# STATISTICAL ANALYSIS OF ADAPTIVE NEURAL NETWORK INVERSION OF HAMMERSTEIN SYSTEMS FOR GAUSSIAN INPUTS

**Mohamed Ibnkahla**

**Electrical and Computer Engineering Department, Queen's University**
**Kingston, Ontario, K7L 3N6 Canada**
**E-mail: Mohamed.ibnkahla@ece.queensu.ca**
**Phone: (613) 533 3074 Fax: (613) 533 6615**

## ABSTRACT

The paper presents a statistical analysis of neural network (NN) inversion of Hammerstein systems. The system model is composed of a memoryless non linearity *g(.)* followed by a linear filter *H.* The inverse system is a nonlinear Wiener system consisting of an adaptive filter $Q$ followed by a memoryless perceptron. The adaptive filter $Q$ aims at inverting the linear part of the system (adaptive deconvolution). The perceptron aims at inverting the memoryless function (adaptive function inversion). The adaptive system is trained using the backpropagation algorithm (BP). The paper proposes recursions for the mean weight behavior during the learning process. The expression of the mean squared error (MSE) is given as function of the Hammerstein system parameters, the adaptive filter coefficients and the NN weights. The paper is supported with illustrations and computer simulations which show good agreement with theoretical analysis.

## 1 Introduction

Neural networks have been successfully applied to adaptive inversion of nonlinear systems (see e.g. [3, 4]). It is important then for the user to understand their learning behavior and estimate their performance.

This paper proposes a statistical analysis of NN inversion of Hammerstein systems (figures 1, 2). The unknown system to be inverted is composed of a memoryless non linearity *g(.)* followed by a linear filter *H*. This model can be found in a wide range of engineering problems such as satellite communications, microwave theory, adaptive control, biomedical applications, etc. Finding the inverse of the system (which is a nonlinear deconvolution problem) allows to extract the transmitted signal *x(n)* from the distorted output signal *y(n)*.

The input sequence *x(n)* is an independent gaussian sequence with zero mean and variance $\sigma_x^2$. The system output is corrupted by a zero mean gaussian noise $n_0(n)$ with variance $\sigma_0^2$. The system output at time n is expressed as: $y(n) = y_0(n) + n_0(n)$. $y_0(n)$ is the output in the noiseless case: $y_0(n) = H^t G(n)$, where $G(n) = [g(x(n))\ g(x(n-1)) \ldots\ g(x(n-N_H+1))]^t$, $N_H$ is the memory of the FIR filter *H.*

The inverse of the Hammerstein system (which is assumed to be invertible) is a Wiener system, i.e. a linear system followed by a memoryless nonlinearity (figure 2). In the ideal case, the inverse of the linear part is $1/H(z)$ and the inverse of the nonlinear part is $g^{-1}(x)$. In practical engineering problems these inverses are not known and have to be approximated.

In this paper, the inverse model is adaptively approximated by a linear FIR filter $Q = [q_1, q_2, \ldots, q_{N_q}]^t$, followed by a memoryless perceptron (figure 3). The adaptive filter $Q$ aims at inverting the linear part of the system (adaptive deconvolution), whereas the perceptron aims at inverting the memoryless function (adaptive function inversion).

The perceptron is composed of a scalar input (which is the output of the Hammerstein system *y(n)*), *M* neurons in the first layer, and a scalar output neuron. The NN output is expressed then as:

$$s(n) = \sum_{k=1}^{M} c_k f\left(w_k Q^t Y(n) + b_k\right),$$

where $Y(n) = [y(n)\ y(n-1) \ldots y(n-N_Q+1)]^t$, $\{w_k\}$ and $\{b_k\}$, *k=1,…,M,* are the weights and bias terms of the input layer neurons, respectively, and $\{c_k\}$, *k=1,…,M,* are the weights of the output neuron. $f$ is the first layer activation function.

The adaptive system is trained using the backpropagation algorithm (BP), which is a gradient descent algorithm that minimizes the error between the desired output (which is a known delayed transmitted sequence, $x(n-\Delta)$) and the adaptive system output *s(n)*: $e^2(n) = (x(n-\Delta) - s(n))^2$.

The adaptive system parameters are updated at each iteration *n* as follows, where $\mu$ is a small positive parameter:

$$\begin{aligned} w_i(n+1) &= w_i(n) - \mu \frac{\partial e^2(n)}{\partial w_i(n)} \\ &= w_i(n) + 2\mu e(n) c_i(n) Q^t(n) Y(n) f'\left(w_i(n) Q^t(n) Y(n) + b_i(n)\right) \end{aligned}$$

$$b_i(n+1)=b_i(n)-\mu\frac{\partial e^2(n)}{\partial b_i(n)}=$$

$$b_i(n)+2\mu e(n)c_i(n)f'\left(w_i(n)Q^t(n)Y(n)+b_i(n)\right),$$

$$c_i(n+1)=c_i(n)-\mu\frac{\partial e^2(n)}{\partial c_i(n)}$$

$$=c_i(n)+2\mu e(n)f\left(w_i(n)Q^t(n)Y(n)+b_i(n)\right),$$

$$Q(n+1)=Q(n)-\mu\, grad_{Q(n)}e^2(n)$$

$$=Q(n)+2\mu e(n)\sum_{k=1}^{M}c_k(n)w_k(n)f'\left(w_k(n)Q^t(n)Y(n)+b_k(n)\right)Y(n)$$

The analysis will be done for any functions *g(x)* and *f(x)*. For the illustrations, we will take the following specific functions: $g(x)=\int_0^x e^{-\frac{u^2}{2\sigma^2}}du$. This function is a reasonable model for saturation-type nonlinearities used in several applications. For *f(x)*, it will be taken as: $f(x)=erf(x)=\sqrt{\frac{2}{\pi}}\int_0^x \exp\left(-\frac{u^2}{2}\right)du$.

This is a sigmoid which has the same shape as other sigmoidal functions used in multi-layer neural networks.

The MSE surface and the mean weight transient behavior are studied is sections 2 and 3, respectively. Finally, computer simulations are presented in section 4 to support the theoretical results.

## 2 MSE surface and optimal Solution

### *2.1 Results of linear deconvolution of Hammerstein systems*

This section recalls the results of linear deconvolution of Hammerstein systems. This problem has been studied in [5] where a linear filter *Q* (alone), trained by the LMS algorithm, has been used for the adaptive deconvolution. In some applications, a linear filter is preferable to the nonlinear Wiener system, especially when the nonlinear distortions introduced by *g(.)* have small or minor effects on the received signal (e.g. in satellite communications when simple modulations schemes like BPSK are used [1]).

The results of [5] demonstrate that the adaptive filter converges to $Q_{optL}=R_y^{-1}P$. (Subscript *L* refers to the case where the adaptive system is composed of a linear filter *Q* only.) *P* is the correlation vector defined by: $P=E(x(n-\Delta)Y(n))$, and $R_y$ is the correlation matrix of vector *Y*.

It can be demonstrated that: $p_k=\left(\sigma_x^2/\sqrt{1+\frac{\sigma_x^2}{\sigma^2}}\right)h_{|k-\Delta|}$ and $R_y=\sigma_0^2 I+R_{y_0}$, where $R_{y_0}$ is the correlation matrix of $Y_0(n)$: $R_{y_0}(j,k)=H^tE(G(n-j)G^t(n-k))H$, which is fully determined by evaluating the correlation matrix of *G(n)*. For the specific function *g(x)* mentioned in the introduction, we have: $R_g=\sigma^2\sin^{-1}[\frac{\sigma_x^2}{\sigma^2+\sigma_x^2}]I$.

Note that, if we denote by $R_{y_0Idn}$ the correlation matrix of $Y_0(n)$ in the case where *g(.)* is the identity function, then $R_{y_0}=\frac{\sigma^2}{\sigma_x^2}\sin^{-1}[\frac{\sigma_x^2}{\sigma^2+\sigma_x^2}]R_{y_0Idn}$. This shows for examples that, in the noiseless case, the optimal deconvoluting filter of the NL system is a scaled version of the optimal deconvoluting filter of the linear system: $Q_{opt\,L}=\alpha Q_{opt\,Idn}$,

where $\alpha=\frac{\sigma_x^2}{\sigma^2}\frac{1}{\sin^{-1}[\frac{\sigma_x^2}{\sigma^2+\sigma_x^2}]}$.

### *2.2 MSE surface*

In what follows we consider the system of figure (3). We denote by $MSE_M$ the MSE when the NN has *M* neurons in the first layer. The MSE can be expressed as:

$$MSE_M=E(e^2(n))=\sigma_x^2+\sum_{k,j}c_kc_jF_{k,j}-2\sum_{k=1}^{M}c_kG_k$$

where $F_{k,j}=E\left[f\left(w_kQ^tY(n)+b_k\right)f\left(w_jQ^tY(n)+b_j\right)\right]$,

and $G_k=E[x(n-\Delta)f(w_kQ^tY(n)+b_k)]$.

For the special functions *f(x)* and *g(x)* mentioned in the introduction, G and F can be approximated (under the assumption of a soft nonlinearity *g(x)*, i.e. large $\sigma$ ) as:

$$F_{k,j}\approx\int_0^{\sigma_r^2}U(r)dr+E[erf(w_kz_2+b_k)]\times E[erf(w_jz_2+b_j)]$$

$$G_k\approx Q^tP\sqrt{\frac{2}{\pi}}\frac{w_k}{\sqrt{1+w_k^2\sigma_r^2}}\exp\left(\frac{b_k^2}{2}\left(\frac{1}{1+\zeta}-1\right)\right),$$

where:

$$U(r)=\frac{2}{\pi}\frac{w_kw_j}{\sqrt{1+w_k^2+\sigma_r^2w_j^2+(\sigma_r^4-r^2)w_k^2w_j^2}}\times$$

$$\exp[\frac{1}{2}[-b_k^2-b_j^2+\frac{1}{w_j^2+\frac{\sigma_r^2}{\sigma_r^2-r^2}}\left[b_j^2w_j^2+\frac{\left(b_kw_k\left(w_j^2+\frac{\sigma_r^2}{\sigma_r^2-r^2}\right)+b_jw_j\frac{r}{\sigma_r^2-r^2}\right)^2}{\left(w_k^2+\frac{\sigma_r^2}{\sigma_r^2-r^2}\right)\left(w_j^2+\frac{\sigma_r^2}{\sigma_r^2-r^2}\right)w_j^2-\frac{r^2}{\sigma_r^2-r^2}}\right]]]$$

$\sigma_r^2=E((Q^tY(n))^2)=Q^tR_yQ$, $\zeta=\frac{\sigma_x^2}{R}-\frac{(Q^tP)^2}{Rw_k^2\sigma_r^2}$, and

$R=\sigma_x^2Q^tQ\sigma_y^2-\left(Q^tP\right)^2$.

Note that in the biasless case (i.e. all the bias terms are set to 0), closed form expression can be obtained:

$$F_{k,j}=\frac{2}{\pi}\sin^{-1}\left(\frac{w_kw_j\sigma_r^2}{\sqrt{1+\sigma_r^2w_k^2+\sigma_r^2w_j^2+\sigma_r^4w_k^2w_j^2}}\right),$$

$$G_k=Q^tP\sqrt{\frac{2}{\pi}}\frac{w_k}{\sqrt{1+w_k^2\sigma_r^2}}$$

Therefore, in the biasless case we have:

$$MSE_M=\sigma_x^2-2Q^tP\sqrt{\frac{2}{\pi}}\sum_{k=1}^{M}\frac{c_kw_k}{\sqrt{1+w_k^2\sigma_r^2}}$$

$$+\frac{2}{p}\sum_{k,j} c_k c_j \sin^{-1}\left(\frac{w_k w_j s_r^2}{\sqrt{1+s_r^2 w_k^2 + s_r^2 w_j^2 + s_r^4 w_k^2 w_j^2}}\right)$$

### 2.2.1 Stationary points of the perceptron:

The stationary points are obtained by setting the gradient of the MSE surface to zero:

$$c_i = \frac{G_i}{F_{i,i}} - \sum_{k=1,k\neq i}^{M} c_k \frac{F_{k,i}}{F_{i,i}}$$

$$\frac{\partial G_i}{\partial w_i} - \sum_{k,k\neq i}^{M} c_k \frac{\partial F_{k,i}}{\partial w_i} - \frac{1}{2} c_i \frac{\partial F_{i,i}}{\partial w_i} = 0$$

$$\frac{\partial G_i}{\partial b_i} - \sum_{k,k\neq i}^{M} c_k \frac{\partial F_{k,i}}{\partial b_i} - \frac{1}{2} c_i \frac{\partial F_{i,i}}{\partial b_i} = 0$$

In the biasless case, closed form expressions of these equations can be obtained:

$$\frac{Q^t P}{\left(1+w_i^2 s_r^2\right)^{\frac{3}{2}}} - \sum_{k=1}^{M}\sqrt{\frac{2}{p}}\frac{c_k w_k s_r^2}{\left(1+s_r^2 w_i^2\right)\sqrt{1+s_r^2\left(w_k^2+w_i^2\right)}} = 0$$

$$Q^t P \frac{w_i}{\sqrt{1+w_i^2 s_r^2}} - \sqrt{\frac{2}{p}}\sum_{k=1}^{M} c_k \sin^{-1}\left(\frac{w_k w_i s_r^2}{\sqrt{1+s_r^2 w_k^2 + s_r^2 w_i^2 + s_r^4 w_k^2 w_i^2}}\right) = 0,$$

*i=1,…,M.*

These equations can be solved numerically in order to determine the optimal weights and the local minima.

### 2.2.2 Stationary points of the linear adaptive filter Q:

The optimal filter $Q_{opt}$ is obtained by setting to zero the gradient of the MSE surface with respect to $Q$. It can be demonstrated that $Q_{opt}$ is a scaled version of the filter obtained in the linear deconvolution case (section 2.1): $Q_{opt} = g\, Q_{optL}$, where $Q_{optL} = R_y^{-1} P$.

The scale factor is: $g = \dfrac{\sum_{k=1}^{M} c_k \frac{\partial G_k}{\partial r}}{\sum_{k,j} c_k c_j \frac{\partial F_{k,j}}{\partial s_r^2} - 2\sum_{k=1}^{M} c_k \frac{\partial G_k}{\partial s_r^2}}$, where $r = Q^t P$. (Note that factor $g$ can be easily determined in a closed form in the biasless case by calculating these derivatives.)

This important result can be exploited in order to improve the algorithm convergence speed and reduce the computational time, by splitting our problem into two parts:

**1- Linear deconvolution**: The adaptive system is composed of a linear filter $Q$ trained with the LMS algorithm.

**2- Nonlinear memoryless function inversion**: We start phase 2 when the filter of phase 1 converges. We freeze the converged filter, and add a memoryless perceptron. Only the perceptron is trained. Note that factor $g$ can be seen as a normalization factor.

Note also that, in the case of a simultaneous adaptation of filter $Q$ and the perceptron, and if the latter is initialized with small values, then filter $Q$ will be the first to converge (i.e. it converges rapidly to a scaled version of $Q_{optL}$), the perceptron weights will take much longer time to converge.

## 3 Mean weight behavior

The mean weight transient behavior is determined by the calculation of the expectations of the updating equations. We take the following notations: $E(w_k(n)) = \overline{w}_k(n)$, $E(b_k(n)) = \overline{b}_k(n)$, $E(c_k(n)) = \overline{c}_k(n)$, $E(Q(n)) = \overline{Q}(n)$. By taking the expectations of both sides of the updating equations of section 1, and assuming that the updating rate $m$ is small, we obtain the following mean weight recursions:

$$\overline{w}_i(n+1) = \overline{w}_i(n) + 2m\overline{c}_i(n)\left[\frac{\partial G_i}{\partial \overline{w}_i(n)} - \sum_{k=1,k\neq i}^{M} \overline{c}_k(n)\frac{\partial F_{i,k}}{\partial \overline{w}_i(n)} - \frac{1}{2}\overline{c}_i(n)\frac{\partial F_{i,i}}{\partial \overline{w}_i(n)}\right].$$

$$\overline{b}_i(n+1) = \overline{b}_i(n) + 2m\overline{c}_i(n)\left[\frac{\partial G_i}{\partial \overline{b}_i(n)} - \sum_{k=1,k\neq i}^{M} \overline{c}_k(n)\frac{\partial F_{i,k}}{\partial \overline{b}_i(n)} - \frac{1}{2}\overline{c}_i(n)\frac{\partial F_{i,i}}{\partial \overline{b}_i(n)}\right]$$

$$\overline{c}_i(n+1) = \overline{c}_i(n) + 2m\left[G_i - \sum_{k=1}^{M}\overline{c}_k(n)F_{i,k}\right].$$

$$\overline{Q}(n+1) = \overline{Q}(n) + 2m\left[\sum_{k=1}^{M}\overline{c}_k\frac{\partial G_k}{\partial s_r^2}P - 2\left(\sum_{k\neq j}\overline{c}_k\overline{c}_j\frac{\partial F_{k,j}}{\partial s_r^2} + \frac{1}{2}\sum_{k=1}^{M}\overline{c}_k^2\frac{\partial F_{k,k}}{\partial s_r^2} + \sum_{k=1}^{M}\overline{c}_k\frac{\partial G_k}{\partial s_r^2}\right)R_y\overline{Q}(n)\right]$$

In the biasless case, closed form expressions of these recursions can be obtained (note that all the weights in the RHS of the equations are at time $n$):

$$\overline{w}_i(n+1) = \overline{w}_i(n) + 2m\overline{c}_i\left[\sqrt{\frac{2}{p}}\frac{\overline{Q}^t P}{\left(1+\overline{w}_i^2 s_r^2\right)^{\frac{3}{2}}} - \sum_{k=1}^{M}\frac{2}{p}\frac{\overline{c}_k\overline{w}_k s_r^2}{\left(1+\overline{w}_i^2 s_r^2\right)\sqrt{1+s_r^2\left(\overline{w}_k^2+\overline{w}_i^2\right)}}\right]$$

$$\overline{c}_i(n+1) = \overline{c}_i(n) + 2mQ^t P\sqrt{\frac{2}{p}}\frac{\overline{w}_i}{\sqrt{1+\overline{w}_i^2 s_r^2}}$$

$$-\sum_{k=1}^{M}\overline{c}_k\frac{2}{p}\sin^{-1}\left(\frac{\overline{w}_k\overline{w}_i s_r^2}{\sqrt{1+s_r^2\overline{w}_k^2 + s_r^2\overline{w}_i^2 + s_r^4\overline{w}_k^2\overline{w}_i^2}}\right)]$$

$$\overline{Q}(n+1) = \overline{Q}(n) + 2m\left[-\sqrt{\frac{2}{p}}\overline{Q}^t P\sum_{k=1}^{M}\frac{\overline{c}_k\overline{w}_k}{\sqrt{1+\overline{w}_k^2 s_r^2}}\;P\right.$$

$$-2\left(\frac{2}{p}\sum_{k\neq j}\frac{\overline{c}_k\overline{c}_j\overline{w}_k\overline{w}_j(1+0.5s_r^2(\overline{w}_k^2+\overline{w}_j^2))}{\sqrt{1+s_r^2(\overline{w}_k^2+\overline{w}_j^2)}(1+s_r^2(\overline{w}_k^2+\overline{w}_j^2)+s_r^4\overline{w}_k^2\overline{w}_j^2)}\right.$$

$$\left.\left.+\sum_{k=1}^{M}\frac{\overline{c}_k\overline{w}_k^2}{1+s_r^2\overline{w}_k^2}\left(\frac{\overline{c}_k}{p\sqrt{1+2s_r^2\overline{w}_k^2}} - \frac{\overline{Q}^t P}{\sqrt{2p}\sqrt{1+s_r^2\overline{w}_k^2}}\right)\right)R_y\overline{Q}\right].$$

(Note that all variables in the RHS of the equations are at time $n$.)

## 4 Application and Simulation Examples

The filter-NN structure has been applied for the inversion of a Hammerstein system. The following parameters have been taken: $H(z) = 1 + 0.5z^{-1}$, $s^2 = 2$, $s_x^2 = 1$, $s_0^2 = 0.0025$, $m = 0.0005$, and $\Delta = 3$. We have implemented a 5-tap filter, followed by a 3 neuron biasless perceptron. The learning curve (MSE versus time), and the mean weight transient behavior have been estimated over 100 Monte Carlo (MC) runs and compared to the theoretical expressions (figures 4 , 5 and 6). The simulations show good fit between the theory and MC estimations.

## 5 Conclusion

The paper presented a statistical analysis of neural network inversion of Hammerstein systems. The backpropagation algorithm was used for the learning process.

The paper studied the MSE surface of the adaptive system and proposed recursions for the mean weight behavior. Computer simulations showed good agreement between MC estimations and theoretical analysis.

## 6 References:


[1] S. Benedetto and E. Biglieri, *Digital Transmission With Wireless Applications*, Kluwer Academic Publishers, 1999.

[2] N. Bershad, J. Shynk, and P. Feintuch, "Statistical analysis of the single layer backpropagation algorithm: Part I", IEEE Trans. Signal processing, Vol. 41, pp. 553-582, 1993.

[3] S. Haykin, *Neural Networks: A Comprehensive Foundation*, IEEE Press, 1994.

[4] M. Ibnkahla, "Applications of neural networks to digital communications: A survey", Signal Processing, Vol. 80, pp. 1185-1215, July 2000.

[5] M. Ibnkahla, "Convergence properties of the LMS adaptive equalizer applied to nonlinear channels", in Proc. Wireless'01 Conference, Calgary, Canada, July 2001.



## Acknowledgement:

This work has been supported by the Ontario Ministry of Energy, Science, and Technology under the PREA program.


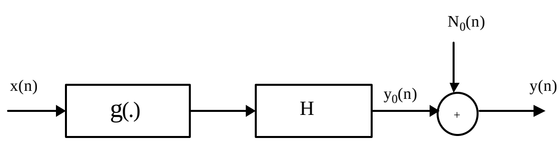


Figure 1: Hammerstein model

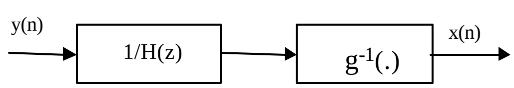


Figure 2: Ideal inversion in the noiseless case.

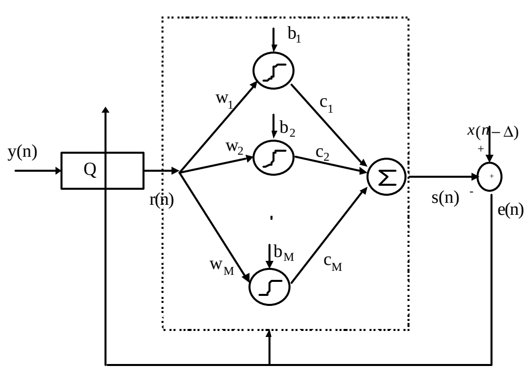


Figure 3: Neural Network structure.

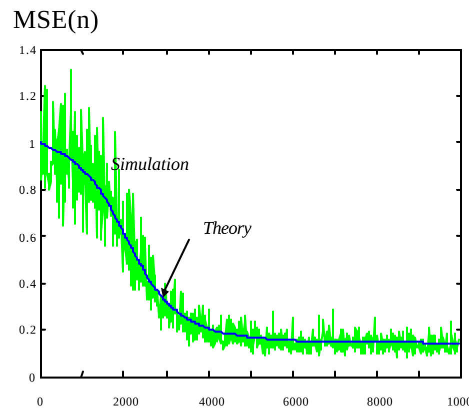


Figure 4: Learning curve (theory and simulation).

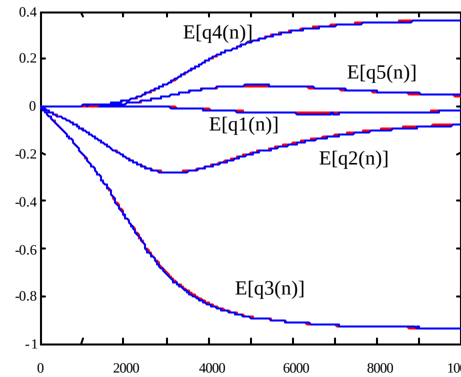


Figure 5: Filter $Q$ mean weight transient behavior, theory and MC estimations (the curves obtained by theory and MC estimations are superimposed).

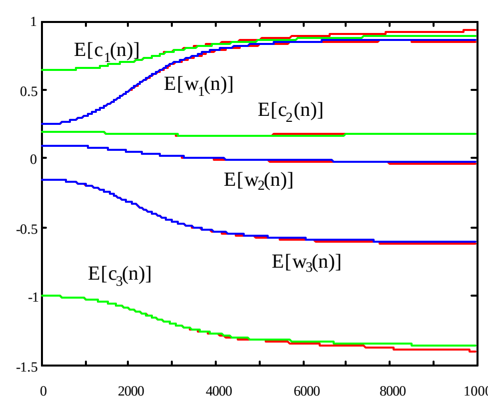


Figure 6: Neural network weights mean transient behavior, theory and MC estimations (the curves obtained by theory and MC estimations are very close).