Modelling of a SISO and MIMO non linear communication channel using two modelling techniques

NABIHA SAIDI, OKBA TAOUALI AND HASSANI MESSAOUD
Unité de Recherche ATSI
Ecole Nationale d'Ingénieurs de Monastir
Rue Ibn El Jazzar 5019 Monastir; Tel: + (216) 73 500511, Fax: + (216) 73 500 514,
TUNISIA
saidinabiha@yahoo.fr; taoualiokba@yahoo.fr; hassani.messaoud@enim.rnu.tn

Abstract: - In this paper, we consider the modelling of a Single Input Single Output (SISO) and Multi Input Multi Output (MIMO) non linear communication channels using two modelling techniques. The first titled Volterra model built using Volterra series and possesses several important properties that make them very useful for the modelling and analysis of non linear systems and the second named RKHS model developed on a particular Hilbert Space the kernel of which is reproducing. This space known as Reproducing Kernel Hilbert Space (RKHS) uses the statistical learning theory to provide RKHS model.The performances of both models in SISO case and in MIMO case are evaluated and the results were successful.

Key-Words: - Modelling, SISO, MIMO, RKHS, Volterra, communication channel

1 Introduction

Till while ago Volterra models [1], [2] [9], [11], [13] still the most usual and popular way to describe non linear systems behaviour as it provides a model linear with respect to its parameters. Volterra model with finite memory are BIBO (Bounded Input Bounded Output) stable, they allow to model a large class of non linear systems. These models have been successfully applied to a wide variety of engineering problems such as modelling of non linear communication channels.
In communication systems, Volterra models have been used for modelling communication channels exhibiting nonlinear behaviours [12] and [24] that is the case of those including amplifiers and optical fiber. Indeed, high power amplifiers, currently used in mobile radio and satellite communication channels, have to operate near their nonlinear region for maximizing the utilization of the available power.
The last few years has registered the birth of a new modelling technique of non linear systems. This technique, developed on a particular Hilbert Space, known as Reproducing Kernel Hilbert Space (RKHS) uses the Statistical Learning Theory (SLT) to provide an RKHS model as a linear combination of the kernels forming this space. Contrary to Volterra model the model complexity is independent of the non linearity degree and the system memory. In [14] the SISO system modelling problem has been investigated and the MISO case has been processed in [15]. The MIMO case has been processed in [16].In this paper we focus on the modelling and identification of a non linear SISO and MIMO communication channel for this we use two modelling techniques such as: RKHS model and Volterra model in SISO case and in MIMO case. Section 2 is devoted to the presentation of the modelling of MISO and MIMO process in RKHS space. In section 3 we are interested to the presentation of the SISO, MISO and MIMO Volterra model. The modelling of a SISO non linear communication channel described by a Wiener-Hammerstein model and the modelling of a MIMO non linear communication channel is confined to section 4.

2 Modelling of MISO and MIMO process in RKHS

2.1 Reproducing Kernel Hilbert Space (RKHS)

Let X be a given space and let H a Hilbert space of functions defined on X. This space is doted with the scalar product $\langle .,. \rangle_H$. Consider the function $K : X^2 \to \mathbb{R}$. K is a reproducing kernel of the space H if and only if

* $\forall\, x \in X$, the function K_x such as

$$K_x : X \to \mathbb{R} \qquad t \mapsto K_x(t) = K(x,t) \tag{1}$$

is a function of the space *H*.

* $\forall\, x \in X \;;\; \forall\, f \in H \qquad \langle f, K_x \rangle_H = f(x)$ (2)

H is then a Reproducing Kernel Hilbert Space (RKHS) of kernel K.

2.2 Statistical Learning Theory (SLT)

The Statistical Learning Theory [21], [22] aims to develop a model of non linear system from a set of data $D=\{(x^{(1)},y^{(1)}),...,(x^{(N)},y^{(N)})\}$ and to evaluate the error risks associated to the resulting model.

We call learning the procedure which enables to select, from the set of observations D, the function $f_0 \in H$ that describes as close as possible the relation between any process input /output couple $(x^{(i)},y^{(i)})$ even it doesn't belong to D. To determine the optimal function, the SLT proposes to minimize the functional risk associated to the chosen function $f \in H$. This risk $R(f)$ is given by:

$$R(f)=\int_{X,Y} V(y,f(x))\ P(x,y)dxdy \quad (3)$$

Where $P(x,y)$ is the probability associated to the input/output couple (x,y) and $V(y,f(x))$ is a cost function which evaluates the error between the process output y and its estimation $f(x)$. In practice $P(x,y)$ is unknown and we minimise the empirical risk $R_{emp}(f)$ instead of $R(f)$, with

$$R_{emp}(f)=\frac{1}{N}\sum_{i=1}^{N} V\left(y^{(i)},f\left(x^{(i)}\right)\right) \quad (4)$$

However the minimisation of $R_{emp}(f)$ in the space H may lead to an over fitting of the given function so that its generalization to new data isn't assured. To solve this problem Vapnik [22] proposes to adopt the (Structural Risk Minimisation: SRM) which can be settled by amending the empirical risk by a function evaluating the complexity of the given model. To do so we minimise instead of the empirical risk, the criterion $D(f)$ which contains a regularization term that depends on the norm of the function f in the function space already chosen.

$$D(f) = \frac{1}{N}\sum_{i=1}^{N} V\left(y^{(i)}, f\left(x^{(i)}\right)\right) + \lambda \|f\|_H^2 \quad (5)$$

The parameter λ allows to tune the compromise between the empirical risk minimization and the generalization ability. The minimization of criterion (5) on an arbitrary function space can be a hard task however this can be handled when this space is an RKHS.

Based on the representer theorem [10] the optimal function f_{opt} which minimizes $D(f)$ can be written as:

$$f_{opt}(x)=\sum_{i=1}^{N} a_i\ K\left(x^{(i)},x\right) \quad (6)$$

Where $a_i,\ i=1,...,N$ are the model parameters. The norm of the function f is then

$$\begin{aligned}\|f\|_H^2 &= \langle \sum_{i=1}^{N} a_i K\left(x^{(i)},.\right), \sum_{i=1}^{N} a_i K\left(x^{(i)},.\right)\rangle_H \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j K\left(x^{(i)},x^{(j)}\right)\end{aligned} \quad (7)$$

2.3 Learning machines

The algorithms used to estimate the parameters a_i in (6) are called learning machines such as support vector machines (SVM) and, regularization network (RN)

2.3.1 Support vector machines

Support Vector Machines (SVM) have been recently developed in the framework of statistical learning theory [6], [22], [23], and have been successfully applied to a number of applications, ranging from time series prediction to face recognition, to biological data processing for medical diagnosis. Their theoretical foundations and their experimental success encourage further research on their characteristics, as well as their further use. Support Vector Regression (SVR) belongs to the category of reproducing-kernel methods, just Kernel Principal Component Analysis KPCA [5], Partial Least Square PLS [17]. Based on the theory of Support Vector Machines, SVR is now a well established method for designing black-box models in engineering. The aim of SVR is to build a model $f:\mathbb{R}^n \to \mathbb{R}$ of the output of a process or system that depends on a set of factors.

$$f(x)=\sum_{i=1}^{N} w_i \Phi(x_i)+b \quad (8)$$

where $\{\Phi(x_i)\}_{i=1,...,N}$ are the data in the features space, $\{w_i\}_{i=1,..,N}$ and b are coefficients. They can be estimated by minimizing the regularized risk function

$$R(C) = \frac{C}{N}\sum_{i=1}^{N} V_\varepsilon\left(y_i, f(x_i)\right) + \frac{1}{2}\|w\|^2 \quad (9)$$

where $V_\varepsilon(y_i, f(x_i))$ is the so-called loss function measuring the approximate errors between expected output y_i and the calculated output $f(x_i)$. And C is a regularization constant determining the trade-off between the training error and the generalization performance.

The second term, $\frac{1}{2}\|w\|^2$ is used a measurement as a of function flatness.

Introduction of slack variables ξ, ξ^* leads (9) to the following constrained function.

$$Minimize\ R(w,\xi^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad (10)$$

s.t.

$$\begin{cases} y_i - \langle w, \Phi(x_i)\rangle - b \le \varepsilon + \xi_i \\ \langle w, \Phi(x_i)\rangle + b - y_i \le \varepsilon + \xi_i^* \end{cases} \quad (11)$$

$\xi_i, \xi_i^* \ge 0$, $i = 1, ..., N$

This formulation of the problem comes back to use ε-insensitive loss function of the following shape:

$$|y - f(x)|_\varepsilon = \begin{cases} 0 & \text{if } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & \text{if } |y - f(x)| > \varepsilon \end{cases} \quad (12)$$

One can interpret this function as creating a tube of ray ε (Fig.1)

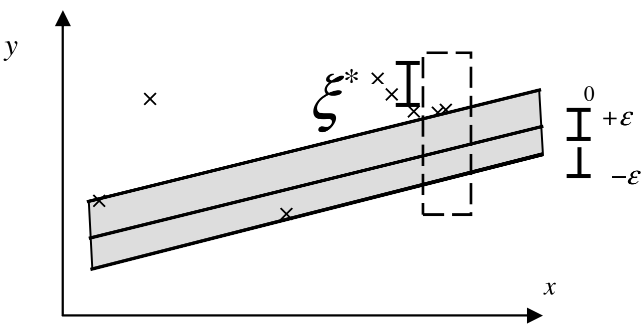

Fig. 1.

Although non-linear function Φ is usually unknown all computations related to Φ can be reduced to the form $\Phi(x)^T \Phi(x')$, which can be replaced with a so-called kernel function $K(x,x') = \Phi(x)^T \Phi(x')$ that satisfies Mercer's condition [8]. Then, Eq. (8) becomes the explicit form.

$$f(x,\alpha_i,\alpha_i^*) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x_i,x) + b \quad (13)$$

In (13), Lagrange multipliers α_i and α_i^* satisfy the equality $\alpha_i \times \alpha_i^* = 0$, $\alpha_i \ge 0$, $\alpha_i^* \ge 0$, $i = 1,...,N$

Those vectors with $\alpha_i \ne 0$ are called support vectors, which contribute to the final solution.

2.3.2 Regularization network

The cost function is:

$$V(y_i, f(x_i)) = (y_i - f(x_i))^2 \quad (14)$$

And the the optimal function is given by (6), where the sequence $\{a_i\}$ are such as:

$$a_i = \sum_{j=1}^{N} \Psi_{i,j} y_j \quad (15)$$

with $\Psi_{i,j}$ the i,j th component of the matrix $\Psi \in \mathbb{R}^{N\times N}$

$$\Psi = (G + \lambda N I)^{-1} \quad (16)$$

And the matrix $G \in \mathbb{R}^{N\times N}$ is such that:

$$G_{ij} = (K(x_i, x_j)).\ i, j = 1,...,N \quad (17)$$

Or in matrix form:

$$A = (G + \lambda N I)^{-1} Y,\ A = (a_1,..., a_N)^T \quad (18)$$

$$Y = (y_1,..., y_N)^T$$

Different types of kernels, such as:

$$Polynomial : K(x,x') = (1 + \langle x,x'\rangle)^p \quad (19)$$

$$RBF : K(x,x') = \exp(-\frac{\|x-x'\|}{p}) \quad (20)$$

$$ERBF : K(x,x') = \exp(-\sqrt{\frac{\|x-x'\|}{p}}) \quad (21)$$

2.4 Modelling of a MISO and MIMO Process in RKHS space

In the case of MISO model the output can be written as:

$$y(k) = \varphi\left[u_1, ..., u_p, k\right] + e(k) \quad (22)$$

Where φ is a non linear function, p is the input number and e(k) is an additive noise. The input vector can be defined as:

$$x = \left[u_1(k),..., u_1(M_1+k-1),...,u_p(k),...,u_p\left(M_p+k-1\right)\right]^{\mathrm{T}} \quad (23)$$

for $k = 1, \ldots, N - M_p + 1$

Where N is the observation number and M_p is the memory of the p^th^ input.

In the MIMO case the process output is a p-dimensional vector, we consider the network of kernel functions illustrated by Figure 2.

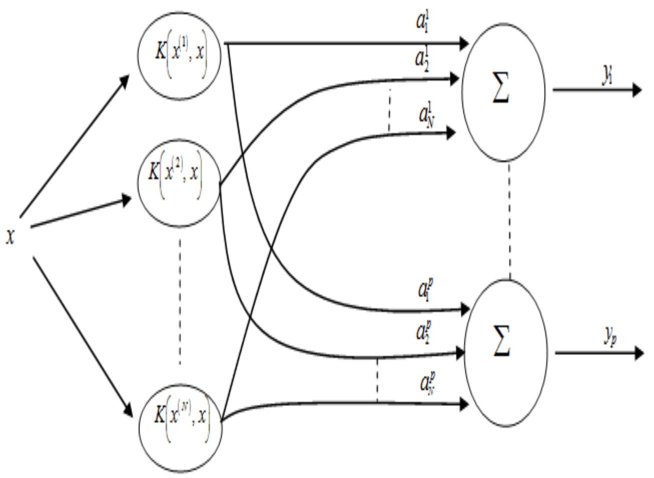

Fig.2. Network of kernels functions for the MIMO modelling

The MIMO process is considered as a set of MISO processes modelled in RKHS space as above. To decrease the model complexity, all the MISO output are linear combinations of the same kernel components and with different parameters.

The output of the q^{th} MISO model is:

$$y_q = \sum_{i=1}^{N} a_i^q K\left(x^{(i)}, x\right) \text{ for } q = 1, ..., p \quad (24)$$

$$y_q = A_q^T\ H(x) \quad q = 1, ..., p \quad (25)$$

Where:

$$H(x) = \left[H_1(x),...,H_N(x)\right] \in \mathbb{R}^N \quad (26)$$

With

$$H_i(x) = K\left(x^{(i)}, x\right), i = 1,...,N \quad (27)$$

$$A_q = \left[a_1^q,...,a_N^q\right]^{\mathrm{T}}, q = 1, ..., p \quad (28)$$

The output vector Y_p is then given by:

$$Y_p = \begin{pmatrix} y_1 \\ . \\ . \\ . \\ y_p \end{pmatrix} = \begin{pmatrix} a_1^1\, a_2^1 & . & . & . & a_N^1 \\ a_1^2\, a_1^2 & . & . & . & a_1^2 \\ . & . & . & . & . \\ a_1^p\, a_2^p & . & . & . & a_N^p \end{pmatrix} \begin{pmatrix} K(x_1, x) \\ . \\ . \\ K(x_N, x) \end{pmatrix} \quad (29)$$

Or from (26) and (28)

$$Y_p = \begin{pmatrix} A_1^T \\ . \\ . \\ . \\ A_p^T \end{pmatrix} H(x) \quad (30)$$

3 Volterra model

Volterra models have several important properties that make them very useful for the modelling and analysis of non linear systems [2], [11], [18]. These models which are linear with respect to their parameters, the kernel coefficients, suffer from the huge increasing of the parameter number depending on non linearity hardness.

3.1 SISO Volterra model

The model output is written as:

$$y(k) = \sum_{i=1}^{\infty} \left\{ \sum_{m_1=0}^{\infty} \cdots \sum_{m_i=0}^{\infty} h_i(m_1, \cdots, m_i) \prod_{n=1}^{i} u(k - m_n) \right\} \quad (31)$$

Where *u* and *y* are the input and the output of the process respectively and $h_i(m_1, \cdots, m_i)$ is the i^th^ Volterra kernel. For causal and stable system, the infinite sums in (31) can be truncated to a finite one as:

$$y(k) = h_0 + \sum_{i=1}^{P} \sum_{m_1=0}^{M-1} \sum_{m_2=0}^{M-1} \dots \sum_{m_i=0}^{M-1} h_i(m_1, ..., m_i) \times \prod_{j=1}^{i} u(k - m_j) \quad (32)$$

Where P is the process non linearity degree, M is the memory and h_0 is the statistical characteristic.

The Volterra model can be seen as a natural extension of the linear system impulse response to non linear systems. Although it is linear with respect to its parameter such model suffers from the increasing of its parameter number and any attempt for its using in real time application may fail if a reduction operation doesn't precede such attempt. The parameter number of the Volterra model given by (32) is:

$$n_p = 1 + \sum_{i=1}^{P} M^{i} \quad (33)$$

To reduce this number we use generally the triangular form of the Volterra model, given as:

$$y(k) = h_0 + \sum_{i=1}^{P} \sum_{m_1=0}^{M-1} \sum_{m_2=m_1}^{M-1} \dots \sum_{m_i=m_{i-1}}^{M-1} h_i(m_1, m_2, ..., m_i) \times \prod_{j=1}^{i} u(k - m_j) \quad (34)$$

And the relevant parameter number of such model is:

$$n_{tri} = 1 + \sum_{i=1}^{P} \frac{(M-1+i)!}{(M-1)!\,i!} \quad (35)$$

3.2 MISO Volterra model

For multiple inputs [19], [20], the output of the Volterra model in its triangular form is:

$$y(k) = h_0 + \sum_{i=1}^{P} \sum_{j_1=1}^{n} \dots \sum_{j_i=1}^{n} \sum_{m_1=0}^{M-1} \dots \sum_{m_i=m_{i-1}}^{M-1} h_{j_1, j_2, \dots, j_i}(m_1, \dots, m_i) \times \prod_{e=1}^{i} u_{j_e}(k - m_e) \quad (36)$$

Where $u(k) = [u_1(k)\ u_2(k)\ \cdots\ u_n(k)]^T$ and $y(k)$ are the process input vector and output respectively and $h_{j_1, j_2, \dots, j_i}(m_1, \dots, m_i)$ is the Volterra kernel. P is the non linearity degree and M is the memory. The corresponding parameter number is:

$$n_{MISO} = 1 + \sum_{i=1}^{P} n^{i} \frac{(M-1+i)!}{(M-1)!\,i!} \quad (37)$$

3.3 MIMO Volterra model

The MIMO system can be considered as a set of Multi Input Single Output (MISO) sub systems. Thus the modelling of the MIMO System is equivalent to the modelling of its sub systems. Let a MIMO system with n inputs and S outputs, each subsystem output $y_s(k)$ can be developed on Volterra series as:

$$y_s(k) = h_0^s + \sum_{i=1}^{P} \sum_{j_1=1}^{n} \dots \sum_{j_i=1}^{n} \sum_{m_1=0}^{M-1} \dots \sum_{m_i=m_{i-1}}^{M-1} h^{s}_{j_1, j_2, \dots, j_i}(m_1, \cdots, m_i) \times \prod_{e=1}^{i} u_{j_e}(k - m_e) \quad (38)$$

$h^{s}_{j_1, j_2, \dots, j_i}(m_1, ..., m_i)$ is the Volterra Kernel of i^th^ order corresponding to the sub system the output of which is $y_s(k)$ and h_0^s is the statistical characteristic corresponding to $y_s(k)$.

And the parameter number is:

$$n_{MIMO} = \left(1 + \sum_{i=1}^{P} n^{i} \frac{(M-1+i)!}{(M-1)!\,i!}\right) * S \quad (39)$$

4 Application

4.1 Modelling of a SISO non linear communication channel

Some physical systems, like the communication channel representing the access to a wireless network via optic fiber can be modelled by simplified Volterra models such as Hammerstein and Wiener models [2]. In this paragraph we will be interested to the identification and the modelling of a Wiener- Hammerstein channel by Volterra model and RKHS model.

The SISO communication channel is described by a Wiener-Hammerstein model [3] as presented in Figure 2.

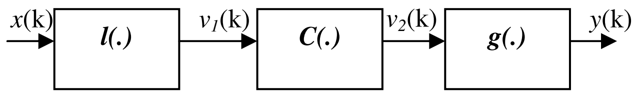

Fig.3: Wiener-Hammerstein model

x(k), y(k), v_1(k) and v_2(k) are respectively the PAM-2 input sequence, the channel output, the input and the output of the non linear filter *C*(.). The channel Wiener Hammerstein model is given by:

$$v_1(k) = \sum_{i=0}^{M_l} l(i)\, x(k-i) \;;\; v_2(k) = \sum_{p=1}^{P} C_p\, v_1^p(k) \;;$$

$$y(k) = \sum_{i=0}^{M_g} g(i)\, v_2(k-i) + v(k)$$

The channel to be modelled and identified, is given in [4] with $M_l = M_g = 2$ and its non linearity degree is P = 3 and its coefficients are: $l(0) = 1$; $l(1) = 0.3$; $l(2) = 0.1$; $C_1 = 2$; $C_2 = 0.8$; $C_3 = 0.5$; $g(0) = 1$; $g(1) = 0.5$;and $g(2) = 0.2$, v(k) is an additive noise.

The *NMSE* between the output of the channel y(k) and the estimated output $\tilde{y}$ (k) is:

$$NMSE = \frac{\sum_{k=1}^{N_m} (y(k) - \tilde{y}(k))^2}{\sum_{k=1}^{N_m} (y(k))^2} \tag{40}$$

The additive noise evaluated by the Signal to Noise Ratio SNR for the output of the channel.

$$SNR = \frac{\sum_{k=0}^{N_m} (y(k) - \bar{y})^2}{\sum_{k=0}^{N_m} (v(k) - \bar{v})^2} \tag{41}$$

With N_m the observation number, $\bar{y}$ and $\bar{v}$ are the mean values of the channel output y(k) of and the noise value v(k) respectively.

- ***SISO Volterra model***

This non linear communication channel can be modelled by a Volterra model of non linearity degree P = 2 and a memory M = 3. The parameter number of the model is 10. Identification of parameters is done with Recursive Least Square algorithm. In Figure 4 we plot the real output of the channel and Volterra output, we notice the high concordance between both outputs.

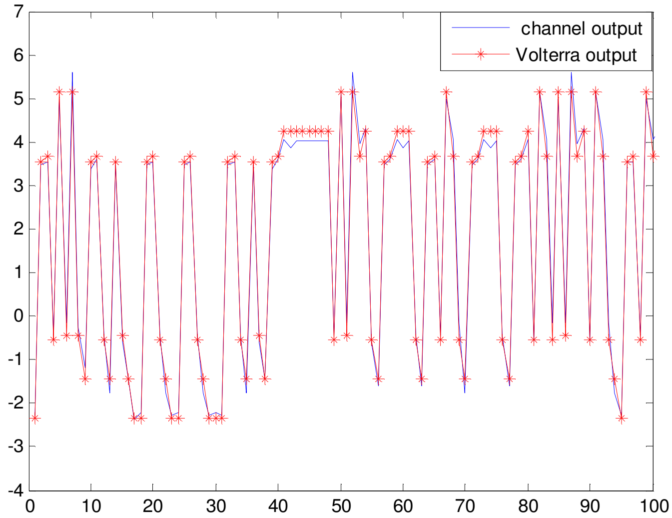

Fig.4: Validation of Volterra model P = 2 and M = 3

- ***RKHS model***

This non linear communication channel can be also modelled by RKHS model with polynomial kernel. The number of observation in the learning phase is equal to 50 and in the validation phase is 100. The total number of parameter to be identified is equal to 50. For the same input sequence we plot in Figure 5 in the validation phase the channel output and the RKHS output we note the high similarity between both outputs.

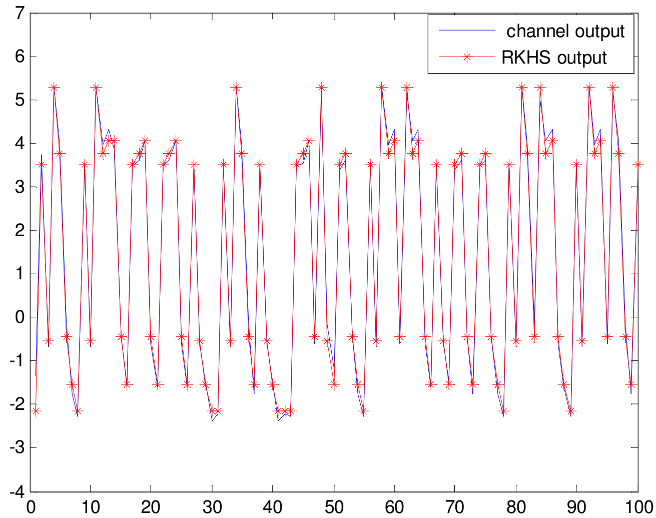

Fig.5: Validation of RKHS model, polynomial kernel

Table 1 summarizes results obtained for the two modelling techniques

Table 1: Performances of both models (without noise)

Models	Parameter number	Computing Time(s)	NMSE (%)	Parameter Of models
Volterra Model	10	0.046	0.42	P = 2 M = 3
	21	0.15	0.15	P = 2 M = 5
RKHS Model	50	0.09	0.34	Polynomial Kernel ($p_1 = 3$)

From Table 1 we can notice that the complexity of RKHS model is high because the parameter number is equal to the observation number. The complexity of Volterra model depends on the non linearity degree and the memory. We conclude that for Volterra model when the complexity of the model increase the NMSE decrease.

- ***Noise effect :***

To raise the influence of an additive noise on the identification quality we draw in Table 2 the NMSE for different values of SNR.

Table 2: Noise effect

SNR	Volterra model	RKHS model
	NMSE(%)	NMSE(%)
50	1.62	1.58
30	2.97	2.78
20	3.80	3.43
10	7.06	5.50
5	12.68	12.57

We note that for a small value of SNR the NMSE become high for the two models. The NMSE for both models is comparable.

4.2 Modelling of a MIMO non linear communication channel

Consider a MIMO non linear communication channel characterised by the number of sources (users) and the number the received antenna. This channel can be modelled by a MIMO Volterra model given by:

$$y_s(k) = h_0^s + \sum_{i=1}^{P} \sum_{j_1=1}^{n} \dots \sum_{j_i=1}^{n} \sum_{m_1=0}^{M-1} \dots \sum_{m_i=m_{i-1}}^{M-1} h^s_{j_1,j_2,\cdots,j_i}(m_1,\cdots,m_i) \times \prod_{e=1}^{i} u_{j_e}(k-m_e) + v_s(k) \quad (42)$$

Where y_s (s = 1, …, S) is the signal received by the s[th] antenna at time instant k, P is the non linearity order of the channel and M is the channel memory. $h^s_{j_1,j_2,\cdots,j_i}(m_1,\cdots,m_i)$ are the kernel coefficients of the s[th] subchannel and $v_s(k)$ is the additive white Gaussian noise to the s[th] antenna, it is assumed that the noise components are zero mean.

Fig.6: MIMO non linear channel

Consider the non linear Multiple Input Multiple Output MIMO Volterra channel [7] described by:

$$\begin{cases} y_1(k) = 2\,u_1(k) + 0.4\,u_1(k-1) + 0.08\,u_1(k-2) \\ \qquad +2\,u_1(k)\,u_1(k-1) + 0.2\,u_1(k)\,u_2(k-1) \\ \qquad +0.2\,u_2(k)\,u_1(k-1) + 0.1\,u_1^2(k-1) + v_1(k) \\ \\ y_2(k) = u_2(k) + 0.3\,u_2(k-1) + 0.09\,u_2(k-2) \\ \qquad + 0.01\,u_1^2(k-1) + 0.01\,u_2^2(k-1) + v_2(k) \end{cases} \quad (43)$$

Where $u_1 \in \{-1, 1\}$ and $u_2 \in \{-2, 2\}$ are the channel inputs, y_1 and y_2 are its outputs and v_1 and v_2 are additive white noise.

The *NMSE* between the output of the channel $y_s(k)$ and the estimated output $\tilde{y}_s(k)$ is given by:

$$NMSE(s) = \frac{\sum_{k=1}^{N_m} (y_s(k) - \tilde{y}_s(k))^2}{\sum_{k=1}^{N_m} (y_s(k))^2} \quad (44)$$

The additive noise is evaluated by the Signal to Noise

Ratio SNR(s) for the s^{th} output of the channel.

$$SNR(s) = \frac{\sum_{k=0}^{N_m} (y_s(k) - \overline{y_s})^2}{\sum_{k=0}^{N_m} (v_s(k) - \overline{v_s})^2} \quad (45)$$

With N_m the observation number, $\overline{y}_s$ and $\overline{v}_s$ are the mean values of the s^{th} channel output $y_s(k)$ and the s^{th} noise value $v_s(k)$ respectively.

4.2.1 Modelling in RKHS space

To build the RKHS model we use the polynomial Kernel (19). Where $p = 2$, the regularisation term is $\lambda = 5\times10^{-9}$.
In the identification phase we use a training set of 250 inputs/outputs and in the validation phase 120 new inputs/outputs are used to evaluate the performance of the resulting RKHS model.

- *First output of the channel*

Figure 7 plots the first output of the MIMO non linear channel we notice a concordance between the RKHS model output and the process output in the validation phase. The NMSE in validation is to 7.12% for an SNR = 10.

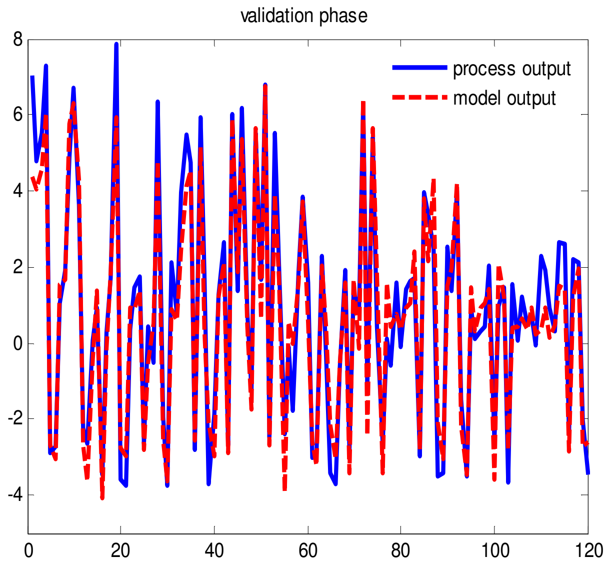

Fig.7: Validation of the first output

- *Second output of the channel*

In Figure 8 we plot the second output of the channel we notice a concordance between the RKHS model output and the process output in the learning phase and this concordance remains excellent in the validation phase. The NMSE validation is 5. 75% for an SNR = 10.

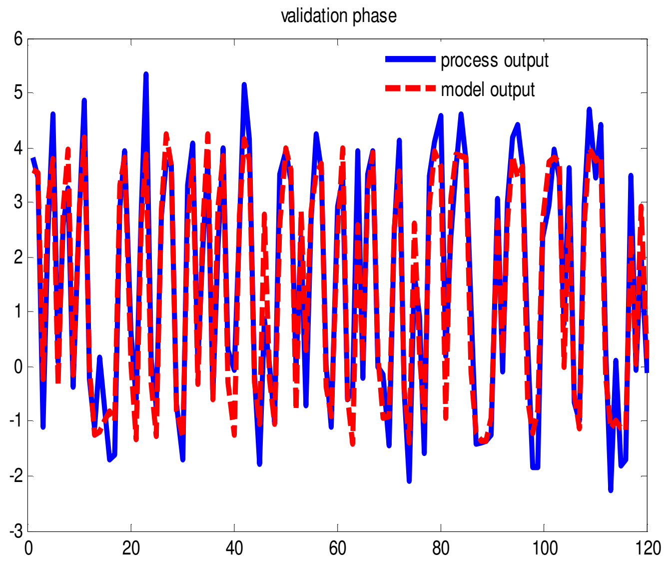

Fig.8: Validation of the second output

4.2.2 MIMO Volterra model

This MIMO non linear channel can be modelled by a MIMO Volterra model with non linearity degree P = 2 and a memory M = 2. The total parameter number of the reduced model is 34.

- *First output of the channel*

We plot in Figure 9 the validation of the first output of the channel and the output of Volterra model; we note the concordance between both outputs.

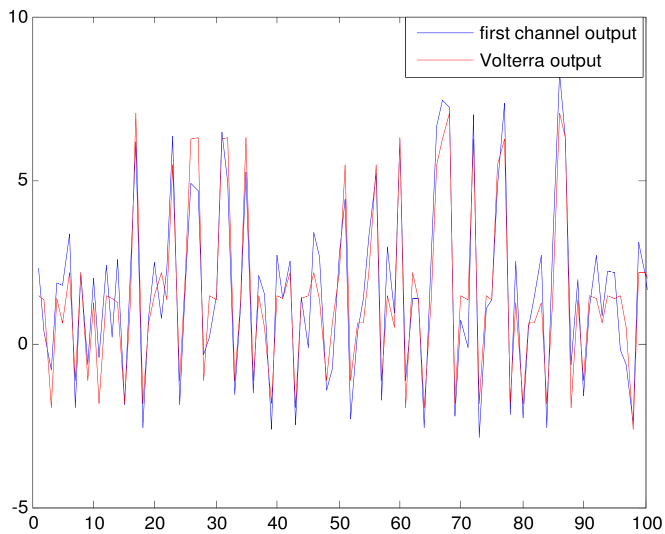

Fig.9: Validation of the first output

The NMSE for the first subsystem is 5.89% for an SNR = 10.

- *Second output of the channel*

In Figure 10 we plot the second output of the channel and the output of the model we note the concordance between both outputs.

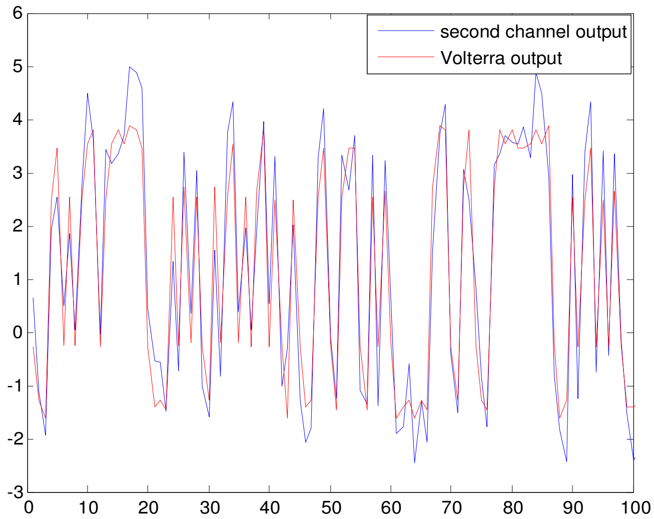

Fig. 10: Validation of the first output

The NMSE for the first subsystem is 7.79% for SNR = 10.

Results are grouped in Table 3

Table 3: Performances of both models

Models	Paramete r number	NMSE		Model parameter
		1^{st} output	2^{nd} output	
MIMO Volterra model	34	5.89%	7.79%	Non linearity degree P = 2 Memory M=2
MIMO RKHS model	250	7,21%	5,77%	- Polynomial Kernel (p = 2) - Regularisa-tion term $\lambda = 5\times10^{-9}$

From Table 3 we conclude that the NMSE for both models is comparable but the complexity of RKHS model is higher than Volterra model.

We can conclude that this MIMO non linear communication channel can be modelled by a MIMO Volterra model, the complexity of this model depends on the non linearity degree and the memory.

This channel can be also modelled by a MIMO RKHS model, this model is characterized by a high number of parameter to be identified and this number depends only on the observation number.

The RKHS modelling proud of its independence of the degree of non linearity and the memory of the model which constraint the models developed on Volterra series and cause the exponential increasing of their parameter number. Contrarily the parameter number depends only on the observation number and may be very smaller compared to that engaged in Volterra series models especially for higher nonlinear systems.

5 Conclusion

This paper has dealt with the study of two non linear SISO and MIMO system modelling techniques the Volterra model and the RKHS model. The complexity of Volterra model depends on degree of non linearity and on the memory of the system and for RKHS model the complexity depends only on the number of observations. These models have been tested for modelling a SISO non linear numerical communication channel described by a Wiener-Hammerstein model and a two input two output non linear communication channel and results are satisfactory. Simulations are carried out to evaluate the models performances and the influence of an additive noise on these performances.

References:

[1] A. Khouaja, *Modélisation et identification de systèmes non linéaires à l'aide de modèle de Volterra à complexité réduite*, Doctorat, Université de Nice Sophia Antipolis – UFR sciences, 2005.

[2] A. Kibangou, *Modèle de Volterra à complexité réduite : estimation paramétrique et application à l'égalisation des canaux de communication*, Doctorat, Université de Nice Sophia Antipolis, France, 2005.

[3] A. Kibangou, G. Favier, Wiener-Hammerstein systems modelling using diagonal Volterra kernels coefficients, *IEEE signal processing letters*, VOL.13, NO.6, 2006.

[4] A. Kibangou, G. Favier, Blind joint identification and equalization of Wiener-Hammerstein communication channels using PARATUCK2

tensor decomposition, *IEEE signal processing letters*, VOL.13, NO.6, 2006.

[5] B. Scholkopf, A. Smola, K-R. Muller, Nonlinear component analysis as Kernel eigenvalue problem, *Neural computation*, vol 10, 1998, pp. 1299-1319.

[6] C. Cortes, Vapnik, V, Support vector networks, *Machine Learning*, vol 20, 1995, pp.1–25.

[7] F. Yangwang, J. Licheng, P. Jin, MIMO Volterra filter equalization using Pth-order inverse approach, *0-7803-6293-4/00/$10.00 02000 IEEE.*

[8] J.P.Vert, *cours de DEA* à l'école des mines 2003/2004, (Jean-Philippe.Vert@mines.org)

[9] G . Budura, C . Botoca, Modeling and Identification of Nonlinear Systems Using the LMS Volterra Filter, *WSEAS TRANSACTIONS on SIGNAl PROCESSING*, No.2, February, 2006, pp. 190-197.

[10] G. Wahba, *An introduction to model building with Reproducing Kernel Hilbert Spaces*. Technical report No 1020, Department of Statistics, University of Wisconsin-Madison, April 2000.

[11] G. Budura, C. Botoca, Efficient Implementation and Performance Evaluation of the Second Order Volterra Filter Based on the MMD Approximation, *WSEAS TRANSACTIONS on CIRCUITS AND SYSTEMS,* Vol 7, March 2008, pp.139-149.

[12] K. Feher, Digital Communications—Satellite/Earth Station Engineering, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[13] M. Schetzen, "The Volterra and Wiener Theories of Nonlinear Systems", Wiley, New York, 1980.

[14] N. Aronszajn, Theory of reproducing kernels, Transaction. *American Mathematical Society,* 1950, pp. 337–404.

[15] O. Taouali, I. Aissi and H. Messouad. Identification des Systèmes MISO non linéaires modélisés dans les Espaces de Hilbert à Noyau Reproduisant (RKHS), *La cinquième Conférence Internationale d'Electrotechnique et d'Automatique*, 02-04 Mai 2008, Hammamet, Tunisie, pp. 1095-1099.

[16] O.Taouli, E. Ilyes ,V. Nathalie and M. Hassani, Identification of Non Linear Multivariable Processes Modelled on Reproducing Kernel Hilbert Space: Application to Tennessee Process. *Accepted at ICONS 2009, 2nd IFAC International Conference on Intelligent Control Systems and Signal Processing. 2009*

[17] P. Geladi, B. Kowlaski, Partial least square regression, *A tutorial, Analytica Chemica Acta*, vol35, 1986, pp.1-17.

[18] S. Boyd, L. Chua, Fading memory and the problem of approximating nonlinear operators with Volterra series, *IEEE Trans.Circuits Systems* No32, vol11,1985 ,pp.1150–1161.

[19] T. Treichl, S. Hofmann, D. SchrÄoder, Identification of nonlinear dynamic MISO systems with orthornormal base function models, *In IEEE-ISIE'02*, L'Aquila, Italy, 2002.

[20] T. Treichl, S. Hofmann and D. SchrÄoder, Identifiation of nonlinear dynamic MISO systems on fundamental basics of the Volterra theory, *PCIM'02*, NÄurnberg, Germany, 2002.

[21] T. Evgeniou, M. Pontil, T. Poggio,Statistical Learning Theory: a primer, *international journal of computer vision* , 2000, pp.9-13

[22] V.N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[23] V.N.Ghate , S.V. Dudul, Induction Machine Fault Detection Using Support Vector Machine Based Classifier, WSEAS TRANSACTIONS on SYSTEMS, Volume 8, May 2009, pp. 591-603

[24] X.N. Fernando, A. Sesay, Adaptive asymmetric linearization of radio over fiber links for wireless access, IEEE Trans. Vehicular Technol. 51 (6) (2002) 1576–1586