{-0.0787}{z-0.6065} \\ \frac{0.2855}{z-0.9048} & \frac{3.2445}{z-0.1889} \end{bmatrix} \begin{bmatrix} u_1(z) \\ u_2(z) \end{bmatrix} \qquad (11)$$

or

$$Y(z) = G(z)\ U(z) \qquad (12)$$

then, the inverse of such a matrix transfer function can be approximated (in NNDD format) by:

$$\tilde{G}^{-1}(z) = \begin{bmatrix} \dfrac{b_{110}+b_{111}z^{-1}+b_{112}z^{-2}}{1+a_{111}z^{-1}+a_{112}z^{-2}} & \dfrac{b_{120}+b_{121}z^{-1}+b_{122}z^{-2}}{1+a_{121}z^{-1}+a_{122}z^{-2}} \\ \dfrac{b_{210}+b_{211}z^{-1}+b_{212}z^{-2}}{1+a_{211}z^{-1}+a_{212}z^{-2}} & \dfrac{b_{220}+b_{221}z^{-1}+b_{222}z^{-2}}{1+a_{221}z^{-1}+a_{222}z^{-2}} \end{bmatrix}^{-1} \tag{13}$$

therefore, the compensated system will have a unit step response given by:

$$Y(z) = G(z)\ \tilde{G}(z)^{-1}\ U(z) \cong U(z) \tag{14}$$

This means that the plant will follow step inputs with an error which depends on the convergence of the neural network weights to their optimum values. The closer it gets to the optimum point the smaller the error is. Figures 6 and 7 compare the open loop system step responses with and without cross-coupling. Figures 8 and 9 show simulation results for the 2x2 MIMO system with NNDD.

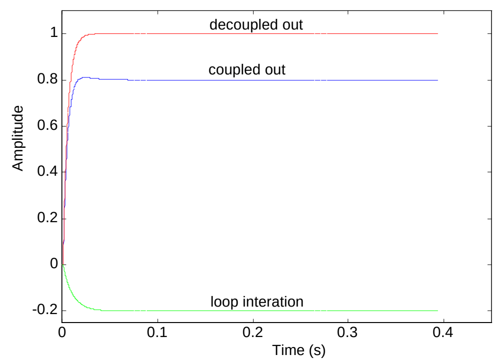

Figure 6. System Step Response - Output 1

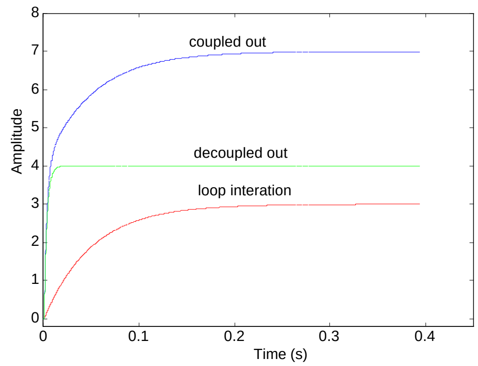

Figure 7. System Step Response - Output 2

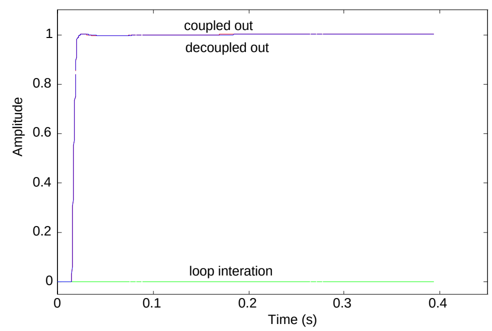

Figure 8. NNDD Step Response - Output 1.

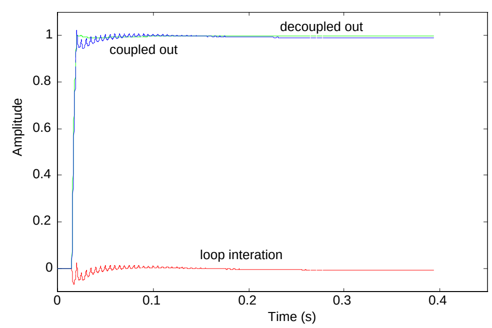

Figure 9. NNDD Step Response - Output 2.

One should observe that the NNDD scheme besides improving the system decoupling also manages to set the control signals m_1 and m_2 such that the plant follows the step inputs, in which case the obtained steady state errors for outputs 1 and 2 were 0.056% and 1.014 %, respectively.

Case II:

In this case, the plant was chosen to have different time delay values in each entry of the plant matrix transfer function such that:

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} z^{-2}\dfrac{0,744}{1-0,9419z^{-1}} & z^{-4}\dfrac{-0,8789}{1-0,9535z^{-1}} \\ z^{-8}\dfrac{0,5786}{1-0,9123z^{-1}} & z^{-4}\dfrac{-1,3015}{1-0,9329z^{-1}} \end{bmatrix} \cdot \begin{bmatrix} U_1(z) \\ U_2(z) \end{bmatrix} \tag{15}$$

The NNDD was designed following the procedure introduced in Fonseca et al (1998). The network structure consists in two Adaline neural networks, one for each input. The Adaline networks were trained using the recursive least mean square method. A PRBS signal was simultaneously applied to the plant and networks inputs. White noise was added to the PRBS signal to test the network rejection to uncorrelated information. The sampling period was chosen as T=15 min and the algorithm took approximately 1000 interactions to converge.

The network training results showed that the maximum convergence error (12.65 %) and error variance (0,0001374) were the same as the applied white noise signal which shows that the neural network truly rejected the uncorrelated noise signal, validating in this way the training procedure.

To compare the NNDD performance with the one of a more conventional control technique for MIMO systems, the Inverse Nyquist Array (INA) approach was chosen as the latter. In this case, the techniques were applied to the same 2x2 MIMO system. The NNDD was designed and implemented as previously shown. The MIMO controller was designed using the Inverse Nyquist Array method (INA). In both cases, a PID controller was designed and tuned for the decoupled system ($G_{12}(z) = G_{21}(z) = 0$). Finally, the coupled systems were simulated by applying a step signal to both inputs.

Figure 10 shows the plant responses for both techniques. It can be observed that the NNDD scheme has a faster response than the classical approach minimizing cross-coupling interaction. As a consequence of this the plant transient response with the NNDD controller is also better than the one with the INA-based MIMO controller. Also, it was observed during simulation that when using the NNDD technique, the PID controllers need to be tuned only once. On the other hand, with the INA-based controller it was necessary to adjust the PID controllers for different patterns of inputs, what makes this technique difficult to be applied to systems with strong interactions.

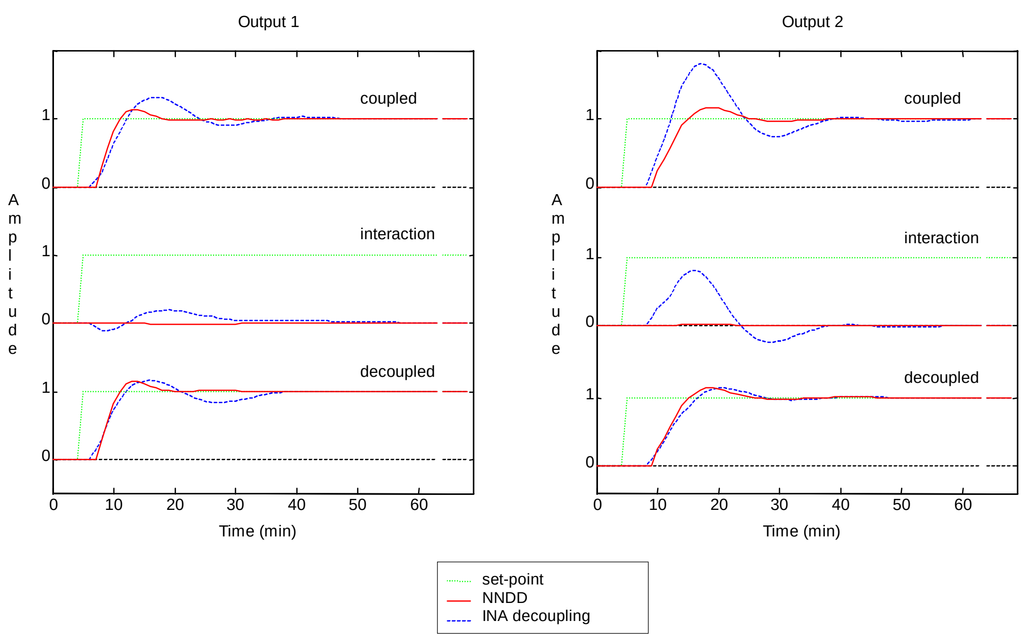

Figure 10. Comparison between NNDD and INA performances.

4. FINAL COMMENTS AND CONCLUSIONS

This work presented a technique based on neural networks to improve the dynamic decoupling of multivariable systems. Particularly, a neural-network-based scheme was applied to MIMO plants, achieving substantial cross-coupling reduction.

For the purpose of simplicity, this paper presented, in Case I, a simple 2x2 MIMO system, in which the neural network behaved in such a way that there was no need for an additional controller. Possibly, some more complex plant might not achieve such a good result. In any case, as showed in Case II, the NNDD approach will always improve the system decoupling making feasible the use of classical SISO control design techniques for MIMO systems.

It has been verified through simulation that one of the technique advantages is the reduced amount of computational effort expended in the design procedure. Besides that, the

NNDD technique application field is not limited to time invariant linear systems and does not require an accurate plant model. Also, in order to improve the robustness of the NNDD controller, the most relevant frequency range may be emphasized in the training stage, as addressed in Fonseca et al (1998).

In the case of non-minimal phase plants an unstable behavior was observed during the neural network training. The reason for that is easily explained if one sees the NNDD role as a computer implementation of inverse dynamics. Finally, as any other heuristic approach, additional research has to be done to fully test the NNDD capabilities.

5. BIBLIOGRAPHY

Bristol, E.H., 1966, On a New Measure of Interaction for Multivariable Process Control, IEEE Transactions on Automatic Control, Vol.11, pp. 133-134.

Wood, R. K. and Berry, M.W., 1973, Terminal Composition Control of a Binary Distillation Column, Chemical Engineering Science, vol. 28, pp. 1707-1717.

Patel, R. V. and Munro, N., 1982, Multivariable System Theory and Design, Pergamon Press.

McAvoy, T. J., 1983, Interaction Analysis - Principles and Applications, Instrument Society of America.

Deshpande, P. B., 1989, Multivariable Process Control, Instrument Society of America.

Athans, M. & Falb, P.L., 1966, Optimal Control, McGraw-Hill, New York.

Doyle, J.C., 1982, Analysis of feedback systems with structured uncertainties, Proceedings of the IEE, Part D, vol 129, pp 242-250.

Francis, B.A., 1987, A course in H^{∞}Control Theory. Lecture Notes in Control and Information Sciences, Berlin:Springer-Verlag.

Guez, A., Eilbert, J. L. and Kam, M., 1988, Neural Network Architeture for Control, IEEE Control Systems Magazine, vol. 8, pp. 22-25.

Widrow, B., & Lehr, M., 1990, 30 Years of Adaptive Neural Networks: Perceptron, Madaline and Back Propagation, Proceedings of the IEEE, Vol. 78, No. 9.

Hunt, K.J., Sbarbaro, D., Zbikowski, R., & Gawthrop, 1992, P.J., Neural Networks for Control, A Survey, Automatica, Vol. 23, No. 6, pp. 1083-1112.

Khalid, M and Omatu, 1992, S., A Neural Network Controller for a Temperature Control System, IEEE Control Systems, pp. 58-64.

Tai, H. M., Wangi, J. and Ashenayi, K., 1992, A Neural Network-Based Control System, IEEE Transactions on Industrial Electronics, vol. 39, no. 6, pp. 504-510.

Yamada, T., & Yabuta, T., 1993, Dynamic System Identification Using Neural Networks, IEEE Transactions on Systems, Management, and Cybernetics, Vol. 23, No.1.

Fonseca, M. O. and Galvez, J. M., 1998, Procedimento para Projeto de Desacopladores Dinâmicos utilizando Redes Neurais Artificiais, V Simpósio Brasileiro de Redes Neurais, Vol. 2, pp. 241-246.