Identification of Non linear MISO Process using RKHS and Volterra models

OKBA TAOUALI, NABIHA SAIDI & HASSANI MESSAOUD
Unité de Recherche d'Automatique, Traitement de Signal et Image (ATSI),
Ecole Nationale d'Ingénieur Monastir
Rue Ibn ELJazzar 5019 Monastir ; Tel : +(216) 73 500 511, Fax : + (216) 73 500 514,
TUNISIA
taoualiokba@yahoo.fr;saidinabiha@yahoo.fr;hassani.messaoud@enim.rnu.tn

Abstract: - This paper treats the comparison between the Volterra model and Reproducing Kernel Hilbert Space (RKHS) model in Multiple Input Single Output (MISO) case. The RKHS model uses the Statistical learning theory to find a solution of a regularization risk. It is characterise by a linear combination of the kernels function. The complexity of Volterra model is depending of the degree and the memory of the model contrarily of the RKHS model which depend only of the number of observations. The performances of both models are evaluated first by using Monte Carlo numerical simulations and then have been tested for modelling of a chemical reactor and results are successful.

Key-Words: - Statistical Learning Theory, RKHS, Volterra, MISO, Modelling, Chemical reactor

1 Introduction

Volterra models process have several important properties that make them very useful for the modeling and analysis of non linear systems [1], [9], [17], [12], [26]. A significant advantage of the Volterra models, if compared with other nonlinear models, is that their input-output relation is linear with respect to the filter coefficients. The nonlinearity is reflected only by the multiple products between the delayed versions of the input signal. Each homogeneous term can be viewed as a multidimensional convolution; Volterra models with finite memory are BIBO (Bounded Input Bounded Output) stable; they allow to model a large class of non linear systems. Indeed, it has been shown that causal time invariant non linear systems with fading memory can be described with a finite degree of precision by truncated Volterra models [8], [18].

Many methods from the linear filter theory can be applied to the Volterra filter. For example, adaptive methods and algorithms are widely used in applications dealing with kernels estimation. Various Least Mean Square (LMS) and recursive least square (RLS) algorithms have been applied to the problem of Volterra kernel estimation [2], [9], [10], [11], [12], [24]. However this elegy is disqualified by the huge increasing of the parameter number depending on non linearity hardness.

As an alternative to this modelling strategy the last few years has registered the birth of a new modelling technique 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 an RKHS model as a linear combination of the kernels forming the RKHS space. The RKHS modelling proud of its independence of the degree 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.

In this paper we are concerned by the comparison of Volterra model and RKHS model in MISO case [15].

In paragraph 2, we remind the Volterra model in SISO and MISO case and we present the statistical learning theory (SLT) which uses the RKHS to yield MISO models.

The paragraph 3 is devoted to the comparative study of these two models, for each model some setting parameters are tuned and some performances such as the parameter number, the Normalized Mean Square Error (NMSE) and the computing time are evaluated. The validation of such models is carried out through Monte Carlo simulations where the influence of an additive noise is evaluated for the two models. These two models are tested on a physical process the Continuous Stirred chemical Reactor and the results are successful.

2 Non linear system modelling

2.1 Volterra model

2.1.1 SISO Volterra model

The model output is written as:

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

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 (1) can be truncated to a finite one as:

$$y(k) = \sum_{i=1}^{P}\sum_{m_1=0}^{M-1}\sum_{m_2=0}^{M-1}\cdots\sum_{m_i=0}^{M-1}h_i(m_1,\ldots,m_i) \times\prod_{j=1}^{i}u(k-m_j) \quad (2)$$

With P the non linearity degree, M the system memory, $h_i(m_1, m_2, \ldots, m_i)$ the i[th] Volterra Kernel. The Volterra model can be seen as a natural extension of the linear system impulse response to non linear systems. Although it is non linear with respect to its inputs, this model is linear with respect to its parameters which enables to apply some identification techniques developed in linear case. The parameter number np of the Volterra model is given by:

$$np=\sum_{i=1}^{P}M^{\,i} \quad (3)$$

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

$$y(k) = \sum_{i=1}^{P}\sum_{m_1=0}^{M-1}\cdots\sum_{m_i=m_{i-1}}^{M-1}h_i(m_1, m_2, \ldots, m_i) \times\prod_{j=1}^{i}u(k-m_j) \quad (4)$$

And the relevant parameter number of such model is:

$$np=\sum_{i=1}^{P}\frac{(M-1+i)!}{(M-1)!\,i!} \quad (5)$$

2.1.2 MISO Volterra model

For multiple input single output process [19], [20], the output of the triangular form of Volterra model is:

$$y(k)=\sum_{i=1}^{P}\sum_{j_1=1}^{n}\cdots\sum_{j_i=1}^{n}\sum_{m_1=0}^{M-1}\cdots\sum_{m_i=m_{i-1}}^{M-1}h_{j_1,j_2,\ldots,j_i}(m_1,\ldots,m_i) \times\prod_{e=1}^{i}u_{j_e}(k-m_e) \quad (6)$$

Where $u(k)=\left[u_1(k)\ u_2(k)\ \cdots\ u_n(k)\right]^T$ and y(k) are the process input vector and output respectively. The parameter number is

$$np=\sum_{i=1}^{P}n^{\,i}\,\frac{(M-1+i)!}{(M-1)!\,i!} \quad (7)$$

2.2 RKHS model

2.2.1 Statistical Learning Theory (SLT)

The Statistical Learning Theory [21], [22] aims to model an input/output process from a set of observations $D=\{(x_1,y_1),\ldots,(x_N,y_N)\}$, by selecting, in a pre-definite set of functions H, the function f_0 that fits as most as possible the relation between the process inputs x_i and outputs y_i and the optimal function is that which minimizes the following functional risk.

$$R(f)=\int_{X,Y}V(y,f(x))\,P(x,y)\,dxdy \quad (8)$$

Where (X,Y) is a random vector of distribution P of which the (x_i,y_i) are independents distributed observations and $V(y,f(x))$ is a cost function that measures the deviation between the process output y and its approximation by $f(x)$. Since the distribution $P(x,y)$ is unknown, the risk $R(f)$ can not be evaluated. To overcome this situation we minimize rather the empirical risk $R_{emp}(f)$.

$$R_{emp}(f)=\frac{1}{N}\sum_{i=1}^{N}V(y_i\ ,f(x_i)) \quad (9)$$

Where N is the observation number and $V(y_i\ ,f(x_i))$ is a loss function. However, the direct minimization of $R_{emp}(f)$ in the H isn't the best estimate of the minimization of the risk $R(f)$. Indeed, the minimization of the empirical risk often, leads to overfitting of the function reserved to the data and the generalization of the model to new observations other than that in D may

not be guaranteed. To solve this problem, [21] proposed the structural risk minimization (SRM) principle.
It consists on penalizing the empirical risk by a function estimating the complexity of the reserved model given by:

$$R_F(f) \le R_{emp}(f) + \sqrt{\frac{h.\left(\ln\left(\frac{2.N}{h}\right)+1\right) - \ln\left(\frac{\eta}{4}\right)}{N}} \quad (10)$$

The idea of SRM is to define a nested sequence of hypothesis spaces:

$$H_1 \subset H_2 ... \subset H_Q, \quad (11)$$

Where each hypothesis space H_q has finite capacity h_q and larger than that of all previous sets, that is:

$$h_1 \le h_1 ... \le h_Q. \quad (12)$$

For example H_q could be the set of polynomials of degree q, or a set of splines with q nodes, or some more complicated non linear parameterization. Using such a nested sequence of more and more complex hypothesis spaces, SRM consists of choosing the minimizer of the empirical risk in the space H_{q^*} for which the bound on the structural risk, as measured by the right hand side of inequality (10), is minimized.
Further information about the statistical properties of SRM can be found in [13], [22].
To summarize, in SLT the problem of learning form examples is solved in three steps:
a- we define a loss function $V(y, f(x))$ measuring the error of predicting the output of input x with $f(x)$ when the actual output is y;
b- we define a nested sequence of hypothesis spaces $H_q, q = 1,...,Q$ whose capacity is an increasing function of q;
c- we minimize the empirical risk in each of H_q and choose, among the solutions found, the one with the best trade off between the empirical risk and the capacity as given by the right hand side of inequality (10).
This leads to the minimization of the criterion defined in the equation

$$\min_{f \in H} D(f) = \frac{1}{N}\sum_{i=1}^{N} V(y_i, f(x_i)) + \lambda \|f\|_H^2 \quad (13)$$

Where the parameter λ is regularization term allowing to adjust the trade-off between the minimization of the empirical risk and generalization ability. The bigger λ is, the more important the regularity for the solution will be.
The minimization of the criterion (13) on any arbitrary function space H, possibly of infinite dimension, is not an easy problem. However, this task may be accomplished easily when this space is a Reproducing Kernel Hilbert Space (RKHS).

2.2.2 Reproducing Kernel Hilbert space :RKHS

Let $E \subset \mathbb{R}^d$ and $L^2(E)$ the Hilbert space of square integrable functions defined on E. Let $k: E \times E \to \mathbb{R}$ be a continuous positive definite kernel and an operator L_k defined by:

$$L_k[f](x) = \int_E k(x,t)\ f(t)\ dt \quad (14)$$

where $f \in L^2(E)$ and $x \in E$.
L_k is a linear operator having a sequence of eigenfunctions $(\psi_1, \psi_2, ..., \psi_l)$ and a sequence of corresponding real positive eigenvalues $(\sigma_1, \sigma_2, ..., \sigma_l)$ (where l can be infinite) and satisfying:

$$L_k[\psi_i] = \sigma_i \psi_i \quad (15)$$

According to the Mercer theorem [4], [5], these eigen functions constitute an orthonormal system in $L^2(E)$ and the kernel k can be written as follows:

$$k(x,t) = \sum_{j=1}^{l} \sigma_j\ \psi_j(x)\ \psi_j(t) \quad (16)$$

Let H be a Hilbert space defined by:

$$H = \left\{ f \in L^2(E), f = \sum_{i=1}^{l} w_i \varphi_i \text{ , with } \sum_{j=1}^{l} \frac{{w_j}^2}{\sigma_j} < +\infty \right\} \quad (17)$$

Where $\varphi_i = \sqrt{\sigma_i}\,\psi_i$ $i = 1, ..., l$, and the inner product is given by:

$$\langle f, g \rangle_H = \left\langle \sum_{i=1}^{l} w_i\ \varphi_i, \sum_{i=1}^{l} z_i\ \varphi_i \right\rangle_H = \sum_{i=1}^{l} w_i\, z_i \quad (18)$$

k is said to be the reproducing kernel of the Hilbert space H if and only if:

a- $\forall x \in E$, $k(x,.) \in H$ (19)

b- $\forall x \in E$ and $\forall f \in H$:

$$\langle f(.), k(x,.) \rangle_H = f(x) \quad (20)$$

H is called reproducing kernel Hilbert space (RKHS) with kernel k and dimension l. Moreover, for any RKHS, there exists only one positive definite kernel and vice versa [4].

Let consider the application Φ :

$$\Phi(x) = \begin{pmatrix} \varphi_1(x) \\ \vdots \\ \varphi_l(x) \end{pmatrix} \quad (21)$$

Φ transforms the input space E into a high dimensional feature space and the relation (16) can then be written:

$$k(x,\ t) = \langle \Phi(x), \Phi(t) \rangle \quad (22)$$

2.2.3 RKHS and representer theorem

Let's assume that the random variable X takes its values in the space $E \subset \mathbb{R}^d$ and let us consider a function $K: E^2 \to \mathbb{R}$ K, called kernel such as:

1 K is symmetric ;
2 K is positive definite, i.e., for any integer n, any sequence of elements $(x_i)_{i=1,\ldots,N} \in E^2$ and any set of coefficients $(c_i)_{i=1,\ldots,N}$, we have :

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j K(x_i, x_j) \geq 0 \quad (23)$$

The representer theorem [7] proves that the solution of the optimization problem given by (9) in the Reproducing Kernel Hilbert space H space can be written as:

$$f_{opt} = \sum_{i=1}^{N} a_i \ K(x_i,.). \quad (24)$$

In this case, the optimization problem (13) is equivalent to quadratic optimization problem of N real (a_i)

Furthermore, the space H becomes implicit and is simply visible by means of its kernel thanks to the property (b) called also kernel trick.

In particular, the square norm of the solution function f in the space H is given by:

$$\|f\|_H^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j K(x_i, x_j) \quad (25)$$

2.2.4 Learning machines

The algorithms used to estimate the parameters a_i in (24) are called learning machines such as support vector machines (SVM) and, regularization network (RN)

2.2.3.1 Support vector machines

Support Vector Machines (SVM) have been recently developed in the framework of statistical learning theory [21], [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 [3], Partial least square PLS [16]. 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**.**

$$f(x) = \sum_{i=1}^{N} w_i \Phi(x_i) + b \quad (26)$$

where $\{\Phi(x_i)\}_{i=1,\ldots,N}$ 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

$$R(C) = \frac{C}{N}\sum_{i=1}^{N} V_\varepsilon(y_i,\ f(x_i)) + \frac{1}{2}\|w\|^2 \quad (27)$$

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 (27) to the following constrained function.

$$Minimize\ R(w,\xi^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \qquad (28)$$

s.t.

$$\begin{cases} y_i - \langle w, \Phi(x_i)\rangle - b \le \varepsilon + \xi_i \\ \langle w, \Phi(x_i)\rangle + b - y_i \le \varepsilon + \xi_i^* \end{cases} \qquad (29)$$

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

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

$$|y - f(x)|_\varepsilon = \begin{cases} 0 & \text{if } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & \text{if } |y - f(x)| > \varepsilon \end{cases} \qquad (30)$$

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

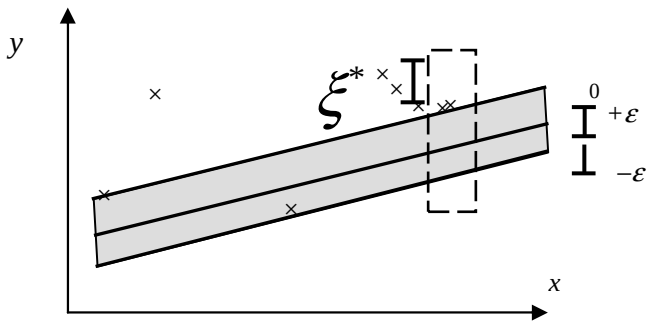

Fig. 1.

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 [5]. Then, Eq. (26) becomes the explicit form.

$$f(x, \alpha_i, \alpha_i^*) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x_i, x) + b \qquad (31)$$

In (31), 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.

2.2.3.2 Regularization network

The cost function is:

$$V(y_i, f(x_i)) = (y_i - f(x_i))^2 \qquad (32)$$

And the the optimal function is given by (24), where the sequence $\{a_i\}$ are such as:

$$a_i = \sum_{j=1}^{N} \Psi_{i,j} y_j \qquad (33)$$

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

$$\Psi = (G + \lambda N I)^{-1} \qquad (34)$$

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

$$G_{ij} = (K(x_i, x_j)).\ i, j = 1,...,N \qquad (35)$$

Or in matrix form:

$$A = (G + \lambda N I)^{-1} Y,\ A = (a_1, ...,\ a_N)^T \qquad (36)$$

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

Different types of kernels can be considered

$$Polynomial\ :\ K(x,x') = (1 + \langle x, x'\rangle)^{p_1} \qquad (37)$$

$$RBF\ :\ K(x,x') = \exp\left(-\frac{\|x - x'\|}{p_1}\right) \qquad (38)$$

$$ERBF\ :\ K(x,x') = \exp\left(-\sqrt{\frac{\|x - x'\|}{p_1}}\right) \qquad (39)$$

Where p_1 is a given parameter

2.2.5 RKHS MISO model

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

$$y(k) = \varphi\left[u_1, \ldots, u_p, k\right] + e(k) \quad (40)$$

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:

$$x = \left[u_1(k), \ldots, u_1(M_1 + k-1), \ldots, u_p(k), \ldots, u_p\left(M_p + k-1\right)\right]^{\mathrm{T}} \quad (41)$$

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.

3 Comparative study of both models

In this paragraph we are interested to compare Volterra and the RKHS models when a MISO process is considered.

3.1 Numerical example

Consider the system described by the following relation:

$$y(k) = 0.3u_1(k-1) + 0.2\,u_2^3(k-1) + e(k) \quad (42)$$

Where u_1 and u_2 are two gaussian inputs and *e* is a white noise. The provided results are issued after 20 Monte Carlo tries each contains 100 runs. The model performance is evaluated by using the Normalized Mean Square Error (NMSE) between the system output and the model one.

$$NMSE = \frac{\sum_{k=1}^{N} (y(k) - \tilde{y}(k))^2}{\sum_{k=1}^{N} (y(k))^2} \quad (43)$$

With N the observation number, y(k) is the output of the system and $\tilde{y}(k)$ is the model output.

3.1.1 Model Comparison

In Figure 2 we draw the process output and the output of the RKHS model using 100 samples for the learning set to estimate the parameters formulated by the representer theorem. The kernel used is polynomial with $p_1 = 3$. For model validation we use 120 samples other than those used for identification. It resorts that the model output fits the process output with a Normalized Mean Square Error (NMSE) equal to 0.48%.

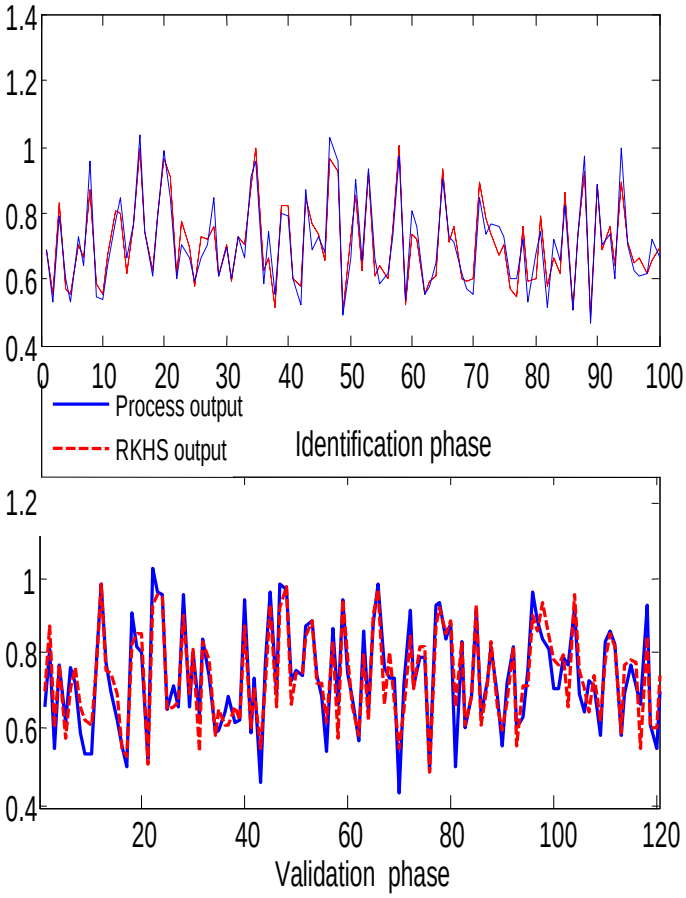

Fig.2: Validation of RKHS model; polynomial kernel

In Figure 3 we plot the Volterra model output and the process output for a non linearity degree P = 2 and a memory M = 2. We notice the concordance between both outputs with an NMSE equal to 0.51%.

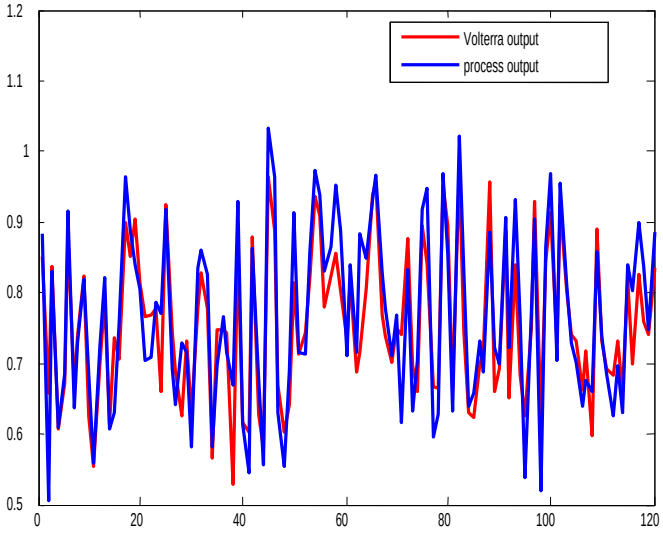

Fig.3: Validation of Volterra model (P = 2; M =2)

Table 1 summarizes the performances of both models such as the parameter number (np) the Normalized Mean Square Error (NMSE) and the computing time (CT).

Table 1: Performances of both models

Models	Tuning Parameters	np	NMSE(%)	CT (s)
RKHS model	polynomial Kernel (p1 = 3)	100	0.48	0.78
Volterra Model	P = 2 et M = 1	6	3	0.046
	P = 2 et M = 2	16	0.51	0.072
	P = 3 et M = 2	48	0.64	0.82

For the Volterra model three sets of structure parameters (degree of non linearity P and memory M) are considered. Even though the NMSE and the computing times are comparable for both models, the parameter number is smaller in case of Volterra model. We conclude that the complexity of the Volterra model increases with the memory M and the non linearity degree P which increases when a hard non linearity is considered, however the complexity of the RKHS model depends only on the number of observations used in learning step. Therefore the RKHS model is more efficiently for modelling hard non linearity.

3.1.2 Noise effect

To raise the influence of an additive noise on the identification quality we plot in Figures 4 and 5 the evolution of the NMSE for different SNR for the both models

Signal to Noise Ratio (SNR) is:

$$SNR = \frac{\sum_{k=0}^{N} (y(k) - \bar{y})^2}{\sum_{k=0}^{N} (e(k) - \bar{e})^2} \qquad (44)$$

With $\bar{y}$ and $\bar{e}$ are the mean values of the output and the noise respectively.

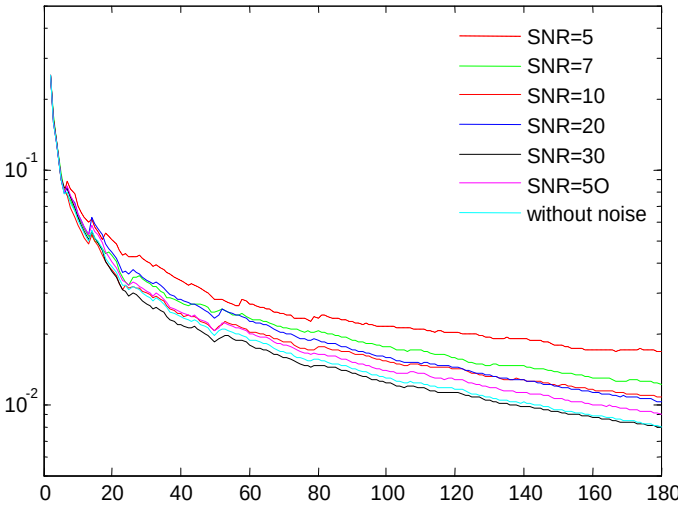

Fig.4: Noise effect on MISO Volterra model

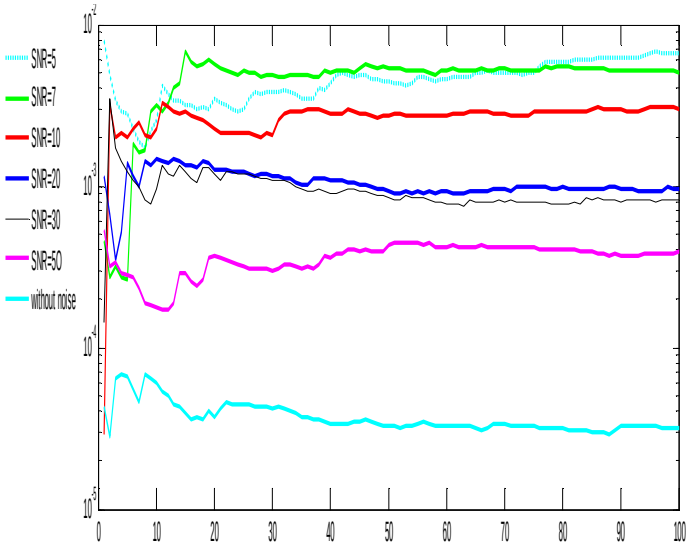

Fig.5: Noise effect on MISO RKHS model

It's noted that the error goes down when the SNR value goes high.

3.2 Chemical reactor modelling

3.2.1 Process description

To test the effectiveness of the RKHS and the Volterra models we test them on a Continuous Stirred Tank Reactor CSTR which is a nonlinear system used for the conduct of the chemical reactions [6]. A diagram of the reactor is given in the Figure 6.

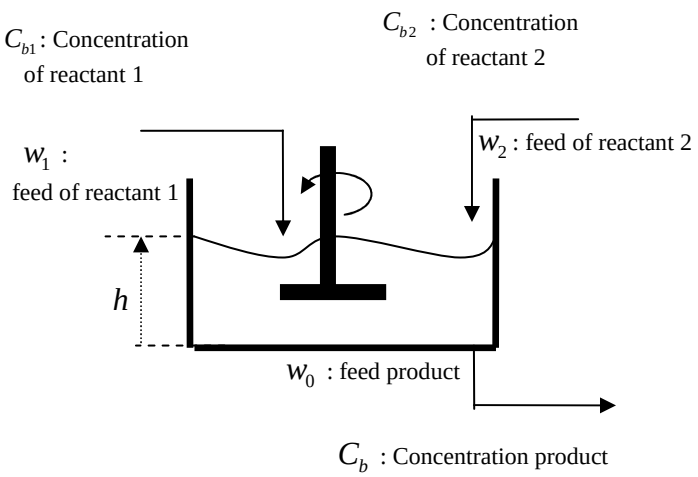

Fig. 6: Chemical reactor Diagram

The physical equations describing the process are:

$$\frac{dh(t)}{dt} = w_1(t) + w_2(t) - 0,2\sqrt{h(t)}$$

$$\frac{dC_b(t)}{dt} = \left(C_{b1} - C_b(t)\right)\frac{w_1(t)}{h(t)} + \left(C_{b2} - C_b(t)\right)\frac{w_2(t)}{h(t)} - \frac{k_1.C_b(t)}{\left(1+k_2.C_b(t)\right)^2}$$

Where $h(t)$ is the height of the mixture in the reactor, w_1 (resp, w_2) the feed of reactant 1(resp, reactant 2) with concentration Cb_1 (resp. Cb_2). The feed product of the reaction is w_0 and its concentration is C_b. k_1 and k_2 are consumption reactant rate. The temperature in the reactor is assumed constant and equal to the ambient temperature. We are interested by modelling the subsystem presented in Figure 7 where k_1, k_2, w_2 and Cb_2 are assumed to be constant so that it fits a MISO process with inputs the feed w_1 and the concentration C_{b1} of the reactant 1 and output the product concentration C_b

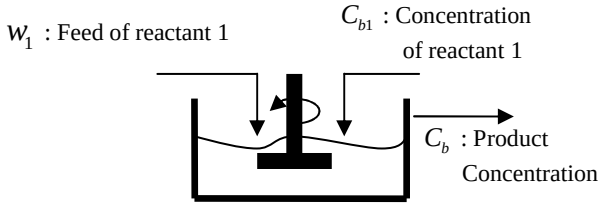

Fig. 7: Considered subsystem

For the purpose of the simulations we used the CSTR model of the reactor provided with Simulink of Matlab.

3.2.2 MISO RKHS model

We used the support vector machine (SVM) with the RBF kernel. The optimal parameters p_1, λ of this learning machine are obtained by a cross validation technique, $p_1 = 400$, $\lambda = 10^{-8}$. In the learning phase we use a training set of 100 inputs/outputs observations and in the validation phase we use 300 new observations to determine the performance of the RKHS model. In Figure 8, we plot the RKHS model output and the process output in the validation phase.

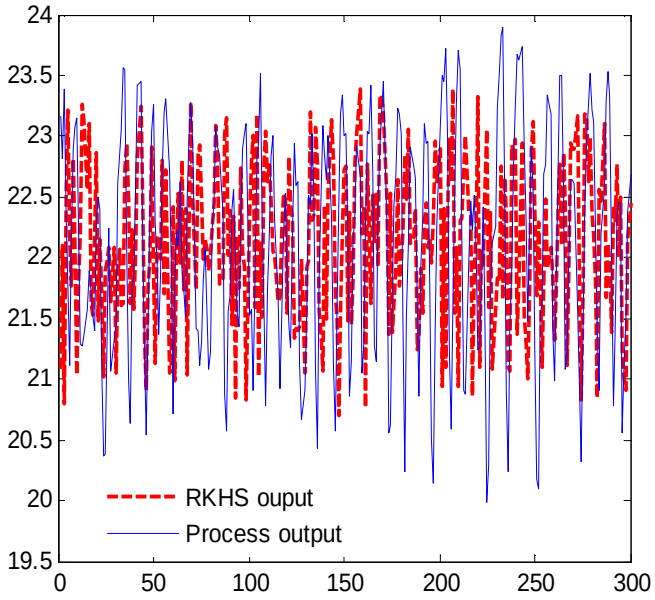

Fig.8: Validation of RKHS model

We notice the concordance between both outputs and the Normalised Mean Square Error (NMSE) is $7.848\ 10^{-3}$%

3.2.3 MISO Volterra model

This process can be modeled by a MISO Volterra model with degree of non linearity P = 2 and a memory M = 2, the input number is n = 2 and the parameter number is 16.

In Figure 9 we draw the validation of the model output; the yielded NMSE is 0.0485 %.

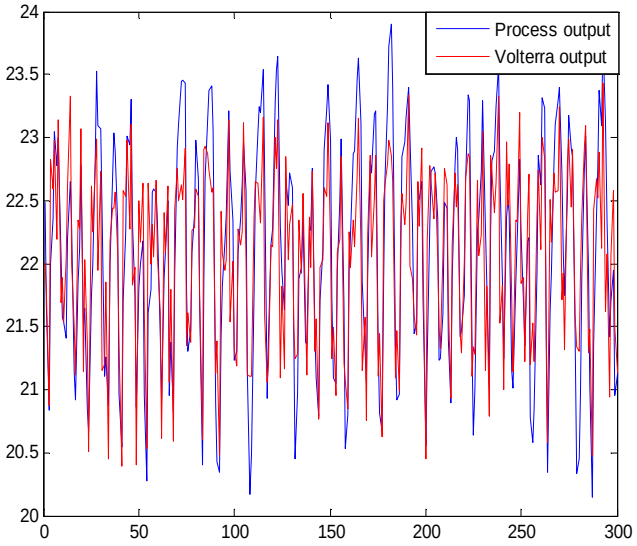

Fig. 9: Validation of MISO Volterra model P = 2 and M = 2

In Table 2 we evaluate the parameter number of the MISO Volterra model and of the MISO RKHS model. We conclude that MISO Volterra model is more efficient because it has the less number of parameters with a comparable NMSE.

Table 2: Performances of both models

Model	Model output	np	NMSE (%)
RKHS model	concentration	16	1.05
		100	0.00783
Volterra model	concentration	16	0.0485
		100	0.32

4 Conclusion

This paper has dealt with the study and the comparison of two non linear MISO system modelling techniques the Volterra model and the RKHS model. It has been shown that in its original form the complexity depends on the kind and on the hardness of the process non linearity. Monte Carlo simulations are carried out to evaluate performances of both models and the influence of an additive noise on the identification qualities. These models have been tested for modelling of a chemical reactor and results are successful

References:

[1] 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.

[2] A. Stenger, W. Kellermann, RLS-Adapted Polynomial Filter for Nonlinear Acoustic Echo Cancelling, *Proceedings of the X EUSIPCO* Tampere, Finland, 2000, pp. 1867-1870

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

[4] J.P .Vert, *Noyaux définis positifs*, Cours Master 2004/2005.

[5] J. Mercer, Functions of positive and negative type and their connection with the theory of integral equations, *Philosophical Transactions of the Royal Society*, London, A 1909, 209, 415–446,

[6] H. Demuth. M. Beale, M. Hagan, *Neural Network Toolbox 5, User's Guide*, The MathWorks, 2007.

[7] G. Wahba, *An introduction to model building with reproducing kernel Hilbert space*, Technical report N° 1020, Department of Statistics, University of Wisconsin, 2000.

[8] 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.

[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. Budura, C. Botoca, Efficient Implementation of the Third Order RLS Adaptive Volterra Filter, *Facta Universitatis Nis*, Ser.: Elec.Energ. vol.19, nr.1, April, 2006, pp. 133-141.

[11] G. Budura, C. Botoca, Practical considerations regarding the identification of nonlinear systems, *Revue roumaine des sciences techniques, serie Electrotechnique et energetique*, Tome 51, 1, 2006, pp. 79-90.

[12] G. Sunil, Performance Evaluation of an Adaptive Volterra/Hybrid Equalizer in a Nonlinear Magneto-Optic Data Storage Channel, *WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS*, Issue 4, Vol.2 October. 2003, pp. 832-835.

[13] L. Devroye, L. Gy¨orfi, G. Lugosi. A Probabilistic, *Theory of Pattern Recognition. Number 31 in Applications of mathematics.* Springer, New York, 1996.

[14] N. Aronszajn , Theory of reproducing Kernels, *Transactions of the American Mathematical Society*, Vol.68, 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] R. Rosipal and L.J. Trecho, Kernel Partial Least Squares in Reproducing Kernel Hilbert Spaces, *Journal of machine learning research* 2, 2001, pp. 97-123.

[17] S. Boyd, L.O. Chua, Fading memory and the problem of approximating nonlinear operators with Volterra series. I*EEE Tr. Circuits and Systems*, vol. CAS-32, No. 11, 1985, pp. 1150–1171.

[18] T.J. Dodd and R.F. Harrison. A new solution to Volterra series estimation. *Proceedings of the 15th IFAC World Congress on Automatic Control*, 2002, Barcelona, Spain.

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

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

[21] V.N. Vapnik, *The nature of Statistical Learning Theory*, Wiley New York., 1995.

[22] V.N. Vapnik, *Statistical Learning Theory*, Wiley New York., 1998

[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] V. J. Mathews, Adaptive Polynomial Filters, *Signal Processing Magazine*, Vol. 8, No. 3, 1991, pp. 10-26