

# Identification of an univariate function in a nonlinear dynamical model

B. David, G. Bastin

Center for Systems Engineering and Applied Mechanics,  
Université catholique de Louvain,  
Av. G. Lemaitre 4, B1348 Louvain-La-Neuve, Belgium.  
Phone: +32-10-472382, Fax: + 32-10-472180,  
Email: david@auto.ucl.ac.be

*Abstract*— This paper addresses the problem of estimating, from measurement data corrupted by highly correlated noise, the shape of an unknown scalar and univariate function hidden in a known phenomenological model of the system. The method makes use of the Vapnik’s support vector regression to find the structure of a parametrized black box model of the unknown function. Then the parameters of the black box model are identified using a maximum likelihood estimation method specially well suited to cope with correlated noise. The ability of the method to provide an accurate confidence bound for the unknown function is clearly illustrated from a simulation example.

*Keywords*— Hybrid Identification; Support Vector Machine; Nonlinear System; Correlated Residuals; Maximum Likelihood.

## I. INTRODUCTION

In [1], [2], [3], a maximum likelihood (ML) parameter estimation method has been developed for dynamical systems with auto and crosscorrelated measurement noise. This method allows to compute an accurate confidence region around the parameter estimate. The purpose of this paper is to extend this method to estimate an unknown function hidden in a phenomenological model and to provide a fair confidence bound on this function.

To achieve this objective, a linear expansion of basis functions is used as a parametrized black box model for the unknown function. In a first stage, the structure of the model (i.e. the number of terms in the linear expansion) and a preliminary value of the parameters are determined by using the support vector regression introduced by Vapnik [4], [5], [6] for function estimation problems. The so-called  $\nu$ -sv method [7] that results in a convex optimization problems is used to fit the black box model. This method allows to easily control the number of terms of the expansion and consequently allows to force a reduced number of parameters. This is essential for the second stage where the above mentioned ML parameter estimation method is applied to

the computation of refined parameter estimates and to the construction of the confidence bound on the unknown function.

A main contribution of the paper is to emphasize how critical the second stage is, in order to obtain an accurate estimation of the confidence region in case of auto and crosscorrelated measurement noise (which is the common situation in practice). The methodology is illustrated with a simulation example.

Compared to a brute force iterative identification procedure (involving both order selection and nonlinear parameter estimation), the method proposed in this paper has the advantage to decompose the identification procedure into several simple tasks which allow the user to better apprehend the relevance of the identified model.

## II. PROBLEM FORMULATION

The system under investigation is assumed to be exactly described by the following state-space equation:

$$\dot{x}(t) = f(x(t), u(t)) + F(x(t), u(t))\psi(x_r(t)), \quad (1)$$

where  $f$  and  $F$  are nonlinear functions of  $x(t)$  and  $u(t)$ ,  $x(t)=[x_1(t), \dots, x_n(t)]^T$  is the state vector,  $u(t)=[u_1(t), \dots, u_m(t)]^T$  is the input vector and  $\psi(x_r(t))$  is an unknown univariate scalar function of  $x_r(t)$ ,  $r$  is known.

The purpose is to identify a black box model and provide accurate confidence bounds for the unknown function  $\psi$  from a single noisy experiment performed on the system. It is assumed that all the state variables are measured, that the input signal  $u(t)$  and the initial state  $x(t_0)$  are known and that the measurements take place at evenly distributed time instants  $t_1, \dots, t_N$  where  $t_j = jT_s$ . The noisy state measurements are denoted  $z(j)=[z_1(j), \dots, z_n(j)]^T$  with:

$$z(j) = x(t_j) + v(j).$$

The noise sequence  $v(j)$  is assumed to be an auto and crosscorrelated zero mean stationary random vector sequence.

This paper presents research results of the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister’s Office for Science, Technology and Culture. The scientific responsibility rests with its authors.

### III. ESTIMATION METHOD

The estimation method is organized into two major stages. The goal of the first stage is to select a parametrized black box structure of the unknown function and to compute a first estimate of the parameter vector, regardless of statistical content of the noise. A linear radial basis function (RBF) expansion is identified using the support vector machine (SVM) method. The goal of the second stage is to refine the first estimate using a two-step ML estimation procedure, taking into account the correlation in the measurement noise. This second stage also provides a fair estimate of the covariance matrix of the parameter vector which is subsequently used to build confidence bounds around the estimated function.

#### A. Fitting the RBF model

Non parametric estimates  $\tilde{\psi}(t_j)$  of the unknown function  $\psi$  at the sampling instants are required to implement the SVM. Obviously, an explicit expression of  $\psi(t)$  as a function of  $x(t)$  and  $\dot{x}(t)$  can be derived from (1) but it would not be appropriate to just replace  $x(t)$  and  $\dot{x}(t)$  by the measurements  $z(j)$  and their (approximate) derivatives  $\dot{z}(j)$  to compute  $\tilde{\psi}(t_j)$  due to the noise amplification by the derivatives. Therefore, a cubic smoothing spline [8] is first fitted on the measured trajectories. The smoothed curve and its derivative are then used to compute the non parametric estimates. Let us therefore consider the cubic smoothing spline interpolant of  $x(t)$  in the piecewise polynomial form:

$$\tilde{x}_i(t) = \sum_{k=0}^3 \beta_{i,k}^{(j)} t^k, \quad (j-1)T_s \leq t \leq jT_s.$$

The  $4n(N-1)$  spline coefficients  $\beta_{i,k}^{(j)}$  are determined by minimizing:

$$\lambda \sum_{j=1}^N (z_i(j) - \tilde{x}_i(t_j))^2 + (1-\lambda) \int \ddot{\tilde{x}}_i^2,$$

where the factor  $\lambda$  ranges from 0 to 1 and allows to tune the smoothness of the spline.

Replacing  $x(t)$  and  $\dot{x}(t)$  by  $\tilde{x}(t)$  and  $\dot{\tilde{x}}(t)$  in (1), a least-squares estimate of  $\psi(t)$  can be computed at any time instant  $t$ :

$$\tilde{\psi}(t) = [F^T(\tilde{x}, u)F(\tilde{x}, u)]^{-1} F^T(\tilde{x}, u)(\dot{\tilde{x}} - f(\tilde{x}, u)). \quad (2)$$

Evaluating (2) at the sampling time instants yields the non parametric estimates of the unknown function and allows us to build the desired set of *input-output* pairs:

$$\{[z_r(j), \tilde{\psi}(t_j)], j = 1, \dots, N\} \quad (3)$$

that will in turn be used to fit the linear RBF expansion:

$$\Psi(x_r, \theta) = \sum_{k=1}^{p-1} \theta_k k(c_k, x_r) + \theta_p, \quad (4)$$

where the RBF are Gaussian kernels of the form:

$$k(c_k, x_r) = e^{-\frac{1}{2\sigma^2} \|c_k - x_r\|^2}.$$

The variance of the RBF,  $\sigma^2$ , is identical for all kernels and has to be fixed by the user. The number of kernels and their center  $c_k$  are *automatically* provided if the SVM method is used to find the parameter values  $\theta_k$ .

The classical method stated in [4] to fit an SVM on a set of *input-output* data solves a constrained optimization problem in its dual formulation. Loosely speaking, one could say that the method *automatically* select the most relevant centers  $c_k$  among all input data points and find the parameters  $\theta_k$  that minimizes a particular fitting criterion while ensuring a sufficient smoothness of the model. The centers  $c_k$  are called the support vectors in the SVM framework even though in the context of this paper these are scalar values since the unknown function is univariate. The SVM method presents two major advantages: It involves a convex optimization problem, more specifically a quadratic programming (QP) problem, and the model complexity (the number of kernels) does not need to be fixed a priori.

Among several alternatives to this method [9], the so-called  $\nu$ -sv method [7] presents an additional advantage that is particularly useful in the context of this paper. In the  $\nu$ -sv method, the QP problem is slightly modified by introducing a tuning coefficient  $\nu$  that is proven in [7] to be a lower bound on the fraction of support vectors among all possible candidates, that is all input values, ( $\nu \leq \frac{p-1}{N}$ ). This tuning coefficient gives a control on the number of parameters necessary to describe the function. This is of great interest here since a model with a reduced number of parameters is desired for the second stage.

Similarly to the classical Vapnik's method, the  $\nu$ -sv method solve the QP problem in its dual formulation. It takes the following form:

$$\begin{aligned} \max_{\alpha_j, \alpha_j^*} & \begin{cases} -\frac{1}{2} \sum_{\substack{j_1=1 \\ j_2=1}}^N (\alpha_{j_1} - \alpha_{j_1}^*)(\alpha_{j_2} - \alpha_{j_2}^*) k(z_r(j_1), z_r(j_2)) \\ + \sum_{j=1}^N \tilde{\psi}(t_j)(\alpha_j - \alpha_j^*) \end{cases} \\ \text{s.t.} & \begin{cases} \sum_{j=1}^N (\alpha_j - \alpha_j^*) = 0 \\ \sum_{j=1}^N (\alpha_j + \alpha_j^*) \leq C\nu N \\ \alpha_j, \alpha_j^* \in [0, C] \end{cases} \end{aligned} \quad (5)$$

This problem contains  $2N$  decision variables ( $\alpha_j, \alpha_j^*$ ) but its solution is typically sparse. That is, a large number of the variables are zero at the optimum. The parameters  $\theta_k$  ( $k = 1, \dots, p-1$ ) are then given by the non-zero  $\alpha_j^{(*)}$  and the centers of the RBF kernels  $c_k$  are the corresponding support vectors  $z_r(j)$ . We see that

the number of kernels (i.e. the number of non-zero  $\alpha_j^{(*)}$ ) and their centers are *automatically* obtained as a by-product of the solution of the QP problem.

The QP problem contains two tuning coefficients:  $\nu$  which has already been defined and  $C$  that determines the tradeoff between the smoothness of the model and the accuracy of the fitting. This coefficient is also an upper bound on the absolute value of each  $\alpha_j^{(*)}$ .

The last parameter  $\theta_p$  of (4) is given by:

$$\begin{aligned} \theta_p = & \frac{1}{2}[\tilde{\psi}(t_{j_1}) - \sum_{j=1}^N \alpha_j k(z_r(j), z_r(j_1)) \\ & + \tilde{\psi}(t_{j_2}) - \sum_{j=1}^N \alpha_j^* k(z_r(j), z_r(j_2))] \\ & \text{for any } j_1, j_2 \text{ such that } \alpha_{j_1}, \alpha_{j_2}^* \in (0, C). \end{aligned}$$

In this stage, the structure (number and position of the kernels) of the black box model has been identified based on the *input-output* values of the unknown function reconstructed using spline. The parameter vector  $\theta = [\theta_1, \dots, \theta_p]$  obtained will now serve as an initial value for the second stage.

### B. ML refining

From now on, we thus assume that the number  $p$  of kernels and the values of the centers  $c_k$  of the black box model (4) are fixed. Only the linear coefficients  $\theta_k$  are still modifiable. The model (1) is rewritten in its parametrized form:

$$\dot{x}(t, \theta) = f(x(t, \theta), u(t)) + F(x(t, \theta), u(t))\Psi(x_r(t, \theta), \theta). \quad (6)$$

In [3] an original ML parameter estimation method for nonlinear dynamical system was proposed for the case of highly correlated measurement noise. The method identifies the auto and the crosscorrelation features in the residual sequences and builds the inverse of the noise covariance matrix required for the ML estimation. The method is made of two steps that are summarized hereafter.

1— In the first step, a weighted least-squares (WLS) estimate of  $\theta$  is computed by minimizing the simulation error:

$$\hat{\theta}^{WLS} = \arg \min_{\theta} \sum_{j=1}^N (z(j) - x(t_j, \theta))^T \zeta (z(j) - x(t_j, \theta)), \quad (7)$$

where  $\zeta$  is a  $n$  by  $n$  diagonal weighting matrix used to normalize the residuals in order to balance the individual contribution of each state variable. The weights can be determined from a preliminary data analysis. For instance, one could take them inversely proportional to the variances of the state measurements.

Assuming that the resulting model  $\Psi(x_r, \hat{\theta}^{WLS})$  of the unknown function is *close enough* to the actual function, the residuals of this WLS estimation are expected to be close to the actual noise sequence:

$$w(j, \hat{\theta}^{WLS}) = z(j) - x(t_j, \hat{\theta}^{WLS}) \approx v(j).$$

2— In the second step, the residuals of the WLS estimate are used to identify the inverse of the covariance matrix of the noise sequence, then the ML estimate of  $\theta$  is computed.

Let us define the following stacked vectors:

$$\begin{aligned} \mathbf{x}(\theta) &= [x(t_1, \theta)^T, \dots, x(t_N, \theta)^T]^T \in \mathbb{R}^{nN}, \\ \mathbf{z} &= [z(1)^T, \dots, z(N)^T]^T \in \mathbb{R}^{nN}, \\ \mathbf{v} &= [v(1)^T, \dots, v(N)^T]^T \in \mathbb{R}^{nN}, \\ \mathbf{w}(\theta) &= \mathbf{z} - \mathbf{x}(\theta). \end{aligned}$$

The full covariance matrix,  $\Sigma = E\{\mathbf{v}\mathbf{v}^T\}$ , of the random vector  $\mathbf{v}$  is a very large block-Toeplitz matrix of the form:

$$\Sigma = \begin{bmatrix} \Sigma_0 & \Sigma_{-1} & & \Sigma_{1-N} \\ \Sigma_1 & \cdot & \cdot & \\ & \cdot & \cdot & \Sigma_{-1} \\ \Sigma_{N-1} & & \Sigma_1 & \Sigma_0 \end{bmatrix} \in \mathbb{R}^{nN \times nN},$$

where

$$\Sigma_k = E\{v(j)v(j-k)^T\} \in \mathbb{R}^{n \times n}$$

is the covariance matrix at lag  $k$  of the stationary random vector sequence  $v(j)$ .

Since the noise is supposed to be auto and crosscorrelated, the large matrix  $\Sigma$  is full. Using the approach developed in [3], an estimate of the inverse of the unknown matrix  $\Sigma$  is directly computed using  $\mathbf{w}(\hat{\theta}^{WLS})$ . This requires to identify a causal and an anticausal multidimensional AR model of the residuals, defined by the following relations:

$$\begin{cases} w(j, \hat{\theta}^{WLS}) + \sum_{l=1}^d A_l^+ w(j-l, \hat{\theta}^{WLS}) = e^+(j), \\ w(j, \hat{\theta}^{WLS}) + \sum_{l=1}^d A_l^- w(j+l, \hat{\theta}^{WLS}) = e^-(j), \end{cases}$$

where  $A_r^+$  and  $A_r^-$  are  $n$  by  $n$  matrices containing the coefficients of the causal and anticausal AR models,  $e^+(j)$  and  $e^-(j)$  are independently and identically distributed random vectors with covariance matrix equal respectively to  $\Sigma_{e^+} = E\{e^+(j)e^+(j)^T\}$  and  $\Sigma_{e^-} = E\{e^-(j)e^-(j)^T\}$ ,  $\forall j$ .

Once the coefficients of the AR models,  $A_l^{+,-}$  and  $\Sigma_{e^+,-}$ , are obtained, it is proven in [3] that an appropriate estimate of the inverse of the covariance matrix is given by the following explicit formula:

$$\Sigma^{-1} = \mathbf{U}^T \Sigma_{e^+}^{-1} \mathbf{U} - \mathbf{V}^T \Sigma_{e^-}^{-1} \mathbf{V}, \quad (8)$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are  $nN \times nN$  lower triangular block-Toeplitz matrices of the form:

$$\mathbf{U} = \begin{bmatrix} I_q & & & & \\ A_1^+ & \cdot & & & \\ & \cdot & \cdot & & \\ A_d^+ & \cdot & \cdot & \cdot & \\ 0 & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \\ 0 & 0 & A_d^+ & A_1^+ & I_q \end{bmatrix}, \mathbf{V} = \begin{bmatrix} 0 & & & & \\ & \cdot & & & \\ 0 & \cdot & & & \\ A_d^- & \cdot & \cdot & \cdot & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \\ A_1^- & & A_d^- & 0 & 0 \end{bmatrix},$$

while  $\Sigma_{e^+}^{-1}$  and  $\Sigma_{e^-}^{-1}$  are  $nN \times nN$  block-diagonal matrices of the form:

$$\Sigma_{e^+}^{-1} = \begin{bmatrix} \Sigma_{e^+}^{-1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \Sigma_{e^+}^{-1} \end{bmatrix}, \Sigma_{e^-}^{-1} = \begin{bmatrix} \Sigma_{e^-}^{-1} & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \Sigma_{e^-}^{-1} \end{bmatrix}.$$

Assuming furthermore a normal probability density function (PDF) for  $\mathbf{v}$ , the ML estimate of  $\theta$  for  $\Sigma^{-1}$  fixed is then given by, see e.g. [10]:

$$\hat{\theta}^{ML} = \arg \min_{\theta} \mathbf{w}(\theta)^T \Sigma^{-1} \mathbf{w}(\theta). \quad (9)$$

Note that the effective computation of the cost function in (9) for a given value of  $\theta$  does not require to form explicitly the large inverse covariance matrix. Indeed, it is shown in [3] that it reduces, due to the particular form of (8), to the application of appropriate filters to the sequence  $w(j, \hat{\theta}^{wLS})$ .

*Remark*— Both minimization problems (7,9) involved in this second stage are computationally very expensive because they require the integration of a dynamical system (6) at each iteration. The load of these minimization problems obviously increases with the number of parameters. This is the reason why it was important to force a rather small model order in the first stage. The minimization routine is initialized at the parameter value obtained in the first stage. The ML estimate is only guaranteed to be a local minimum of (9) near the initial value. For that reason we rather speak about a ML refining of the initial estimate. However this ML estimate will allow us to build accurate confidence bounds around the final estimate of the unknown function  $\Psi(x_r, \hat{\theta}^{ML})$ .

#### IV. CONFIDENCE BOUNDS

It is known that if the true system belongs to the model set (6), that is the selected black box structure (4) encompasses the true function, and if the noise covariance matrix is exactly known, the ML estimate converges asymptotically to a normal distribution with a covariance matrix given by the Cramér-Rao lower bound. In our case, the true function does likely not belong to the model set and moreover, the noise covariance matrix we use is only an estimate. Nevertheless, it is of common practice to use even though the Cramér-Rao bound as an approximate of the covariance matrix of the estimate. This bound is given by:

$$\Sigma_{\hat{\theta}^{ML}} \approx [\mathbf{G}(\hat{\theta}^{ML})^T \Sigma^{-1} \mathbf{G}(\hat{\theta}^{ML})]^{-1} \quad (10)$$

where the stacked state sensitivity matrix

$$\mathbf{G}(\theta) = [G(t_1, \theta)^T, \dots, G(t_N, \theta)^T]^T, \quad G(t_j, \theta) = \frac{\partial x(t_j, \theta)}{\partial \theta}$$

is obtained by the integration of the following matrix differential equation along with the system state equations, see e.g. [11]:

$$\frac{d}{dt} G(t, \theta) = \left[ \frac{\partial f}{\partial x^T} + \Psi \frac{\partial F}{\partial x^T} + F \frac{\partial \Psi}{\partial x^T} \right] G(t, \theta) + F \frac{\partial \Psi}{\partial \theta}.$$

Again, it is proven in [3] that the computation of (10) does not require to form explicitly the large inverse covariance matrix.

Assuming a normal PDF  $\mathcal{N}(\hat{\theta}^{ML}, \Sigma_{\hat{\theta}^{ML}})$  for the ML estimate, it becomes trivial to derive the PDF of any point of the estimated function  $\Psi(x_r, \hat{\theta}^{ML})$ . Since this function depends linearly on the parameters, it admits also a normal PDF given by:

$$\mathcal{N}(\Psi(x_r, \hat{\theta}^{ML}), R(x_r) \Sigma_{\hat{\theta}^{ML}} R(x_r)^T),$$

where

$$R(x_r) = \frac{\partial \Psi(x_r, \theta)}{\partial \theta} = [k(c_1, x_r), \dots, k(c_{p-1}, x_r), 1].$$

The 99% confidence interval is therefore given for any input point  $x_r$  by:

$$\Psi(x_r, \hat{\theta}^{ML}) \pm 2.58 \sqrt{R(x_r) \Sigma_{\hat{\theta}^{ML}} R(x_r)^T}.$$

#### V. SIMULATION EXAMPLE

Let us consider the following two-dimensional nonlinear state-space model:

$$\begin{cases} \dot{x}_1 = -x_1 - \psi(x_1) + u \\ \dot{x}_2 = -x_2 + \psi(x_1) \end{cases}$$

where the function  $\psi(x_1)$  has the following (supposed unknown) expression:

$$\psi(x_1) = 2 \frac{x_1}{1 + x_1^4}.$$

The state trajectories of this model are computed for a zero initial state. A double ramp command signal has been specially selected in order to ensure a rich enough coverage of the input space of  $\psi$ :

$$u(t) = 3.5 \left[ 1 - \frac{|t-5|}{5} \right].$$

The experiment lasts 10 seconds and the state trajectories are sampled with a frequency of  $20Hz$  such that  $N=200$  data samples are available for the identification. The measurement noise is generated by a first order bidimensional AR stochastic process with i.i.d. normal random innovation vectors. The exact structure of the process is:

$$\begin{cases} v(j) + \begin{bmatrix} -0.65 & -0.6 \\ -0.2 & -0.55 \end{bmatrix} v(j-1) = e(j), \\ \Sigma_e = 10^{-3} \begin{bmatrix} 0.625 & 0.0625 \\ 0.0625 & 0.1563 \end{bmatrix}. \end{cases}$$

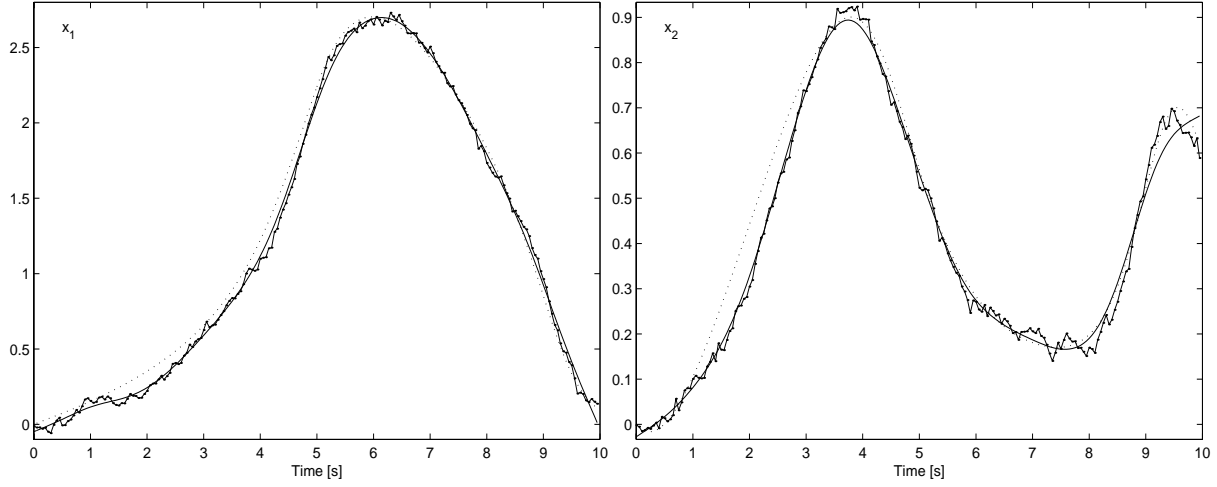


Fig. 1. State trajectories: the dotted lines represent the actual trajectories  $x_i(t_j)$ , the noisy lines represent the measured trajectories  $z_i(j)$  and the plain lines represent the measured trajectories smoothed by cubic splines  $\tilde{x}_i(t_j)$ .

The coefficients of this stochastic process have been scaled to produce highly auto and crosscorrelated noise sequences with signal-to-noise ratio of about 20dB on each state measurement.

Fig.1 illustrates the true state trajectories, the measured trajectories and the cubic spline interpolant with a smoothing factor of  $\lambda=0.95$ .

The size of the QP problem (5) involved in the fitting of the SVM is twice the number of data pairs. It can therefore become computationally too expensive as  $N$  increases. For large values of  $N$  it could be necessary to fit the SVM using only a subset of pairs. Here is an example of an iterative pruning procedure that could be applied:

0. Compute the mean distance  $l(j)$  of each pair relatively to its two neighbors:

$$l(j) = \frac{1}{2} \left[ \|(z_r(j), \tilde{\Psi}(t_j)) - (z_r(j-1), \tilde{\Psi}(t_{j-1}))\| + \|(z_r(j), \tilde{\Psi}(t_j)) - (z_r(j+1), \tilde{\Psi}(t_{j+1}))\| \right]$$

1. Find the pair with the minimum distance and remove it from the set,
2. update the mean distance of the two neighbors of the dropped pair according to their new neighbor,
3. return to 1. until the size of the remaining set is small enough.

The advantage of this pruning procedure is to preserve, as much as possible, the spatial coverage of the function. Only pairs that are close to other ones are removed, in an iterative way. In other words, redundant pairs are dropped.

We have applied this scheme in order to extract, from the set of 200 input and reconstructed output values, a subset of 100 pairs that are effectively used to fit the SVM. The QP problem has been solved using the following tuning coefficients:  $C=5$ ,  $\nu = 0.03$  and  $\sigma = 0.4$ . This yields six support vectors (six non-zero  $\alpha_j^{(*)}$ ) and thus seven parameters in  $\theta$ . The selected and dropped pairs are represented in Fig.2 along with the true function and the function estimated by the SVM. The position of the six support vectors are also represented. We can already see that this preliminary estimate of the unknown function is quite rough and must be refined.

The first step of the refining, that is the WLS estimate, results in a quite better shape for the unknown function as it is illustrated in the left plot of Fig.3. In this case the confidence bounds are computed using (10) with a diagonal  $\Sigma$  containing the weighting matrix factor  $\zeta$ . We see that in this case, the confidence bounds around the estimated function are quite underestimated. A large part of the true function lies outside

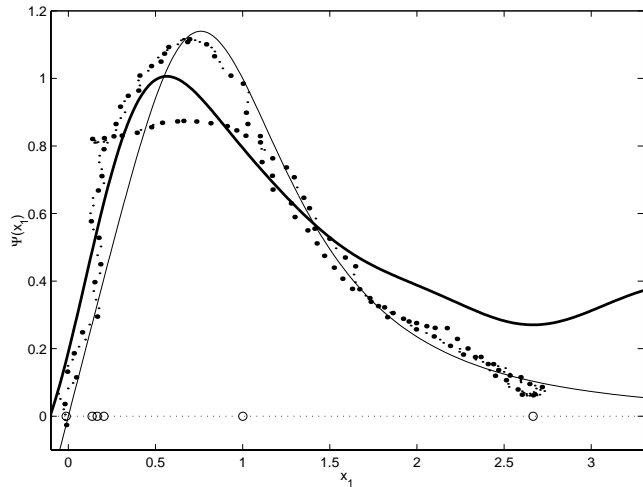


Fig. 2. First estimate of  $\Psi(x_r)$ : the thin line represents the actual function, the bold line represents the SVM estimate, the small dots are the 100 dropped pairs, the big dots are the 100 selected pairs and the circles on the base line point the support vectors.

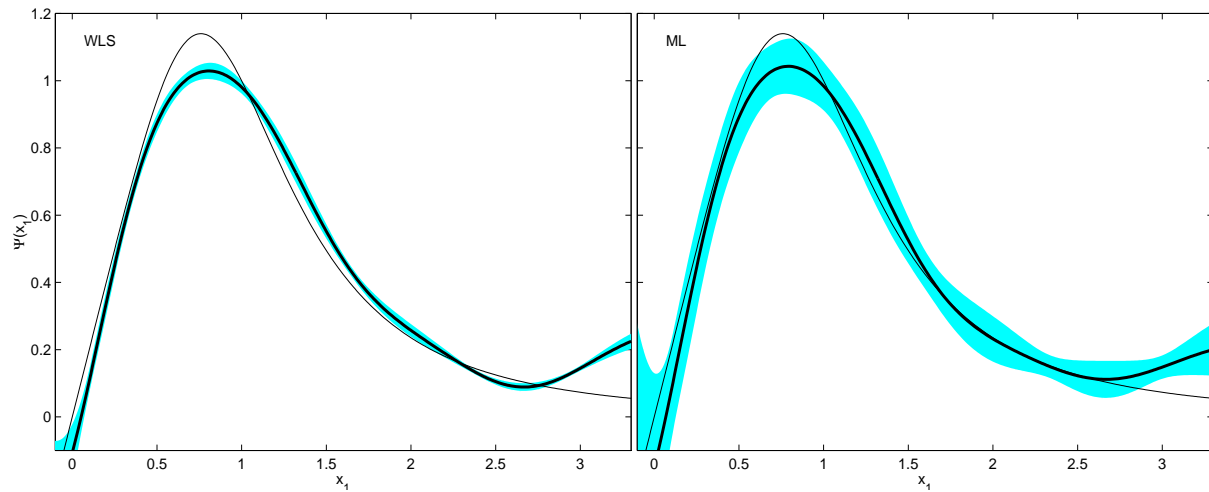


Fig. 3. Second (WLS) and third (ML) estimates of  $\Psi(x_r)$ : the thin line represents the actual function, the bold line represents the SVM estimate and the shaded area represents the 99% confidence bounds.

the confidence bounds. This results from the fact that the correlation features of the noise is still not taken into account in the WLS estimation step.

The second step of the refining, that is the ML estimate, results in a model of the function that is still closer to the true one. This last estimate is illustrated in the right plot of Fig.3. With respect to the WLS estimate, the improvement in terms of the function estimate is not really impressive. However, the confidence bounds obtained in this last step, using a full noise covariance matrix, are quite more realistic. They encompass the true function almost in the entire input area covered by the trajectory  $x_1$ . This is a result of taking into account the correlation in the noise.

The parameter values for the three estimates are given in Table I along with the support vectors.

## VI. CONCLUSION

A two stage method has been developed in this paper to identify an unknown function hidden in a known state space representation of the system. The method is particularly intended for the context of correlated measurement noise. The purpose of the first stage is to identify a black box structure for the unknown function using its input-output values reconstructed from the measurement data using spline. The purpose of the

second stage is to refine the model, using a maximum likelihood estimation method that captures the correlated features of the noise. An important message of this paper is to tell how it is essential to take into account the non whiteness of the noise to obtain a good estimate of the function and especially to build realistic confidence bounds.

## REFERENCES

- [1] B. David and G. Bastin, "An estimator of the inverse covariance matrix and its application to ML parameter estimation in dynamical systems," *Automatica*, to appear.
- [2] B. David and G. Bastin, "A maximum likelihood parameter estimation method for nonlinear dynamical systems," in *38th IEEE Conf. on Decision and Control*, Phoenix, Arizona USA, Dec. 1999, pp. 612–617.
- [3] B. David and G. Bastin, "Parameter estimation in nonlinear dynamical systems with auto and crosscorrelated noise," in *IFAC Symposium on System Identification (SYSID2000)*, Santa Barbara, California USA, June 2000.
- [4] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New-York, 1995.
- [5] V. Vapnik, *Statistical Learning Theory*, John Wiley, New-York, 1998.
- [6] V. Vapnik, "The support vector method of function estimation," in *Nonlinear Modeling: Advanced black-box techniques*, J.A.K. Suykens and J. Vandewalle, Eds., pp. 55–85. Kluwer Academic Publishers, Boston, 1998.
- [7] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson, "Support vector regression with automatic accuracy control," in *8th International Conference on Artificial Neural Networks*, L. Niklasson, M. Bodén, and T. Ziemke, Eds., Berlin, 1998, pp. 111–116, Springer Verlag.
- [8] C. de Boor, *A Practical Guide to Splines*, vol. 27 of *Applied Math. Sciences*, Springer Verlag, New-York, 1978.
- [9] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," NeuroCOLT2 Technical Report Series NC2-TR-1998-030, ESPRIT Working Group in Neural and Computational Learning II, Oct. 1998.
- [10] G.A.F. Seber and C.J. Wild, *Nonlinear Regression*, John Wiley & Sons, 1989.
- [11] E. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*, Springer, 1997.

TABLE I  
PARAMETER VALUES AND SUPPORT VECTORS.

	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$
First fit	-2.34	-5.00	0.25	-0.14	2.25	5.00	0.41
WLS step	-2.29	-5.11	0.59	-0.19	2.25	4.87	0.28
ML step	-2.33	-4.25	0.62	-0.12	2.15	4.17	0.24
sv's	-0.01	0.16	1.00	2.67	0.20	0.14	