Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 113–118

ISBN 978-83-60810-14-9
ISSN 1896-7094

Support Vector Machines with Composite Kernels for NonLinear systems Identification

Amina El Gonnouni, Abdelouahid Lyhyaoui
Engineering System
Lab.(LIS)
Abdelmalek Essaidi University
Tangier, Morocco
Email: amina_elgo@yahoo.fr, lyhyaoui@gmail.com

Soufiane El Jelali, Manel Martínez Ramón
Departamento de Teoria de
la Señal et Comunicaciones
Universidad Carlos III
De Madrid
Email: {soufiane, manel}@tsc.uc3m.es

***Abstract*—In this paper, a nonlinear system identification based on support vector machines (SVM) has been addressed. A family of SVM-ARMA models is presented in order to integrate the input and the output in the reproducing kernel Hilbert space (RKHS). The performances of the different SVM-ARMA formulations for system identification are illustrated with two systems and compared with the Least Square method.**

I. Introduction

SYSTEM identification treats the problem of constructing mathematical models from observed input and output data. Three basic entities must be taken into consideration to construct a model from data [1]:

1) Data: represent the input and the output data of the system;
2) Candidate models: are obtained by specifying within which collection of models the suitable one exists;
3) Identification method: determining the best model guided by the data.

In the literature of system identification, a large variety of nonlinear methods were used, such as neural networks, high order statistic and fuzzy system [2], [3], [4]. However these models have weaknesses. For example in neural network case, some problems appear, like slow convergence speed and local minima. Support Vector Machines (SVMs) overcomes these problems and seems to be a powerful technique for nonlinear systems where the required model complexity is difficult to estimate.

The Support Vector Machines (SVM) was originally proposed as an efficient method for pattern recognition and classification [3]. Then the technique became a general learning theory. The Support Vector regressor (SVR) was subsequently proposed as the SVM implementation for regression and function approximation [5]. SVM has been widely used to solve problems in text recognition, bioinformatics [6] and bioengineering or image processing [7] and these represent only a few of the practical applications of support vector machines. The key characteristic of SVM is that it maps the input space into a high dimensional feature space or a reproducing kernel Hilbert space through some nonlinear mapping, chosen a priori, in which the data can be separated by a linear function.

The autoregressive and moving average (ARMA) modelling is used when the candidate model is linear and time invariant. The explicit consideration of ARMA models in some reproducing kernel Hilbert space (RKHS) based on support vector machines (SVM-ARMA$_{2k}$) presents a new approach for identifications applications [9]. An analytical relationship between residuals and SVM-ARMA coefficients allows the linking of the fundamentals of SVM with several classical system identification methods. Additionally the effect of outliers can be cancelled [9]. By using the Mercer's Kernels trick, a general class of SVM-based nonlinear system identification can improve model flexibility by emphasizing the input-output cross information (SVM-ARMA$_{4k}$), which leads to straightforward and natural combinations of implicit and explicit ARMA models (SVR-ARMA$_{2k}$) and SVR-ARMA$_{4k}$) [10].

In this paper, we present the different SVM-ARMA models for the system used in [9] and for Bessel difference equation. We present the sensitivity of the SVM-ARMA models to the training data and to the noise power. Additionally, we compare our models with the least square method (LS) and we show each one's performance and the moment they exhibit the same results.

This work is structured as follows: we present the SVR algorithm for nonlinear system identification in section I. In section II, we summarize the explicit ARMA models in RKHS. Simulations and examples are included in section IV. Finally, in section V, we conclude the work.

II. SVR System Identification

Consider a nonlinear system whose input and output are DTP $\{x_n\}$ and $\{y_n\}$. Let $u_n = [u_n, u_{n-1}, \ldots, u_{n-Q+1}]$ and $y_{n-1} = [y_{n-1}, y_{n-2}, \ldots, y_{n-P}]$ represent the states of input and output DTP at instant n. The vector $z_n = [y_{n-1}^T, u_n^T]^T$ correspond to the concatenation of the two DTP at that instant *n*.

Giving a training set $\{z_i, y_i\}_{i=1}^N \in \Re^d$ with $d = P + Q - 1$. The linear regression model is:

$$y_n = \langle w, \phi_n(z_n)\rangle + e_n \; . \tag{1}$$

where $\phi(z_n) : \Re^P \times \Re^Q \to H_z$ represents the high dimensional feature space, or RKHS, which is nonlinearly mapped from the

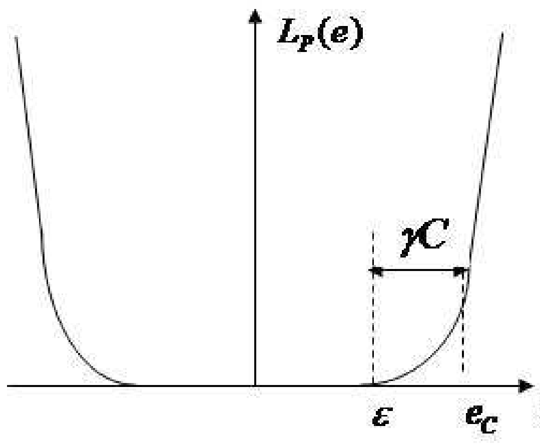

Fig. 1. ε-Huber cost Function.

input space,$\langle\cdot,\cdot\rangle$ represents the dot product and e_n denotes error terms, or residuals, comprehends both measurement and model approximation errors.

In SVR, several cost functions for residuals (CFR) have been used, such as Vapnik's loss function [3], Huber's robust cost [8] or the ridge regression approach [6]. However, in [9] they used ε-Huber CFR, which is a more general cost function that has the above-mentioned ones as particular cases. This cost function is depicted in 1 and it is expressed as:

$$l_p(e_n) = \begin{cases} 0, & |e_n| < \varepsilon \\ \frac{1}{2\gamma}(|e_n| - \varepsilon)^2, & \varepsilon < |e_n| < e_C \\ C(|e_n| - \varepsilon) - \frac{1}{2}\gamma C^2, & |e_n| > e_C \ . \end{cases} \tag{2}$$

where $e_C = \varepsilon + \gamma C$. The ε-Huber CFR can deal with different kinds of noise thanks to the three different intervals.

Using the ε-Huber CFR cost function, the algorithm of SVR system identification corresponds to the minimization of:

$$L_P = \frac{1}{2}\sum_{j=1}^{H_z} w_j^2 + \frac{1}{2\gamma}\sum_{n\in I_1}(\xi_n^2 + \xi_n^{*2}) + C\sum_{n\in I_2}(\xi_n + \xi_n^*) - C\sum_{n\in I_2}\frac{\gamma C^2}{2} \ . \tag{3}$$

with the constraints:

$$y_n - w^T\phi_n(z_n) \leq \varepsilon + \xi_n \qquad \forall n = n_0, \cdots, N \ . \tag{4}$$

$$-y_n + w^T\phi_n(z_n) \leq \varepsilon + \xi_n^* \qquad \forall n = n_0, \cdots, N \ . \tag{5}$$

where ξ_n,ξ_n^* are the slack variables or losses, $\xi_n^{(*)} \geq 0$ ($\xi_n^{(*)}$ represents both $\xi_n$and ξ_n^*), I_1 is set of samples for which $\varepsilon < \xi_n^{(*)} < e_C$, I_2 is the set of samples for which $\xi_n^{(*)} > e_C$, n_0 is given by the initial conditions and N is the number of available samples.

By introducing a nonnegative coefficient, Lagrange multiplier, for each constraint (α_n to (4) and α_n^* to (5)), we obtain the Lagrangian for this problem [6] this way:

$$\begin{aligned} L_{PD} = {} & \frac{1}{2}\sum_{j=1}^{H_z} w_j^2 + \frac{1}{2\gamma}\sum_{n\in I_1}(\xi_n^2 + \xi_n^{*2}) + C\sum_{n\in I_2}(\xi_n + \xi_n^*) \\ & -C\sum_{n\in I_2}\frac{\gamma C^2}{2} + \sum_{n=n_0}^{N}\alpha_n(y_n - w^T\phi_z(z_n) - \varepsilon - \xi_n) \\ & + \sum_{n=n_0}^{N}\alpha_n^*(-y_n + w^T\phi_z(z_n) - \varepsilon - \xi_n) \ . \end{aligned} \tag{6}$$

By minimizing the Lagrangian with respect to the primal variables w_j and $\xi_n^{(*)}$ we obtain:

$$w = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)\phi_z(z_n) = \sum_{n=1}^{N}\beta_n\phi_z(z_n) \ . \tag{7}$$

and $0 < \alpha_n^{(*)} < C$, where $\beta = \alpha_n - \alpha_n^*$

The dual problem is obtained by introducing (7) in (6) and it is expressed as:

$$L_D = -\frac{1}{2}(\alpha - \alpha^*)^T[G + \gamma I](\alpha - \alpha^*) + (\alpha - \alpha^*)^T y + \varepsilon 1^T(\alpha - \alpha^*) \ . \tag{8}$$

where G is gram matrix of dot product or kernel matrix with $G_{ij} = \langle\phi_z(z_i), \phi_z(z_j)\rangle = K_z(z_i, z_j)$, $\alpha_n^{(*)} = [\alpha_1^{(*)}, \ldots, \alpha_N^{(*)}]^T$ and $y = [y_1, \ldots, y_N]^T$. Finally the predicted output for a new observed sample y_r given $z_r$is:

$$\hat{y}_r = \sum_{n=1}^{N}\beta_n K_z(z_n, z_r) \ . \tag{9}$$

With the kernel function $K_z(z_n, z_r)$, we can deal with feature space of arbitrary dimension without having to compute the map $\phi_z$explicitly. Any function that satisfies Mercer's condition can be used as the kernel function [6]. The widely used Gaussian Mercer's kernel is given by $K_z(z_i, z_j) = exp(\frac{-\|z_i - z_j\|^2}{2\sigma^2})$, where σ^2 is the kernel parameter.

III. SVR System Identification and Composite Kernels

A family of composite kernels appears in SVM formulation by exploiting the direct sum of Hilbert spaces [10], which allow us to analyse the explicit form of ARMA process in feature space.

A. Explicit ARMA In Feature Space

By using two possibly different nonlinear mappings $\phi_n(u_n) : \Re^Q \rightarrow H_u$ and $\phi_y(y_n) : \Re^P \rightarrow H_y$, the input and output state vectors u_n and y_n can be separately mapped to RKHS H_x and H_y. So, an ARMA difference equation can be built using two linear models; MA (moving average) in H_x and AR (auto regressive) in H_y:

$$y_n = a^T\phi_n(y_{n-1}) + b^T\phi_n(u_n) + e_n \ . \tag{10}$$

where $a = [a_1, \ldots, a_{H_u}]^T$ and $b = [b_1, \ldots, b_{H_y}]^T$ are vectors representing the coefficients MA and AR of the system, respectively, in RKHS. After formulating the primal problem, stating the Lagrangian and making its gradient to zero, removed the primal variables and formulating the dual problem, the SVM-ARMA$_{2k}$ is obtained by including the kernel matrix $K(z_n, z_r) = K_y(y_{n-1}, y_{r-1}) + K_u(u_n, u_r)$ in (9) [10]:

$$\hat{y}_r = \sum_{n=1}^{N} \beta_n (K_y(y_{n-1}, y_{r-1}) + K_u(u_n, u_r)) \ . \tag{11}$$

where $K_y(y_{i-1}, y_{j-1}) = \langle \phi_y(y_{i-1}), \phi_y(y_{j-1}) \rangle$ and $K_u(u_i, u_j) = \langle \phi_u(u_i), \phi_u(u_j) \rangle$ are two different Gram matrices, one for the input and the other for the output.

B. Composite Kernels

The SVM-ARMA$_{2k}$ model could be limited in some cases, because (11) provides an apparent uncoupling between the input and the output. This limitation will be come out explicitly when strong cross information between the two DTP is present. An SVM-ARMA model considering the input and output could simultaneously solve this problem. By using the sum of Hilbert spaces property, the kernel components are:

$$\begin{aligned} K(z_i, z_j) = K_y(y_{i-1}, y_{j-1}) + K_x(u_i, u_j) \\ + K_{xy}(u'_i, y'_{j-1}) + K_{yx}(y'_{i-1}, u'_j) \ . \end{aligned} \tag{12}$$

When including this kernel in (9), we obtain the SVM-ARMA$_{4k}$ [10].

A new algorithm SVR-ARMA$_{2k}$ can be built by considering the combination between SVR and SVM-ARMA$_{2k}$

$$K(z_i, z_j) = K_y(y_{i-1}, y_{j-1}) + K_x(u_i, u_j) + K_z(z_i, z_j). \tag{13}$$

or an other one, SVR-ARMA$_{4k}$ by combining SVR and :

$$\begin{aligned} K(z_i, z_j) = K_y(y_{i-1}, y_{j-1}) + K_x(u_i, u_j) \\ + K_{xy}(u'_i, y'_{j-1}) + K_{yx}(y'_{i-1}, u'_j) + K_z(z_i, z_j). \end{aligned} \tag{14}$$

IV. Experimental Results

To examine the performance of SVM-ARMA formulations and to compare it with standard SVR and Least Square method, we use two examples. We focus on radial basis function (RBF) $K_z(z_i, z_j) = exp(\frac{-\|z_i - z_j\|^2}{2\sigma^2})$, where $\sigma^2 \in \Re$ represent the width of the kernel.

For the first example, the prediction performance is evaluated using the mean square error in test set:

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$

where N denotes the total number of data points in the test, y_i, $\hat{y}_i$ are the actual value and prediction value respectively

For the second example, we use the normalized mean square error in test set:

$$nMSE = \log_{10} \sqrt{\frac{MSE}{var(y)}}$$

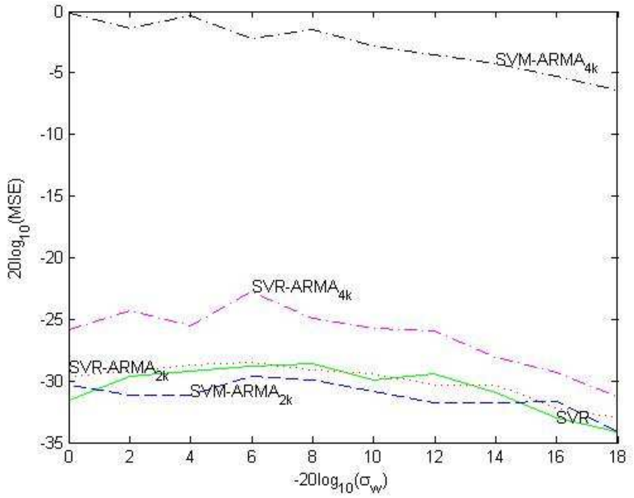

Fig. 2. The MSE as a function of additive noise of power σ_w.

To train our models, we use the cross validation method; 100 data are used as training data and 100 as testing data. For the first example, the results are averaged over 100 realizations and for example 2, over 200 realizations.

Example1: The first example of a system to be identified is [9]:

$$y_n = 0.03y_{n-1} - 0.01y_{n-2} + 3x_n - 0.5x_{n-1} + 0.2x_{n-2} \ . \tag{15}$$

The Input DTP is a white Gaussian noise sequence of unit variance $\{x_n\} \sim N(0, 1)$. An additive small variance random process, $\{e_n\} \sim N(0, 0.1)$, corrupts the corresponding output DTP and modelling the measurement errors. The observed process is $\{o_n\} = \{y_n\} + \{e_n\}$.

Impulsive noise $\{j_n\}$ is generated as a sparse sequence, for which 30% of the samples, randomly placed, are of high-amplitude, having the form $\pm 10 + U(0, 1)$, where $U()$ represents the uniform distribution in the given interval. The remaining are zero samples. The observations consist of DTP input $\{x_n\}$ and the observed output plus impulsive noise; $\{o_n\} + \sigma_w \{j_n\}$. Values of σ_w go from 18 to 0 dB [9].

We tried various values for ε, γ, C. For all the SVM-ARMA formulations, $\varepsilon = 0$ is used. In SVM-ARMA$_{4k}$, the values of SVM parameters that give the minimum MSE in testing set are like $C = 1$ and $\gamma = 0.01$, but for other SVM-ARMA models they are fixed in $C = 100$ and $\gamma = 0.001$. The results of our first system are shown in Figure 2, 3, 4, 5, 6. In Figure2, the SVM-ARMA$_{2k}$ model exhibit better performance, whereas SVM-ARMA$_{4k}$ and SVR-ARMA$_{4k}$ provide a poor model in terms of prediction error, that can be explained by the poor cross information between the input and output.

On the other hand, we compare the performance of SVM-ARMA models with the least square method, in which we use the same expression of the kernel components in each case (for example, in the case of SVR-ARMA$_{2k}$, the kernel components are like $K = K_x + K_y + K_z$ for SVM and LS methods). The results are shown in Fig.3-a, 4-a, 5-a, 6-a, 7-a, and they show

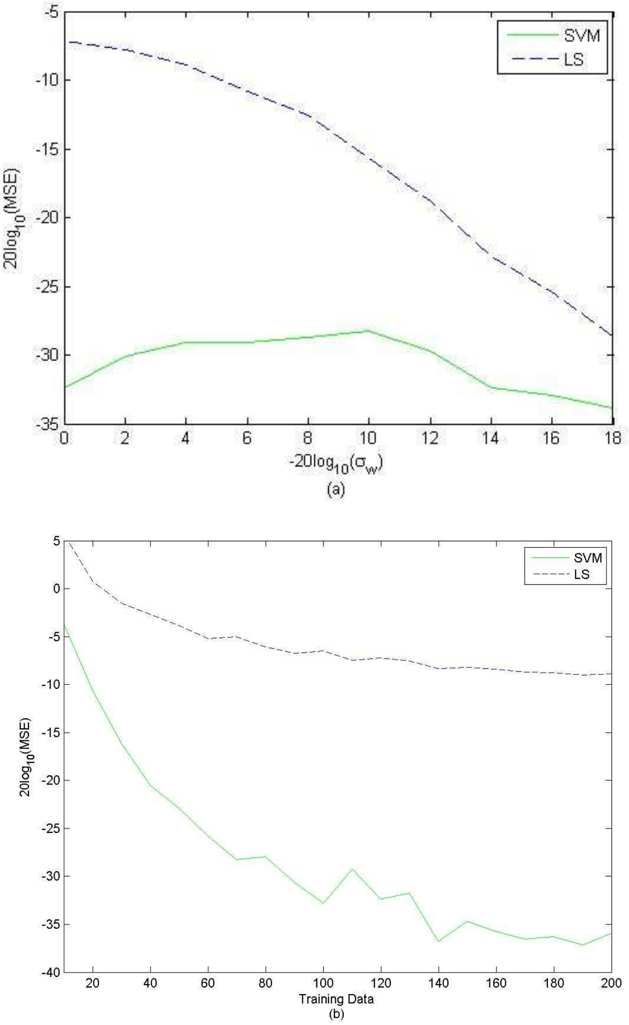

Fig. 3. (a) The MSE as a function of additive noise of power σ_w for SVR model. (b) The MSE as a function of training data for SVR model, where $\sigma_w = 1$.

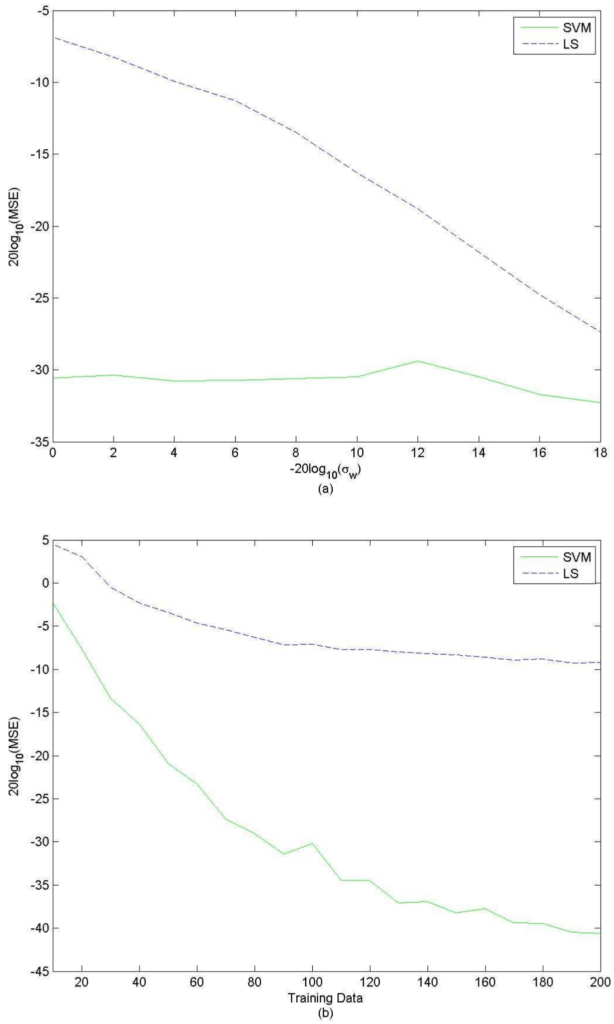

Fig. 4. (a) The MSE as a function of additive noise of power σ_w for SVM-ARMA$_{2k}$ model. (b) The MSE as a function of training data for SVM-ARMA$_{2k}$ model, where $\sigma_w = 1$.

that the SVM method exhibits a good performance in high impulsive noise power with a difference of almost 24 dB in comparison with LS method. Besides, SVM-ARMA methods show that there is no significant difference between the different values of MSE as a function of noise parameter, σ_w, which mean that the SVM-ARMA models, in this example, are not sensitive to the noise parameter σ_w.

Fig.3-b, 4-b, 5-b, 6-b, 7-b show that in the case of high impulsive noise power, $\sigma_w = 1$, the minimum MSE of SVM and LS methods are stabilized in affixed values even if the number of training data is augmented. We can say that the MSE is saturated. The SVM method needs 160 training data to saturate and LS requirements 100 data, but SVM gives a very small MSE in comparison with LS.

Example2: The second system to be identified is described by the difference equation of Bessel:

$$\left\{ \begin{array}{c} u(t) = 0.6\sin^{\alpha}(\pi t) + 0.3\sin(3\pi t) + 0.1\sin(\alpha t) \\ \tilde{y}(t+1) = \frac{\tilde{y}(t)\tilde{y}(t-1)[\tilde{y}(t)-2.5]}{1+\tilde{y}^2(t)+\tilde{y}^2(t-1)} \\ y(t+1) = \Re[\tilde{y}(t+1)] \end{array} \right. . \quad (16)$$

where $\Re[.]$ denotes the real part and α is a random variable uniformly distributed in the interval $[3, 4]$ with the mean $E\{\alpha\} = 3.5$.

For all the SVM-ARMA formulations, the SVM parameters, $\varepsilon = 0$, $C = 100$ and $\gamma = 0.01$ are used.

From Table 1, we notice that Bessel equation exhibits the best performance in SVM-ARMA$_{2k}$ and that the SVM-ARMA algorithms show the same results as the LS method.

Table 2 reports the nMSE of Bessel difference equation corrupted with additive Gaussian noise. The SVM formulations give the same value of nMSE, 0.003 dB, and the SVM method exhibits the same results as LS method.

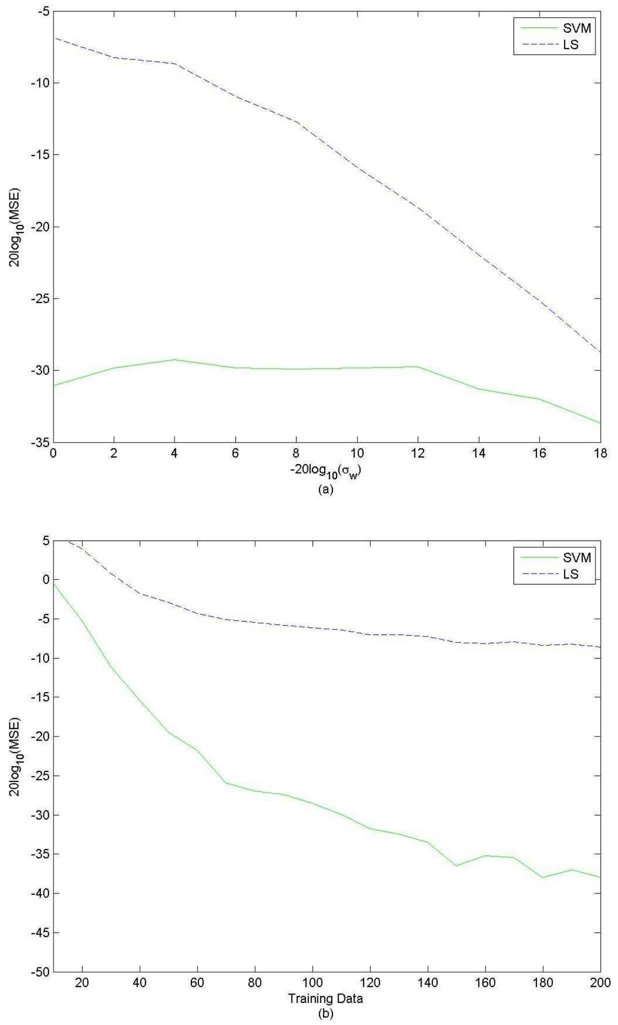

Fig. 5. (a) The MSE as a function of additive noise of power σ_w for SVR-ARMA$_{2k}$ model. (b) The MSE as a function of training data for SVR-ARMA$_{2k}$ model, where $\sigma_w = 1$.

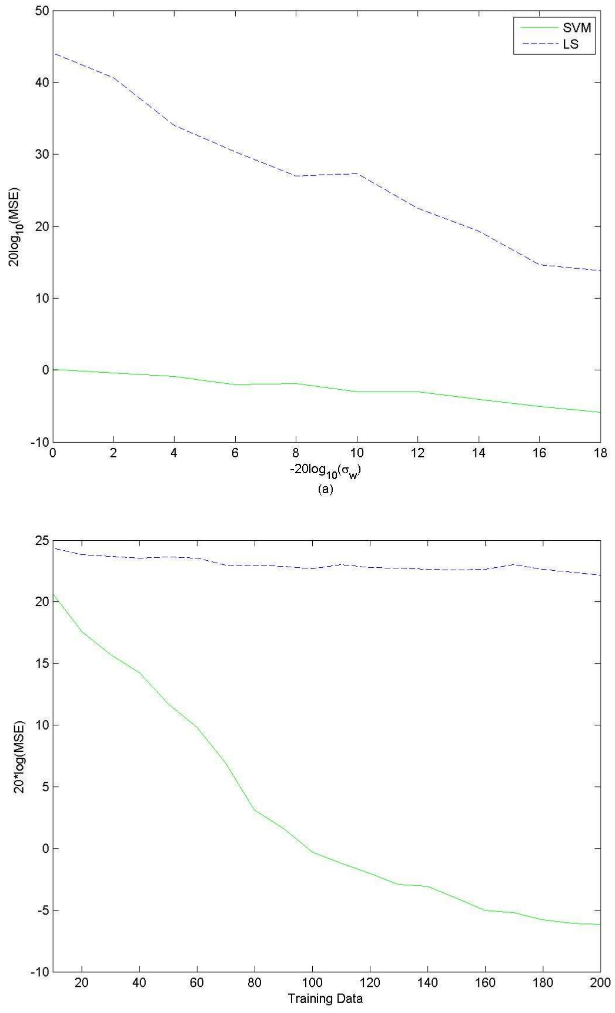

Fig. 6. (a) The MSE as a function of additive noise of power σ_w for SVM-ARMA$_{4k}$ model. (b) The MSE as a function of training data for SVM-ARMA$_{4k}$ model, where $\sigma_w = 1$.

TABLE I
THE NMSE OF BESSEL EQUATION.

		LS method
SVR	-0.927	-0.927
SVM-ARMA$_{2k}$	-0.935	-0.935
SVR-ARMA $_{2k}$	-1.045	-1.045
SVM-ARMA $_{4k}$	-1.246	-1.045
SVR-ARMA $_{4k}$	-1.262	-1.045

TABLE II
THE NMSE OF BESSEL EQUATION WITH GAUSSIAN NOISE.

		LS method
SVR	0.003	0.003
SVM-ARMA$_{2k}$	0.003	0.003
SVR-ARMA $_{2k}$	0.003	0.003
SVM-ARMA $_{4k}$	0.003	0.003
SVR-ARMA $_{4k}$	0.003	0.003

Therefore, we may conclude that SVM-ARMA methods provide as good results for Bessel difference equation as the best method LS, with and without additive Gaussian noise.

V. CONCLUSION

This paper has presented a full family of SVM-ARMA methods for nonlinear system identification in RKHS. These methods are proposed by taking the advantage of composite kernel, in which dedicated mappings are used for input, output and cross terms. Simulation results show the performance of the different SVM-ARMA models and compare it with the least square method.

REFERENCES

[1] L. Ljung, *System Identification. Theory for the User*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[2] J. Antari,R. Iqdour, S. Safi, A. Zeroual,A. Lyhyaoui, *Identification of Quadratic Non Linear Systems Using Higher Order Statistics and Fuzzy Models*, IEEE International Conference on Acoustics, Speech and Signal Processing. Proceedings of ICASSP 2006.

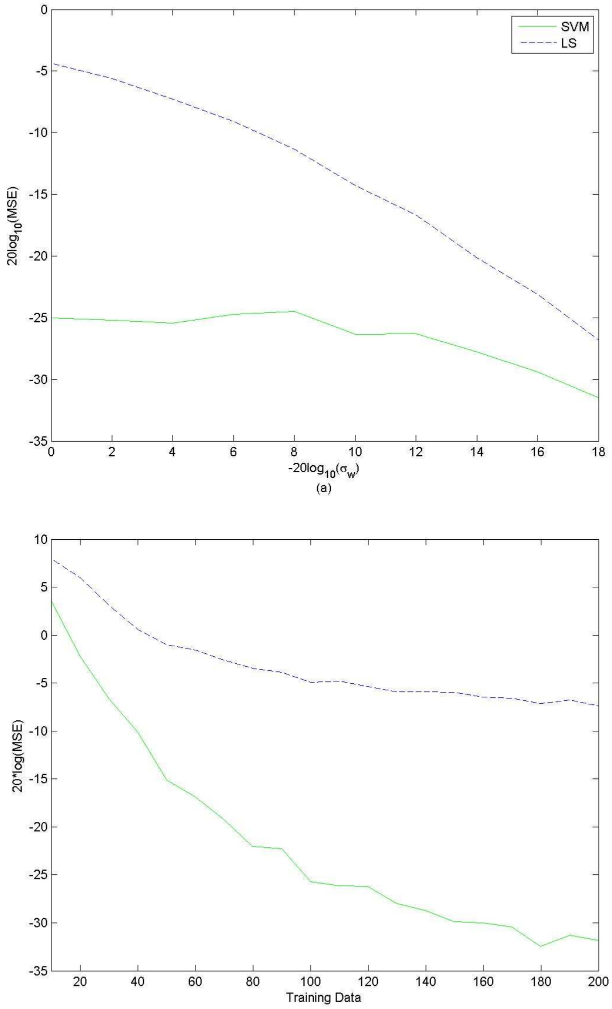

Fig. 7. (a) The MSE as a function of additive noise of power σ_w for SVR-ARMA$_{4k}$ model. (b) The MSE as a function of training data for SVR-ARMA$_{4k}$ model, where $\sigma_w = 1$.

[3] M. Ibnkahla, *Nonlinear System Identification Using Neural Networks Trained with Natural Gradient Descent*, EURASIP Journal on Applied Signal Processing 2003, pp.1229-1237.

[4] M. Kaneyoshi,H. Tanaka, M. Kamei,H. Farata, *New System Identification Technique Using Fuzzy Regression Analysis*, International Symposium on Uncertainty Modeling and Analysis, College Park, MD, USA, 1990, pp. 528-533.

[5] A.J. Smola,B. Schölkopf, *A Tutorial on Support Vector Regression*, ESPRIT, Neural computational theory. NeuroCOLT2 NC2-TR-1998-030, 1998.

[6] N.Cristianini, J.Shawe-taylor, *An Introduction to Support Vector Machines, and Other Kernel-Based Learning Methods*, Cambridge press, 2000.

[7] G. Camps-Valls, J.L. Rojo-Álvarez, M. MartínezRamón, *Kernel Methods in Bioengineering, Signal and Image Processing*, Hershey, PA: Idea Group Inc., 2006.

[8] K.R.Müller and al., *Predicting Time Series with Support Vector Machines. Articial Neural Networks*, ICANN'97, pages 999 - 1004, Berlin, 1997.

[9] J.L. Rojo-Álvarez, M. Martínez-Ramón, A.R. Figueiras-Vidal, M. de-Prado Cumplido, A. Artés-Rodriguez, *Support Vector Method for Robust ARMA System Identification*, IEEE Trans. Signal Process., vol. 52, no. 1, pp. 155-64, Jan. 2004.

[10] ¸M. Martínez-Ramón,J.L. Rojo-Álvarez,G. Camps-Valls, J. Muñoz-Marí, Á. Navia-Vázquez, E. Soria-Olivas,A.R. Figueiras-Vidal, *Support Vector Machines for Nonlinear Kernel ARMA System Identification*, IEEE Tans. Neural Networks, vol. 17, no. 6, pp 1617-1622, Nov. 2006.