

## Design Procedure for Neural Network Based Dynamics Decouplers

Marcos de Oliveira Fonseca, M.Sc.<sup>1</sup> & José Maria Gálvez, Ph.D.<sup>2</sup>

<sup>1</sup>ATAN Sistemas – Departamento de Pesquisa e Desenvolvimento  
R. Pernambuco 353 16<sup>o</sup> andar, 30.130-150, Belo Horizonte, MG, Brasil  
Fone: +55 31 3261-8880, Fax: +55 31 3261-8900

<sup>2</sup>Departamento de Engenharia Mecânica - Universidade Federal de Minas Gerais  
Av. Antônio Carlos 6627, 31.270-901, Belo Horizonte, MG, Brasil  
Fone: +55 31 3499-5236, Fax: +55 31 3443-3783  
E-mails: oliveira@atan.com.br, jmgm@dedalus.lcc.ufmg.br

### Abstract

*This work presents a design procedure for neural network based dynamics decouplers for multivariable systems. Theoretical aspects related to the determination of the inverse dynamics for multivariable linear systems are considered. The application of artificial neural networks in inverse dynamics implementation is discussed. Finally, a design example is presented.*

### 1. Introduction

The control of multivariable plants has traditionally been a challenging task for the control community. The difficulties are directly related to the existence of nonlinear plants, time delays, inaccurate models, measurement noise, cross-coupling interaction between inputs and outputs, etc. [1, 2].

Traditionally, SISO control techniques have failed in the presence of I/O cross coupling. Thus, the reduction of the I/O cross interaction plays an important role in the control design for MIMO systems. A well accepted technique is to design a dynamic uncoupling control structure that has as its main objective the reduction of the cross coupling interaction between inputs and outputs of the MIMO plant.

The use of dynamic decoupling structures allows the application of SISO multi-loop tuning techniques to solve the control problem of MIMO systems. This is achieved making the plant diagonal dominant by reducing the cross interaction among loops. Several contributions can be found in the literature based on this idea [9, 11]. Most of them are based on trial and error procedures.

Artificial Neural Networks (ANN's) have been the focus of the attention of the scientific community, especially for control applications [7]. This is a direct consequence of the ANN properties and characteristics:

- Capability of Nonlinear Mappings.
- Distributed Parallel Processing.
- Software or Hardware Implementation.

- Learning and Adaptation Capabilities.
- Quantitative and Qualitative Data Processing.
- MIMO Data Structures Processing.

Recently, ANN techniques have been combined with fuzzy logic algorithms delivering new neuro-fuzzy systems [8].

Several contributions can be found in the literature considering the use of ANN to implement direct and inverse dynamics for SISO systems. This work presents an extension of the SISO Yamada and Yabuta procedure [12] for MIMO systems. It focuses on the use of ANN in the implementation of dynamic decoupling structures for MIMO systems with time delays [2, 3, 4, and 5].

### 2. Direct Control & Dynamic Decoupling

Direct control schemes usually consist of a controller placed in the direct path of the system as shown in Figure 1.

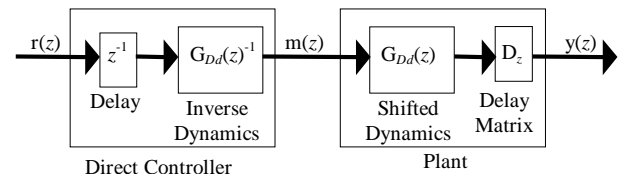


Figure 1. The Direct Controller Scheme.

In the ideal case, the controller cancels the plant dynamics such that the new controller-plant system behaves as a scalar and thus the plant output exactly follows the input. In this case, the system output is given by:

$$\mathbf{y}(z) = \mathbf{D}_z \mathbf{G}_{Dd}(z) \mathbf{G}_{Dd}(z)^{-1} z^{-1} \mathbf{r}(z) \quad (1a)$$

$$\mathbf{y}(z) = \mathbf{D}_z z^{-1} \mathbf{r}(z)$$

where

$$\mathbf{D}_z(z) = \begin{bmatrix} z^{-d11} & 0 & \cdot & 0 \\ 0 & z^{-d22} & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & z^{-dmm} \end{bmatrix} \quad (1b)$$

The time delay  $z^{-1}$  is due to the controller digital implementation. Also, it should be noted that the plant output,  $y(z)$ , corresponds to the controller input,  $r(z)$ , shifted by  $d_{ii}+1$  samplings. This is consistent with the fact that it is not possible to remove the common time delays from the plant dynamics.

In practical applications, however, this type of control technique must deal with difficult tasks such as unstable pole-zero cancellations, poor modeling, and the presence of measurement noise and output disturbances. Because of these, direct controllers are not implemented alone but in combination with some other type of control. In the MIMO case, a good model for the inverse dynamics of the plant allows the reduction of cross coupling interaction between the plant inputs and outputs improving the diagonal dominance in some frequency range and facilitating the implementation of a second controller that is designed for performance (Figure 2).

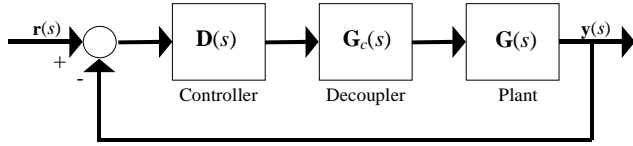


Figure 2. Multivariable Control.

### 3. Neural Network Dynamic Decoupling

This section presents a brief review on the procedure to obtain the inverse dynamics of MIMO linear systems. Without loss of generality only 2x2 MIMO linear systems will be considered [2].

Let a 2x2 linear system be defined by

$$[y(z)] = [G(z)] [u(z)] \quad (2a)$$

where

$$[G(z)] = \begin{bmatrix} z^{-p_{11}} \frac{A^{m_{11}}(z^{-1})}{B^{n_{11}}(z^{-1})} & z^{-p_{12}} \frac{E^{m_{12}}(z^{-1})}{F^{n_{12}}(z^{-1})} \\ z^{-p_{21}} \frac{C^{m_{21}}(z^{-1})}{D^{n_{21}}(z^{-1})} & z^{-p_{22}} \frac{G^{m_{22}}(z^{-1})}{H^{n_{22}}(z^{-1})} \end{bmatrix} \quad (2b)$$

the matrix transfer function  $[G(z)]$  can be factored as

$$[G(z)] = [D_z] [G_{Dd}(z)] \quad (3a)$$

where

$$[D_z] = \begin{bmatrix} z^{-d_{11}} & 0 \\ 0 & z^{-d_{22}} \end{bmatrix} \quad (3b)$$

$$[G_{Dd}(z)] = \begin{bmatrix} z^{q_{11}} \frac{A^{m_{11}}(z^{-1})}{B^{n_{11}}(z^{-1})} & z^{q_{12}} \frac{E^{m_{12}}(z^{-1})}{F^{n_{12}}(z^{-1})} \\ z^{q_{21}} \frac{C^{m_{21}}(z^{-1})}{D^{n_{21}}(z^{-1})} & z^{q_{22}} \frac{G^{m_{22}}(z^{-1})}{H^{n_{22}}(z^{-1})} \end{bmatrix} \quad (3c)$$

and

$d_{ii}$  = pure time delays

$q_{ij}$  = complementary time delays

The entries of matrix  $[D_z]$  are pure time delays that are larger or equal than smallest time delay among all time delays from any input to any output of the plant. Equation 3 is used to determine the common delay matrix  $[D_z]$ .

Since the delay matrix  $[D_z]$  does not have a causal inverse, finding the inverse dynamics of the plant basically consists in computing the inverse of the matrix  $[G_{Dd}(z)]$ . The necessary and sufficient condition for the existence of such an inverse matrix is that the matrix  $[G_{Dd}(z)]$  must have full rank over the field of the rational functions in  $z$  [6][10].

In the 2x2 MIMO case, the inverse of the shifted matrix  $[G_{Dd}(z)]$  is given by

$$G_c(z) = \frac{z^{-1} \cdot \begin{bmatrix} N_{c11}(z^{-1}) & N_{c12}(z^{-1}) \\ N_{c21}(z^{-1}) & N_{c22}(z^{-1}) \end{bmatrix}}{D_c(z^{-1})} \quad (4a)$$

where

$$\begin{aligned} N_{c11}(z^{-1}) &= z^{q_{22}} B^{n_{11}} D^{n_{21}} F^{n_{12}} G^{m_{22}}(z^{-1}) \\ N_{c12}(z^{-1}) &= -z^{q_{12}} B^{n_{11}} D^{n_{21}} E^{m_{12}} H^{n_{22}}(z^{-1}) \\ N_{c21}(z^{-1}) &= -z^{q_{21}} B^{n_{11}} C^{m_{21}} F^{n_{12}} H^{n_{22}}(z^{-1}) \\ N_{c22}(z^{-1}) &= z^{q_{11}} A^{m_{11}} D^{n_{21}} F^{n_{12}} H^{n_{22}}(z^{-1}) \\ D_c(z^{-1}) &= z^{(q_{11}+q_{22})} A^{m_{11}} D^{n_{21}} F^{n_{12}} G^{m_{22}}(z^{-1}) - \dots \\ &\quad - z^{(q_{12}+q_{21})} B^{n_{11}} C^{m_{21}} E^{m_{12}} H^{n_{22}}(z^{-1}) \end{aligned} \quad (4b)$$

The denominator,  $D_c(z^{-1})$ , in Equation 4a corresponds to the determinant of  $[G_{Dd}(z)]$  and its degree is given by

$$s = \max[(q_{11} + q_{22}), (q_{12} + q_{21})] \quad (5)$$

It can be shown that

$$G_c(z) = \frac{z^{-1} \cdot \begin{bmatrix} z^{-rd_{11}} \text{num}_{11}^{g_{11}}(z^{-1}) & z^{-rd_{12}} \text{num}_{12}^{g_{12}}(z^{-1}) \\ z^{-rd_{21}} \text{num}_{21}^{g_{21}}(z^{-1}) & z^{-rd_{22}} \text{num}_{22}^{g_{22}}(z^{-1}) \end{bmatrix}}{\text{den}^{gd}(z^{-1})} \quad (6)$$

where the  $rd_{ij}$  are time delays associated with  $r(z)$ .

The implementation of ANN based inverse dynamics requires the previous knowledge of input-output data as well as of the network structure. This information can be found solving the system:

$$[m(z)] = [G_c(z)] [r(z)] \quad (7a)$$

$$m_1(z) = \frac{1}{\text{den}_0} [z^{-(rd_{11}+1)} \text{num}_{11}^{g_{11}}(z^{-1}) r_1(z) - \dots - z^{-(rd_{12}+1)} \text{num}_{12}^{g_{12}}(z^{-1}) r_2(z) - \text{den}^{gd-1}(z^{-1}) m_1(z)] \quad (7b)$$

$$m_2(z) = \frac{1}{\text{den}_0} [-z^{-(rd_{21}+1)} \text{num}_{21}^{g_{21}}(z^{-1}) r_1(z) + \dots + z^{-(rd_{22}+1)} \text{num}_{22}^{g_{22}}(z^{-1}) r_2(z) - \text{den}^{gd-1}(z^{-1}) m_2(z)] \quad (7c)$$

From Equation 7 one has

$$m_1(z) = \mathbf{w}_1 \cdot \begin{bmatrix} z^{-(rd11+1)} r_1(z) \\ \vdots \\ z^{-(g11+rd11+1)} r_1(z) \\ z^{-(rd12+1)} r_2(z) \\ \vdots \\ z^{-(g12+rd12+1)} r_2(z) \\ z^{-1} m_1(z) \\ \vdots \\ z^{-gd} m_1(z) \end{bmatrix}, \quad m_2(z) = \mathbf{w}_2 \cdot \begin{bmatrix} z^{-(rd21+1)} r_1(z) \\ \vdots \\ z^{-(g21+rd21+1)} r_1(z) \\ z^{-(rd22+1)} r_2(z) \\ \vdots \\ z^{-(g22+rd22+1)} r_2(z) \\ z^{-1} m_2(z) \\ \vdots \\ z^{-gd} m_2(z) \end{bmatrix} \quad (8)$$

Vectors  $\mathbf{w}_1$  e  $\mathbf{w}_2$  are built from the coefficients of  $num_{11}$ ,  $num_{12}$ ,  $num_{21}$ ,  $num_{22}$  and  $den$  and dividing everything by  $den_0$ . It should be noted that using Equation 8, two linear ANN's can be used to obtain the  $m_1(k)$  and  $m_2(k)$  signals, as it is shown in Figure 3.

The NNBD stability can be assessed considering the monic polynomial built from the network weights related with  $m(z)$ . The monic polynomial roots correspond to the system transmission zeros. Figure 4 shows the Neural Network Based Dynamic Decoupler (NNBDD) control scheme for a 2x2 MIMO system.

Figure 5 shows the learning mode for the network structure. The training can be performed in on-line or off-line mode. The ANN structures can be trained one at a time. The system input signals  $u(k)$  are shifted and used as supervision signals  $g_1$  and  $g_2$ . Thus, Equation 9 gives the training vectors.

$$\mathbf{x}_1 = \begin{bmatrix} z^{-d11} y_1(z) \\ \vdots \\ z^{-(g11+d11)} y_1(z) \\ z^{-d12} y_2(z) \\ \vdots \\ z^{-(g12+d12)} y_2(z) \\ z^{-(dlx1+1)} u_1(z) \\ \vdots \\ z^{-(gd+dlx1)} u_1(z) \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} z^{-d21} y_1(z) \\ \vdots \\ z^{-(g21+d12)} y_1(z) \\ z^{-d22} y_2(z) \\ \vdots \\ z^{-(g22+d12)} y_2(z) \\ z^{-(dlx2+1)} u_2(z) \\ \vdots \\ z^{-(gd+dlx2)} u_2(z) \end{bmatrix} \quad (9)$$

#### 4. A 2x2 MIMO System Design Example

Determining the structures of the NNBD and the ANN's identifier (Figures 4 and 5, respectively) requires the previous knowledge of the ANN's input vectors and the supervision signals. For systems with stable inverse, this can be reached by:

I. Defining the plant model structure.

II. Finding the time delay  $d_c$  using:

$$d_c = \min[(p11 + p22), (p12 + p21)]$$

III. Determining the plant common time delays:

$$d11 = d_c - \min(p21, p22)$$

$$d22 = d_c - \min(p11, p12)$$

IV. Determining the complementary time delays:

$$q11 = d11 - p11$$

$$q12 = d11 - p12$$

$$q21 = d22 - p21$$

$$q22 = d22 - p22$$

V. Computing the value of  $s$  using:

$$s = \max[(q11 + q22), (q12 + q21)]$$

VI. Computing the polynomial degrees using:

$$g11 = n11 + n21 + n12 + m22$$

$$g12 = n11 + n21 + m12 + n22$$

$$g21 = n11 + m21 + n12 + n22$$

$$g22 = m11 + n21 + n12 + n22$$

$$gd = \max[(abs(-s + (q11 + q22)) + m11 + n21 + n12 + m22), (abs(-s + (q12 + q21)) + n11 + m21 + m12 + n22)]$$

VII. Computing the time delays of the reference:

$$rd11 = s - q22$$

$$rd12 = s - q12$$

$$rd21 = s - q21$$

$$rd22 = s - q11$$

VIII. Finding the dimensions of the input vectors:

$$\alpha = g11 + g12 + gd + 2$$

$$\beta = g21 + g22 + gd + 2$$

IX. Determining the time shift of the training vectors:

$$dlx1 = \max[(-rd11 + d11), (-rd12 + d22)]$$

$$dlx2 = \max[(-rd21 + d11), (-rd22 + d22)]$$

X. Finding the time delays of the training signals:

$$d11 = rd11 - d11 + dlx1$$

$$d12 = rd12 - d22 + dlx1$$

$$d21 = rd21 - d11 + dlx2$$

$$d22 = rd22 - d22 + dlx2$$

### 5. Experimental Results

For simulation purpose, the Wood and Berry model [11] with  $T_s = 1$  min was used (Equation 10).

$$\begin{bmatrix} X_D(z) \\ X_B(z) \end{bmatrix} = \begin{bmatrix} z^{-2} \frac{0,744}{1-0,9419z^{-1}} & z^{-4} \frac{-0,8789}{1-0,9535z^{-1}} \\ z^{-8} \frac{0,5786}{1-0,9123z^{-1}} & z^{-4} \frac{-1,3015}{1-0,9329z^{-1}} \end{bmatrix} \begin{bmatrix} R(z) \\ S(z) \end{bmatrix} \quad (10)$$

The NNBD performance was compared against a pre-compensator designed using the Inverse Nyquist Array (INA) technique.

The NNBD controller was designed following the procedure presented in the previous section. The network training was performed in simulation. A white noise signal was added to the plant input signals to evaluate the system behavior in the presence of interferences.

The INA pre-compensator was determined following the work presented by Deshpande [1]. In this

case, the pre-compensator matrix was designed to improve the diagonal dominance at the frequency of 0.6 rd/s resulting in a matrix  $[K]$  given by

$$K = \begin{bmatrix} 1 & 0.9 \\ -0.44 & 1 \end{bmatrix} \quad (11)$$

The closed loop included multi loop PID type controllers. The training block diagram is shown in Figure 6.

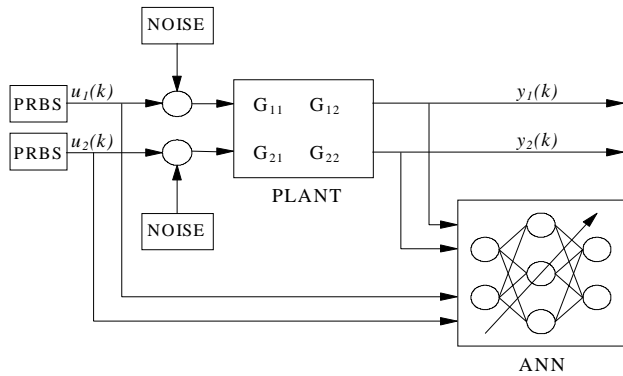


Figure 6. Training Setup.

Simulation was carried out using Matlab from Mathworks. The network weights converged after 1000 iterations to a maximum error of 12.65 % and variance of 0.0001374 both measured over the last 200 iterations. It was observed that the network error characteristics (mean and variance) are highly correlated with the white noise signal added to the system. It means that, the ANN's matched the system dynamics, canceling the system output and resulting the white noise as the main difference. Finally, the ANN's weights defined monic polynomials with stable roots that ensure the system stability. Table 1 shows the simulation data.

Table 1. NNBD Training Data.

<p>Sampling Period: 1 min</p> <p>Test Signal: PRBS</p> <p>Repeating Interval: 15 min</p> <p>Period (16 bits): 21845 h</p> <p>Amplitude: 0,25</p> <p>Interference Signal: white noise</p> <p>Variance: 0,000103</p> <p>Mean: -0,000215</p> <p>Maximum Absolute Value: 0,035 (14 %)</p> <p>Training Algorithm: "Recursive Least Square"</p> <p>Forgetting Factor = 1</p> <p>Weights Initial Values: Random [-1,1]</p>
---

The controllers were tuned using step inputs. The

tuning was performed separately for each loop. It was observed that the NNBD controller delivered an improved system decoupling and short settling time. In the PID-INA controller case, the controller required additional tuning to reach an acceptable performance. Figure 7 presents the simulation results.

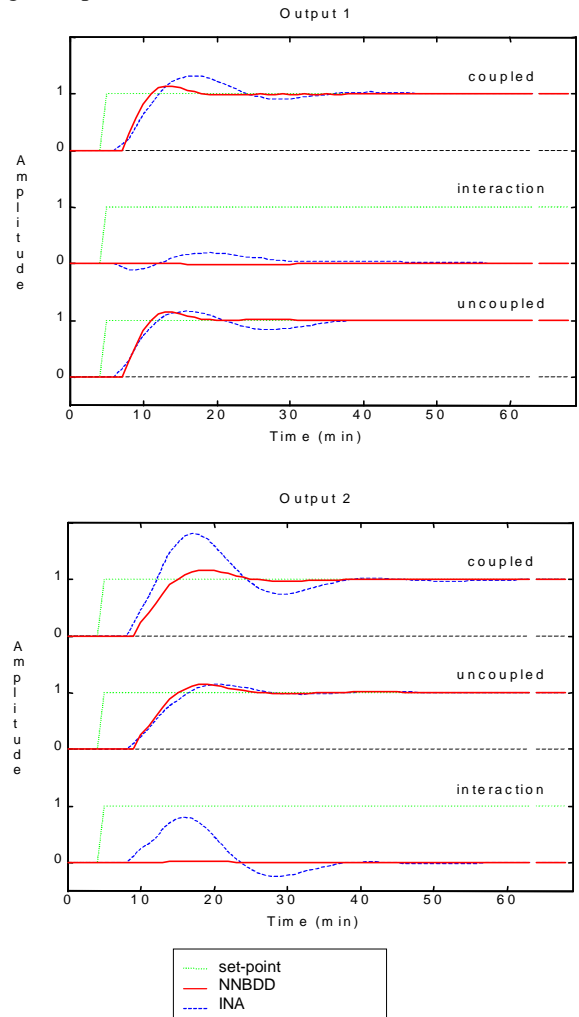


Figure 7. Comparison between NNBD and INA Multivariable Decoupling Control.

## 6. Final Comments and Conclusions

This work presented a design procedure of the NNBD applied to linear 2x2 MIMO systems. The proposed procedure simplifies the analysis allowing to address important control aspects such as stability and learning times. The simulation results showed the superior performance of the NNBD scheme over classical techniques such as Inverse Nyquist Arrays. The results presented here can be readily extended to higher order MIMO systems. Also the proposed control technique can be easily modified to perform in an adaptive scheme.

## 7. References

- [1] Deshpande, P. B. "Multivariable Process Control", Instrument Society of America, 1989.
- [2] Fonseca, M. O. "Controle Direto e Desacoplamento Dinâmico de Sistemas Utilizando Redes Neurais Artificiais", Dissertação de Mestrado, PPGEE-UFMG, 1995.
- [3] Galvez, J. M. and Fonseca, M. O. "Neural Network Based Decoupling Control for Multivariable Systems". Ieee-Icecs'96 - Third International Conference on Electronics Circuits and Systems, University of Patras, Rodos, Greece, 1996.
- [4] Galvez, J. M. and Fonseca, M. O. "Neural Network Precompensation for Multivariable System Decoupling". XI CBA/I SAA, Escola Politécnica da Universidade de São Paulo, São Paulo, Brazil, 1996.
- [5] Galvez, J. M. and Fonseca, M. O. "Multivariable System Decoupling through Neural Networks Precompensation", IEEE - MWSCS'96 - Midwest Symposium on Circuits and Systems, Ames-Iowa, USA, 1996.
- [6] Holt, B. R. and Morari, M. "Design of Resilient Processing Plants, Part V - The Effect of Dead Time on Dynamic Resilience", Chemical Engineering Science, vol. 40, no. 7, pp. 1229-1237, 1985.
- [7] Hunt, K. J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. J. "Neural Networks for Control Systems - A Survey", Automatica, vol. 28, no. 6, pp. 1083-1112, 1992.
- [8] Jang, J. S. R. et al "Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence", Prentice Hall, 1997.
- [9] McAvoy, T. J. "Interaction Analysis - Principles and Applications", Instrument Society of America, 1983.
- [10] Patel, R. V. and Munro, N. "Multivariable System Theory and Design", Pergamon Press, 1982.
- [11] Wood, R. K. and Berry, M.W. "Terminal composition control of a binary distillation column", Chemical Engineering Science, vol. 28, pp. 1707-1717, 1973.
- [12] Yamada, T. and Yabuta, T. "Dynamic System Identification Using Neural Networks", IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no.1, Jan./Feb. 1993.

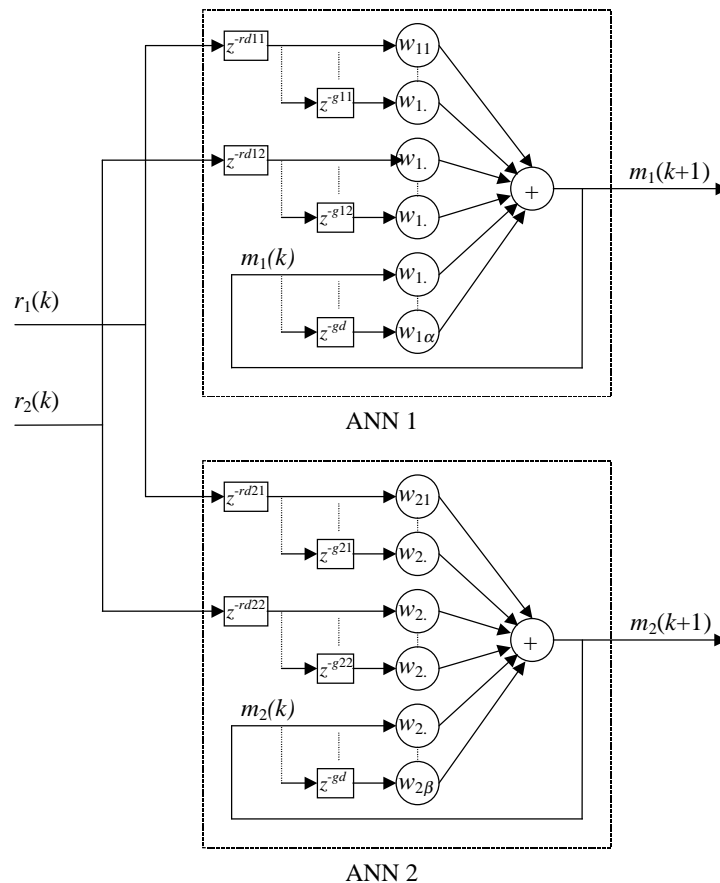


Figure 3. The NNBD Implementation for 2x2 MIMO Systems.

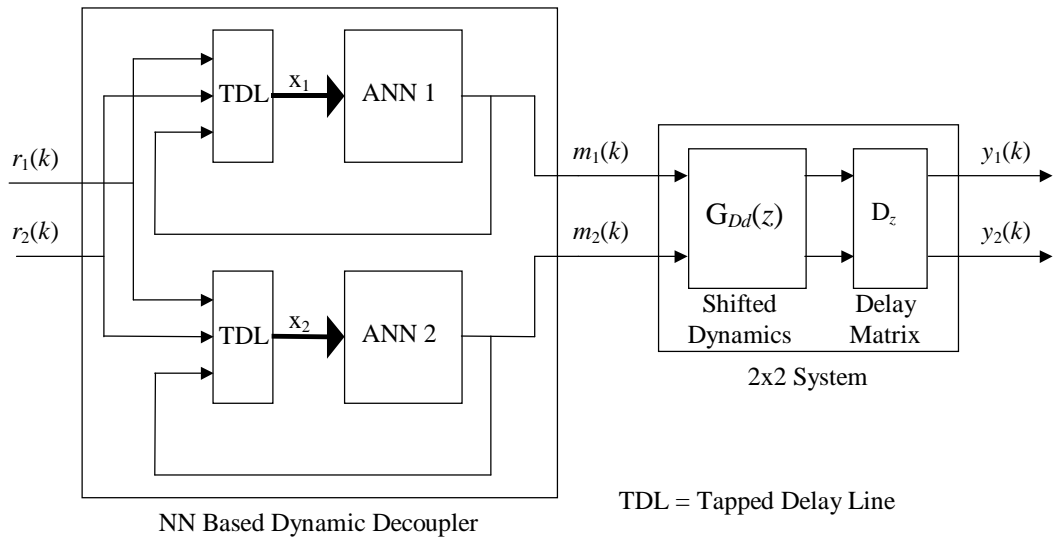


Figure 4. The NNBD Based Dynamic Decoupler for 2x2 MIMO Systems.

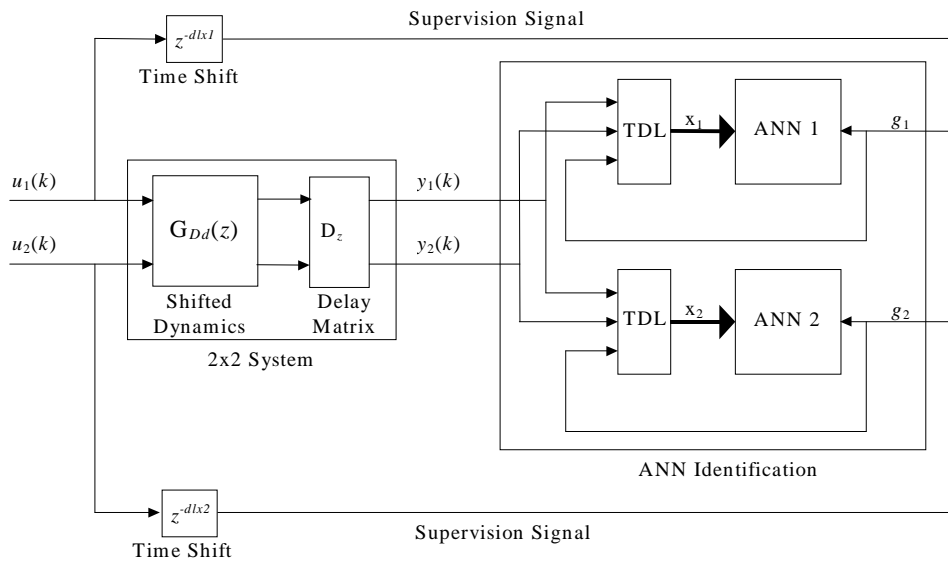


Figure 5. ANN Structure Learning Mode Scheme.