



HAL
open science

Identification of non linear fractional systems using continuous time neural networks

François Benoit-Marand, Laurent Signac, Thierry Pointot, Jean-Claude
Trigeassou

► **To cite this version:**

François Benoit-Marand, Laurent Signac, Thierry Pointot, Jean-Claude Trigeassou. Identification of non linear fractional systems using continuous time neural networks. IFAC Workshop on Fractional Differentiation and its Applications, Jul 2006, Porto, Portugal. vol2 part1. hal-00211783

HAL Id: hal-00211783

<https://hal.science/hal-00211783>

Submitted on 21 Jan 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

IDENTIFICATION OF NON LINEAR FRACTIONAL SYSTEMS USING CONTINUOUS TIME NEURAL NETWORKS

François Benoît-Marand* Laurent Signac*
Thierry Poinot* Jean-Claude Trigeassou*

* *Laboratoire d'Automatique et d'Informatique Industrielle
Bâtiment de Mécanique 40, avenue du recteur Pineau
86022 Poitiers, France*

Abstract: Fractional systems allow us to model the processes governed by a diffusive equation. The simulation of such processes can be realized by using a non integer integrator. The aim of this paper is to estimate, in the time domain, both the non integer derivative order and the physical law of a non linear fractional system. To achieve this goal, we use Continuous Time Neural Networks. CTNN are dynamical neural structures that differ from the classical recurrent neural networks on the use of integrator blocks rather than delay blocks. This difference allows us to access the physical law of the system rather than only having a black box model. A non-integer CTNN is composed of a neural network and a non integer integrator. The identification stage –which consists in finding the good parameters for the neural network and for the integrator block– will be performed by using an output error identification. At the end of the procedure, a model reduction stage can be performed in order to revert from the neural network to a more realistic expression of the physical law of the process. To illustrate the method we'll give some simulation results.

Keywords: Neural networks, Non integer integrator, Non linear fractional systems, Output error identification

1. INTRODUCTION

Fractional systems are used to model the physical processes governed by a diffusive equation and have already been used in many different fields like thermics (Battaglia, 2002; Cois, 2002), electro-chemistry (Lin *et al.*, 2000a) or electromagnetism (Lin *et al.*, 2000b; Retière and Ivanès, 1998). The works in progress at the LAII* are based on the definition of a non integer integrator (Lin, 2001) that allows us to simulate linear and non linear fractional systems (Poinot and Trigeassou, 2004). In this paper, we will focus on the modeling and the identification of non linear fractional systems by using neural networks.

An original method using neural networks has already been developed in the case of non linear integer systems (Benoît-Marand *et al.*, 2004). This method, referred to as CTNN, combines a neural network with an integer integrator block. Contrary to the classical recurrent neural networks, that use delay blocks and only give a black box model of the process, CTNN, by separating the non linear static part of the process (the neural network) from the dynamical one (the integrator block), allows us to access the physical law of the process.

In this article, we will show how, by using a non integer integrator in the CTNN structure,

it is possible to extend the previous method to fractional systems. During the identification stage, we will not only have to estimate the parameters of the network which characterize the system's law but to estimate the derivative order of the system too. These parameters will be found by using an output error identification technique with temporal data.

First, we point out the definition of the non integer integrator and its state space representation. Then we show the principles of CTNN and their extension to fractional systems. Then, the identification stage is exposed. Finally, we illustrate our method by simulation results.

2. FRACTIONAL SYSTEMS AND NON INTEGER INTEGRATOR

2.1 Simulation of fractional systems

Let us consider the following integer differential equation ($u(t)$ and $y(t)$ are the input and the output of the system):

$$\frac{dy(t)}{dt} = f(u(t), y(t)) \quad (1)$$

This differential equation, generally non-linear, may be integrated with the help of an operator $I(s) = \frac{1}{s}$, which performs time integration of $\frac{dy(t)}{dt}$ (in order to give $y(t)$ at its output). Practically, we use a numerical integrator (Euler, Runge-Kutta 4,...) instead of an analog one ($\frac{1}{s}$).

Then, let us consider the fractional differential equation (n is the non integer derivative order):

$$\frac{d^n y(t)}{dt^n} = f(u(t), y(t)) \quad (2)$$

Assume that a fractional integrator $I_n(s) = \frac{1}{s^n}$ is available (its input is equal to $\frac{d^n y(t)}{dt^n}$ so that its output is equal to $y(t)$).

Thus, $I_n(s)$ generalizes to fractional systems the previously presented simulation principle. As a conclusion, integration of fractional systems simulation essentially relies on the definition of a fractional integrator, which will be a generalization of the conventional one.

Notice that this technique applies either to linear or to non-linear fractional systems, which is the main interest of the non-integer operator.

2.2 Definition of the non integer integrator

The ideal fractional integrator is defined by using the transfer function $I_n(s) = \frac{1}{s^n}$ which corresponds to a magnitude's slope equal to $-n \times$

20 dB/decade and to a phase equal to $-n \times 90^\circ$ for any frequency.

This theoretical frequency response can be approximated on a limited band $[\omega_b, \omega_h]$ (according to the desired accuracy) using cascade phase lead filters (Oustaloup, 1995), associated with an $1/s$ integrator (Pointot and Trigeassou, 2003).

Consider the following approximation of the theoretical non integer integrator (Lin, 2001) which corresponds to the Bode diagram on figure 1:

$$\hat{I}_n(s) = \frac{G}{s} \prod_{i=1}^N \frac{1 + \frac{s}{\omega'_i}}{1 + \frac{s}{\omega_i}} \quad (3)$$

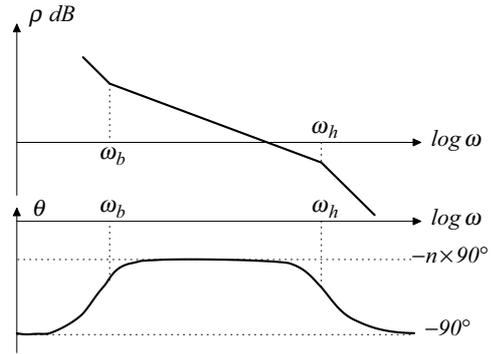


Fig. 1. Bode diagram of $\hat{I}_n(s)$

Its frequency response is a suitable approximation of the theoretical one :

- The intermediate zone ($\omega_b < \omega < \omega_h$) corresponds to the non integer behavior of the integrator, characterized by n .
- For low and high frequencies, the behavior of the integrator is conventional, with $n = 1$.

Finally, given N, n, ω_b, ω_h , the operator introduced by (3) is completely defined by the following relations¹ :

$$\begin{aligned} \omega'_1 &= \omega_b, & \omega_N &= \omega_h, & \omega_u &= \sqrt{\omega_b \omega_h}, \\ \omega_i &= \alpha \omega'_i, & \omega'_{i+1} &= \eta \omega_i, & n &= 1 - \frac{\ln \alpha}{\ln \eta} \end{aligned} \quad (4)$$

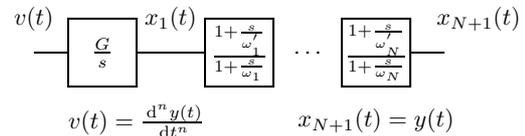


Fig. 2. $\hat{I}_n(s)$ block diagram

Using the block diagram shown in figure 2, $\hat{I}_n(s)$ can be described by the following state-space representation (Pointot and Trigeassou, 2003):

$$\dot{\underline{x}} = A^* \underline{x} + \underline{B}^* v \quad (5)$$

¹ In (3) G is defined so that the magnitude of $\hat{I}_n(s)$ is equal to the magnitude of $(\frac{1}{s})^n$ for the frequency ω_u .

where $A^* = M^{-1}A$, $B^* = M^{-1}B$, $B^T = [G_n \ 0 \ \dots \ 0]$,

$$M = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -\alpha & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & -\alpha & 1 \end{bmatrix}, A = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \omega_1 & -\omega_1 & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \omega_N & -\omega_N \end{bmatrix}$$

3. CONTINUOUS TIME NEURAL NETWORKS

There are two main approaches to simulate dynamical systems with a neural network. The first one consists in using dynamical units (*continuous time neurons*). Many models have been proposed (Hopfield, 1982; Pearlmutter, 1995; matoros *et al.*, 1995) and complex processes have been simulated or controlled by such networks. Nevertheless, the physical law of the process can not be extracted because dynamical and static components of the model are not separated.

The second approach is based on the use of a static net associated with an external dynamical component. Delay blocks (figure 3) may be used (Narendra and Levin, 1997) in order to obtain a discrete representation of the system: $\hat{y}_{k+1} = g(u_k, \hat{y}_k)$ where g is the function computed by the net.

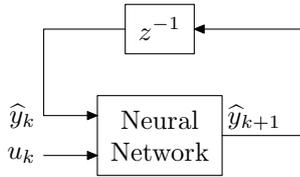


Fig. 3. Discrete time neural network

Such models have been successfully investigated in process control (Narendra, 1992) but don't give direct access to the physical law of the system². An enhanced design has been proposed (Wang and Lin, March 1998) to obtain a discretization of the process using the Runge-Kutta 4 (RK4) scheme: the net approaches the physical law of the system with a high accuracy. Nevertheless, by using a discrete representation of the process, the whole neural structure and the identification procedure have to be adapted according to the desired accuracy (Euler, RK4) or to the desired task.

We proposed a method (Benoît-Marand *et al.*, 2004; Benoît-Marand *et al.*, 2005), referred to as CTNN: we use a static feed-forward neural network followed by a continuous integration method (figure 4). Our model is directly related with

² Suppose $\frac{dy(t)}{dt} = f(u(t), y(t)) = au(t) + by(t)$, then the function computed by the static net is not equal to f but to $e^{aT}x(t) + \frac{b}{a}(1 - e^{-aT})u(t)$, and in the non linear case we do not have such relation to revert to the physical law.

the continuous equation of the system: $\frac{dy(t)}{dt} = g(u(t), \hat{y}(t))$ where g is the function computed by the net. Obviously, we can separate the static part of the process (the function computed by the net) from the dynamical one (Benoît-Marand *et al.*, 2005). By definition the net gives us direct access to the physical law of the simulated process. The idea is simple but powerful. Indeed, we need to adapt neither the net structure nor the principles of the identification procedure if we change the integration method. In particular, the integrator may be a fractional integrator.

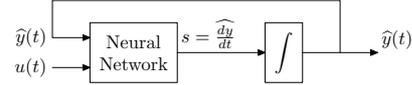


Fig. 4. A Continuous Time Neural Network CTNN

4. OUTPUT ERROR IDENTIFICATION

4.1 Presentation

In the following, we will consider a CTNN dedicated to single input single output fractional systems. We use the single hidden layer static net described figure 5 followed by a non integer integrator (figure 6). Adjustable parameters of the model, called $\underline{\theta}$, are the $4N_c$ parameters of the net which characterize the physical law and parameter α characterizing the derivative order of the fractional integrator.

$$\underline{\theta}^T = [w_{1,1} \dots w_{3,N_c} \quad b_1 \dots b_{N_c} \quad \alpha]$$

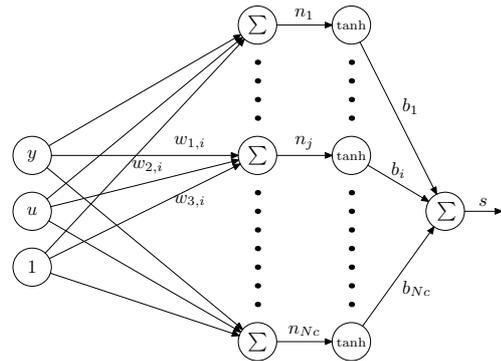


Fig. 5. The static neural network used in the CTNN

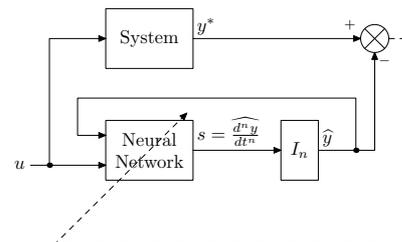


Fig. 6. Output error identification

The aim of the identification stage is to find $\underline{\theta}^*$ which minimizes the following criterion³:

$$J = \sum_{k=1}^K (y_k^* - \hat{y}_k)^2 = \sum_{k=1}^K e_k^2$$

4.2 The Levenberg-Marquardt algorithm

We solve the previous optimization problem using the iterative Levenberg-Marquardt algorithm (Marquardt, 1963; Hagan and Menhaj, 1994):

$$\underline{\theta}_{i+1} = \underline{\theta}_i + \Delta \underline{\theta} \text{ with } \Delta \underline{\theta} = (\tilde{H}_{\underline{\theta}} + \lambda I)^{-1} J'_{\underline{\theta}}$$

$J'_{\underline{\theta}}$ is the gradient of J with respect to $\underline{\theta}$, $\tilde{H}_{\underline{\theta}}$ is the pseudo-hessian matrix of J with respect to $\underline{\theta}$, and λ is a control parameter of the algorithm. $J'_{\underline{\theta}}$ and $\tilde{H}_{\underline{\theta}}$ are computed using the following relations:

$$J'_{\underline{\theta}} = -2 \sum_{k=1}^K \left(e_k \cdot \frac{\partial \hat{y}_k}{\partial \underline{\theta}} \right)$$

$$[\tilde{H}]_{j,l} = 2 \sum_{k=1}^K \left(\frac{\partial \hat{y}_k}{\partial \theta_j} \cdot \frac{\partial \hat{y}_k}{\partial \theta_l} \right)$$

Let's define the value of the sensitivity function of the parameter p for the sample k as $\sigma_p(k) = \frac{\partial \hat{y}_k}{\partial \theta_p}$. To solve the optimization problem we have to compute this value for all p and all k .

4.3 Calculus of the sensitivity functions associated with the parameters of the neural network

This calculus is referred to as dynamic back propagation (Narendra and Parthasarathy, March 1991) and has already been successfully used in the case of integer CTNN (Benoît-Marand *et al.*, 2004). It can directly be extended to fractional CTNN ($0 < n < 1$). We simply write:

$$\begin{aligned} \frac{d^n \sigma_p(t)}{dt^n} &= \frac{\partial}{\partial \theta_p} \left(\frac{d^n \hat{y}}{dt^n}(\underline{\theta}, u(t), \hat{y}(\underline{\theta}, t)) \right) \\ &= \frac{\partial}{\partial \theta_p} (s(\underline{\theta}, u(t), \hat{y}(\underline{\theta}, t))) \\ &= \frac{\partial s(t)}{\partial \theta_p} + \frac{\partial s(t)}{\partial \hat{y}(t)} \cdot \frac{\partial \hat{y}(t)}{\partial \theta_p} \\ &= \frac{\partial s(t)}{\partial \theta_p} + \frac{\partial s(t)}{\partial \hat{y}(t)} \cdot \sigma_p(t) \end{aligned}$$

So, the sensitivity functions associated with the neural network parameters can be simulated by

using the non integer integrator used in the CTNN model.

The computation of $\frac{\partial s(t)}{\partial \theta_p}$ is performed with the well-known back propagation algorithm (Cun, 1985; Rumelhart *et al.*, 1986)⁴:

$$\frac{\partial s(k)}{\partial b_i} = \tanh(n_i(k))$$

$$\frac{\partial s(k)}{\partial w_{i,j}} = r_i(k) \cdot \tanh'(n_j(k)) \cdot b_j$$

The value of $\frac{\partial s(t)}{\partial y(t)}$ is easy to compute:

$$\frac{\partial s(t)}{\partial \hat{y}(t)} = \sum_{i=1}^{i=Nc} w_{1,i} \cdot \tanh'(n_i(t)) \cdot b_i$$

4.4 Calculus of the sensitivity function associated with the parameter of the non integer integrator

To evaluate $\sigma_\alpha(t) = \frac{\partial \hat{y}(t)}{\partial \alpha}$ we derivate the state space representation (5) with respect to α . Extending the state space vector with $x_{(N+1+i)}(t) = \frac{\partial x_i(t)}{\partial \alpha}$, we obtain the following modified state space representation:

$$\dot{\underline{x}} = C^* \underline{x} + D^* \underline{v} \quad (6)$$

where:

$$\underline{v}^T = \left[f(u(t), \hat{y}(t), \underline{\theta}) \quad \frac{\partial f}{\partial \hat{y}(t)} \sigma_\alpha(t) \right]$$

$$C^* = R^{-1} C \quad D^* = R^{-1} D$$

$$\hat{y}(t) = x_{(N+1)}(t) \quad \sigma_\alpha(t) = x_{(2N+2)}(t)$$

$$D = \begin{bmatrix} D_1 & 0 \\ D_2 & D_1 \end{bmatrix} \quad R = \begin{bmatrix} M & 0 \\ R_1 & M \end{bmatrix} \quad C = \begin{bmatrix} A & 0 \\ C_1 & A \end{bmatrix}$$

M , A are the matrices defined in (5). D_1 , D_2 (vectors of size $N+1$) and R_1 , C_1 (square matrices of size $N+1$) are defined as follows:

$$\underline{D}_1^T = [G \ 0 \ \dots] \quad \underline{D}_2^T = \left[\frac{\partial G}{\partial \alpha} \ 0 \ \dots \right]$$

$$R_1 = \begin{bmatrix} 0 & & & & & \\ -1 & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & -1 & & \\ & & & & 0 & \end{bmatrix} \quad C_1 = \begin{bmatrix} 0 & & & & & \\ \frac{\partial \omega_1}{\partial \alpha} & -\frac{\partial \omega_1}{\partial \alpha} & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \frac{\partial \omega_N}{\partial \alpha} & -\frac{\partial \omega_N}{\partial \alpha} \end{bmatrix}$$

$$\frac{\partial \omega_1}{\partial \alpha} = \omega_b \quad \frac{\partial \omega_i}{\partial \alpha} = \alpha \eta \frac{\partial \omega_{i-1}}{\partial \alpha} + \omega_{i-1} \left[\eta + \alpha \frac{\partial \eta}{\partial \alpha} \right]$$

$$\frac{\partial \eta}{\partial \alpha} = \left[\frac{\omega_h}{\omega_b} \right]^{\frac{1}{N-1}} \left(\frac{N}{1-N} \right) \alpha^{\frac{2N-1}{1-N}}$$

Finally, let's calculate $\frac{\partial G}{\partial \alpha}$. G is such that magnitudes of $\hat{I}_n(\omega) = \frac{G}{j\omega} \prod_{k=1}^{1+\frac{j\omega}{\omega_k}} \frac{1}{1+\frac{j\omega}{\omega_k}}$ and $\frac{1}{(j\omega)^n}$ are equal if $\omega = \omega_u$. As a consequence:

³ K is the size of the data set

⁴ In the following equation $r_i(k)$ represents the input i of the net for the sample k . For example $r_1(k) = \hat{y}(k)$. Notice that $\tanh'(x) = 1 - \tanh^2(x)$

$$G = \omega_u^{1-n} \frac{1}{\prod G_k}$$

Where G_k is the magnitude of the k^{th} cell of $\hat{I}_n(s)$. Then :

$$\frac{\partial G}{\partial \alpha} = -\ln(\omega_u) \frac{\partial n}{\partial \alpha} G + \frac{1}{2} G \sum_{k=1}^{k=N} G_k^2 \frac{\partial}{\partial \alpha} \left(\frac{1}{G_k^2} \right)$$

$$\frac{\partial}{\partial \alpha} \left(\frac{1}{G_k^2} \right) = \{ \omega_k^2 + \alpha^2 \omega_u^2 \}^{-2} \{ 2\omega_k \frac{\partial \omega_k}{\partial \alpha} (\omega_k^2 + \alpha^2 \omega_u^2) - (\omega_k^2 + \omega_u^2) (2\omega_k \frac{\partial \omega_k}{\partial \alpha} + 2\alpha \omega_u^2) \}$$

$$\frac{\partial n}{\partial \alpha} = \frac{(\eta + \alpha \frac{\partial \eta}{\partial \alpha}) \ln(\alpha) - \eta \ln(\alpha \eta)}{\alpha \eta [\ln(\alpha \eta)]^2}$$

Often, ω_b and ω_h are such that $\omega_u = \sqrt{\omega_b \omega_h} = 1$. In this particular case :

$$\frac{\partial G}{\partial \alpha} = \frac{1}{2} G \sum_{k=1}^{k=N} G_k^2 \frac{\partial}{\partial \alpha} \left(\frac{1}{G_k^2} \right)$$

$$\frac{\partial}{\partial \alpha} \left(\frac{1}{G_k^2} \right) = \frac{2(\alpha^2 - 1) \left(\frac{\partial \omega_k}{\partial \alpha} \omega_k \right) - 2\alpha(\omega_k^2 + 1)}{(\omega_k^2 + \alpha^2)^2}$$

We are now able to evaluate the sensitivity function $\frac{\partial \hat{y}}{\partial \alpha}$.

5. SIMULATION RESULTS

We simulated the following system ⁵ :

$$\frac{d^n y(t)}{dt^n} = ay(t) + by^2(t)$$

with $a = 1$, $b = -1$, $n = 0.6$, $\omega_b = 10^{-4}$ rad/s, $\omega_h = 10^4$ rad/s and $N = 20$. Noise free and noisy data sets are plotted on figures 7 and 8.

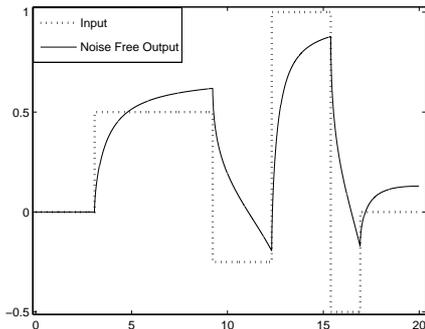


Fig. 7. Noise free data set

Then, we have chosen to use a neural network containing 3 neurons to create the CTNN model. The network has been initialized using a simple heuristic and n has been initialized to 0.3.

⁵ All simulations have been performed by using MATLAB[®] and SIMULINK[®] products.

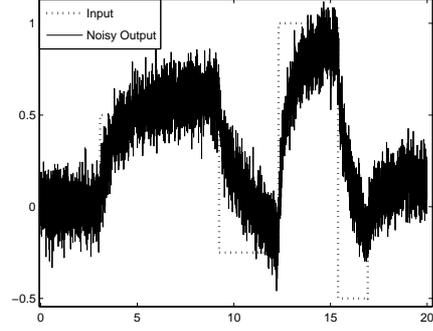


Fig. 8. Noisy data set (white noise, $S/B = 10$)

CTNN optimization results using either noise free or noisy data are plotted on figure 9 and 10. We performed 60 iterations of the Levenberg-Marquardt algorithm.

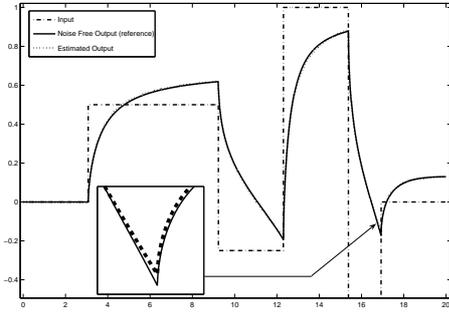


Fig. 9. Results obtained with noise free data

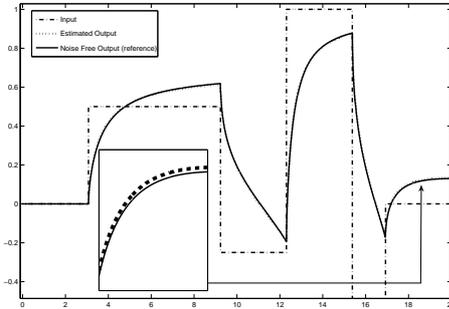


Fig. 10. Results obtained with noisy data

Finally we performed a model reduction stage in order to revert from the net to the physical law of the system. This easy step (which is a simple static identification) consists in finding the values of parameters a and b of the function ⁶ : $ay(t) + by^2(t)$ by using the estimated static net. The results are shown in the following table :

Estimations	a	b	n
with noise free data	1.0038	-1.0145	0.6074
with noisy data	0.9713	-0.9391	0.5859
Exact values	1	-1	0.6

⁶ In this particular case, the function to be identified is known but in more realistic cases we should try different expressions and see which one is the more adapted to the system.

6. CONCLUSION

In this article, we introduced the fractional CTNN model, based on the use of a static neural network followed by a non integer integrator block. Our results show that fractional CTNN allow to estimate, with a high accuracy, both the non integer derivative order and the physical law of a non linear fractional system.

At present, we are validating our identification method of fractional systems using CTNN on a real heat transfer system. Moreover, we are trying to extend the equation error method (EEI) to the identification of fractional order n . Indeed, we already experimented, in the case of integer systems, that a combination of EEI and OEI drastically reduces the identification time (which is the main problem encountered with the OEI method).

REFERENCES

- Battaglia, J.-L. (2002). Méthodes d'identification de modèles à dérivées d'ordre non entier et de réduction modale. Application à la résolution de problèmes thermiques inverses dans des systèmes industriels. PhD thesis. Université de Bordeaux I.
- Benoit-Marand, François, Laurent Signac and Jean-Claude Trigeassou (2005). Neural identification of physical processes: Analysis of disruptive factors. In: *Engineering Applications of Neural Networks*.
- Benoit-Marand, François, Laurent Signac, Smail Bachir and Jean-Claude Trigeassou (2004). Non linear identification of continuous time systems using neural networks. In: *International Workshop on Electronics and System Analysis*. Bibao, Spain.
- Cois, O. (2002). Systèmes linéaires non entier et identification par modèle non entier : application en thermique. PhD thesis. Université de Bordeaux I.
- Cun, Y. Le (1985). Une procédure d'apprentissage pour réseau à seuil assymétrique. *Cognitiva* pp. 599–604.
- Hagan, Martin T. and Mohammad B. Menhaj (1994). Training feedforward networks with the marquardt algorithm. *IEEE Transactions on neural networks*.
- Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **79**, 2554–2558.
- Lin, J. (2001). Modélisation et identification de systèmes d'ordre non entier. PhD thesis. Université de Poitiers, France.
- Lin, J., T. Poinot, J.C. Trigeassou and R. Ouvrard (2000a). Parameter estimation of fractional systems: application to the modeling of a lead-acid battery. In: *SYSID 2000, 12th IFAC Symposium on System Identification*.
- Lin, J., T. Poinot, J.C. Trigeassou, H. Kabbaj and J. Faucher (2000b). Modélisation et identification d'ordre non entier d'une machine asynchrone. In: *CIFA '2000, Conférence Internationale Francophone d'Automatique*. pp. pp. 53–56.
- matopou los, E.B. Kos M.M. Polycarpou, M.A. Chiostodoulou and P.A. Ioannou (1995). High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks* **6**(2), 422–431.
- Marquardt, D. W. (1963). An algorithm for least-squares estimation of non-linear parameters. *Soc. Indust. Appl. Math.*
- Narendra, K.S. (1992). Adaptative control of dynamical systems using neural networks. In: *Handbook of Intellignet Control*. pp. 141–183. Van Nostrand Reinhold.
- Narendra, K.S. and A.U. Levin (1997). Identification of nonlinear dynamical systems using neural networks.. In: *Neural Systems for Control*. pp. 129–160. San Diego, CA: Academic Press.
- Narendra, K.S. and K. Parthasarathy (March 1991). Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Transactions on Neural Networks* **2**, 252–262.
- Oustaloup, A. (1995). *La dérivation non entière : théorie, synthèse et applications*. hermès ed., Paris.
- Pearlmutter, Barak (1995). Gradient calculation for dynamic recurrent neural networks: a survey. *IEEE Transactions on Neural Networks* **5**, 1212–1228.
- Poinot, T. and J.-C. Trigeassou (2003). A method for modelling and simulation of fractional systems. In: *Signal Processing: Special Issue on Fractional Signal Processing and Applications*. pp. pp. 2319–2333.
- Poinot, T. and J.-C. Trigeassou (2004). Modelling and simulation of fractional systems. In: *1st IFAC Workshop on Fractional Differentiation and its Applications*. pp. pp. 656–663.
- Retière, N. and M. Ivanes (1998). Modeling of electric machines by implicit derivative half-order systems. *IEEE Power Engineering Review* pp. pp. 62–64.
- Rumelhart, D. E., G.E. Hinton and R.J. Williams (1986). Learning representations by back-propagating errors. *Nature* **323**, 533–536.
- Wang, Y. J. and C. T. Lin (March 1998). Runge-kutta neural network for identification of dynamical systems in high accuracy. *IEEE Transactions on Neural Networks* **9**, 293–307.