THEORETICAL AND EXPERIMENTAL STATISTICAL IDENTIFICATION OF NONLINEAR AND DYNAMICAL MODEL USING CLPC ALGORITHM

Maurizio Melluso
University of Palermo (Italy)
Automation and Systems Center Research
Viale delle Scienze, 90128, Palermo
Email- maurizio.melluso@alice.it
maurizio.melluso@unipa.it

***Abstract*— A statistical method based on Cluster Last Principal Component (CLPC) algorithm to identify nonlinear and dynamical models from input-output data clusters of black boxes is presented. Each of data clusters is on a time window. For every data cluster an appraiser updates the parameters of a Gaussian time-varying model via an optimality design criterion that maximizes the Likelihood function. The estimated steady-state parameters of this model are quasi constant values. An experimental application to identify the nonlinear model of the angular position control system of a brushless motor is developed.**

***Keywords*— Clustering, Last principal components, model identification**

I. INTRODUCTION

The Principal Component Analysis has been the subject of considerable research effort over the last few years. In [1], [5] the Total Least Square (TLS) algorithm is proposed. In classical least square regression, errors are defined as the squared distance from the data points to the fitted function, as measured along a particular axis direction. Often, one has a data set in some large-dimensional space, but the actual data are closed to lying within a much smaller-dimensional subspace. In such cases, one would like to project the data into the small-dimensional space in order to analize them. Specifically a subspace that captures most of the summed squared vector lengths of the data must be determined. The axes for this low-dimensional space are known as the Principal Components of the data. Generally fundamental objective of Principal Component Analysis (PCA) [6],[7],[9],[10] is projecting a set of input output data of a system into a lower dimensional space that accurately characterizes the state of the process; but this method effectiveness is limited by its global linearity. In [10] and [14] a Multiway Principal Component Analysis (MPCA) is proposed for predictive diagnosis of a Wastewater Treatment plant. In [4], [11], PCA is formulated within a maximum-likelihood framework and the subspace is estimated in Maximum Likelihood sense using a probabilistic generative model. In [2], [3], [8] there are techniques of identification of nonlinear systems based on training and adaptive-training algorithms and using neural networks. In [9] is proposed a statistical estimate algorithm based on Last Principal Component (LPC) to estimate the nonlinear model of a planar manipulator. While TLS algorithm can be only applied to identification of static systems, LPC algorithm can be also applied to on-line identification of dynamical systems.

In this paper a new identification approach based on statistical Cluster LPC (CLPC) algorithm applied to estimate nonlinear models from input-output data of black boxes is developed. Since many process are highly nonlinear, one must to take knowledge about the same process and the operational behaviour of process parameters along the time. For this reason a clustering of the input output data has been taken into consideration where each of data clusters is on a window time. The main idea is that, particularly in motion problem, large data sets (for example angular positions of motors) can be nonlinear, but data in small clusters are sufficiently linear. In any case from the last principal component an information of linearity degree in a cluster can be obtained opportunely. This paper is organized as follows. In Section II the known TLS problem is presented and relation between this method and the LPC algorithm is developed. In Section III the details of the statistical CLPC algorithm will be pointed out. The statistical gaussian LPC model is developed and the problem of parameters estimation of this model in maximum likelihood sense is presented. The LPC model is similar to the ARMA model [12], [13]. However, in this paper a method of least squares is used to estimate the parameters of the model. In this case the solutions are the LPC filters coefficients, therefore the model is called “LPC model”. In section IV a generalization of the LPC problem to identify multi input multi output (MIMO) systems from a linear experimental data sets is presented. In section 5 a cluster LPC approach is adopted to identify nonlinear models. For every data cluster this algorithm updates the parameters of the LPC Gaussian model according to an optimality criterion that maximises the likelihood function and using an appraiser called “*finite state machine*”. The inputs of this appraiser are the data clusters of the black box, while the outputs are the time-varying parameters of the LPC gaussian filter. The problem of degree of linearity of the data in a cluster by means of LPC algorithm has been discussed and in particular the difference from the minimum and the maximum principal component is very important to value them. In Section V an application to identify the brushless motor model is presented. The inputs of the appraiser are input-output data clusters of an experimental control system of a brushless motor. The position commands for testing the CLPC algorithm have been implemented in C language using software, hardware and graphic interface of a brushless motor. Using the statistical CLPC algorithm, the actual angular position of the brushless motor and the non linear control torque have been estimated.

II. LPC VERSUS TLS ALGORITHM

Suppose one wants to fit the N dimensional data with a subspace (line/plane/hyperplane) of dimensionality N-1. The space is defined as containing all vectors orthogonal to a unit vector **u** and the optimization TLS problem can be expressed as:

$$\min_{\mathbf{u}} \|\mathbf{Mu}\|^2 \qquad (1)$$

where

$$\|\mathbf{u}\|^2 = 1$$

and **M** is a matrix containing data vectors in its rows. In particular let the Singular Value Decomposition as:

$$\mathbf{M} = \mathbf{USV}^T \qquad (2)$$

where **U** and **V** are orthogonal matrix and **S** is a diagonal matrix with positive decreasing element, it yields:

$$\|\mathbf{Mu}\|^2 = \mathbf{u}^{\mathbf{T}}\mathbf{M}^{\mathbf{T}}\mathbf{Mu} = \mathbf{u}^{\mathbf{T}}\mathbf{VS}^{\mathbf{T}}\mathbf{SV}^{\mathbf{T}}\mathbf{u} \qquad (3)$$

The problem (1) can be modified as follows:

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\mathbf{S}\boldsymbol{\alpha} \qquad (4)$$

where

$$\boldsymbol{\alpha} = \mathbf{V}^T\mathbf{u}, \quad \|\boldsymbol{\alpha}\| = 1$$

The matrix $\mathbf{S^TS}$ is a square and diagonal matrix with diagonal elements s_n^2 (singular values). Let s_N the smallest (last) value of s_n, it yields:

$$\boldsymbol{\alpha}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\mathbf{S}\boldsymbol{\alpha} = \sum_{\mathbf{n}} s_n^2\alpha_n^2 \geq s_N^2 \sum_n \alpha_n^2 = s_N^2\|\boldsymbol{\alpha}\|^2 = s_N^2 \qquad (5)$$

Therefore the solution of the problem (4) is:

$$\boldsymbol{\alpha}_{\mathbf{opt}} = [0 \quad 0.....0 \quad 1]^T = \mathbf{e_n} \qquad (6)$$

The solution (6) is the standard basis vector of *N*th axis and it can be transformed into the original coordinate system (cf. eq. 1) as follows:

$$\mathbf{u_{opt}} = \mathbf{V}\boldsymbol{\alpha}_{\mathbf{opt}} = \boldsymbol{\alpha}_{\mathbf{N}} \qquad (7)$$

where $\boldsymbol{\alpha}_{\mathbf{N}}$ is the *N*th column of **V** matrix. Therefore the solution of the problem (1) is the column of **V** matrix associated with the minimal singular value.

Total Least Squares (TLS) and Principal Components problems are often stated in terms of eigenvectors. The eigenvectors of a **S** square matrix are considered:

$$\mathbf{Sv} = \lambda\mathbf{v} \qquad (8)$$

where λ is the eigenvalue associated with the **v** eigenvector.

Considering the **M** matrix of data values and using the singular value decomposition (2), the total least square problem may be formulated as follows:

$$\min_{\mathbf{v}} \|\mathbf{Mv}\|^2 = \min_{\mathbf{v}} (\mathbf{v}^{\mathbf{T}}\mathbf{M}^{\mathbf{T}}\mathbf{Mv}) \qquad (9)$$

Therefore it yields:

$$\mathbf{v^TVS^TU^TUSV^Tv} = \mathbf{v}(\mathbf{VS^TSV^Tv}) \qquad (10)$$

If $\mathbf{v} = \mathbf{v_n}$ is the *n*th column of **V,** we have:

$$\mathbf{M^TMv_n} = (\mathbf{VS^TSV})\mathbf{v_n} = \\ = \mathbf{V}s_n^2\mathbf{e_n} = s_n^2\mathbf{v_n} \qquad (11)$$

We observe that $\mathbf{v_n}$ and $\lambda_n = s_n^2$ are the eigenvectors and the eigenvaules of the $\mathbf{M^TM}$ matrix respectively.

In summary the principal components are the eigenvector of the $\mathbf{M^TM}$ matrix. The Last Principal Component (LPC) is the eigenvector which corresponds to smallest eigenvalue and this solves the problem (9). In particular the correspondent eigenvectors constitute an hyperplane. This hyperplane minimizes the squared distances from each row of the **M** matrix to the same hyperplane. If there are eigenvalues equal to zero, the correspondent eigenvectors show linear relations between the data column of **M** matrix. In this approach the nonlinearities of the data are not considered.

III. LAST PRINCIPAL COMPONENTS ALGORITHM

The dynamical linear-in-the-parameters LPC model is given by:

$$\sum_{i=0}^{N} (\mathbf{Y}_n(n-i)+\boldsymbol{\varepsilon}_n(n-i))a_i + \sum_{i=0}^{N}(\mathbf{U_n}(n-i)+\xi_n(n-i))b_i = \mathbf{r}(n) \quad n \in Z \qquad (12)$$

where $\mathbf{Y_n}$ is the vector of nominal output data, $\boldsymbol{\varepsilon}_n$ is the output noise vector, $\mathbf{U_n}$ is the vector of nominal input data, ξ_n is input noise vector, $\boldsymbol{\varepsilon}_n$ is the output noise error, a_i and b_i are the parameters of the model, *N* is the order of the model and *n* is a discrete-time variable. The *r(n)* term is a random noise. Also is:

$$\mathbf{Y_n}(n)=[y_n(n), y_n(n-1),..., y_n(n-r)]^T$$
$$\mathbf{U_n}(n)=[u_n(n), u_n(n-1),..., u_n(n-r)]^T$$
$$\boldsymbol{\varepsilon}_n(n)=[\varepsilon_n(n), \varepsilon_n(n-1),..., \varepsilon_n(n-r)]^T$$
$$\xi_n(n)=[\xi_n(n), \xi_n(n-1),..., \xi_n(n-r)]^T \qquad (13)$$
$$\mathbf{r}(n)=\left[r(n)\ \ r(n-1)...r(n-r)\right]^T$$

From (12) and (13) we have a system of *N+1* unknown quantities and *r* equations. The unknown quantities are:

$$\mathbf{a}=\left[a_0\ \ a_1...a_N\right]$$
$$\mathbf{b}=[b_0\ \ b_1...b_N]$$

The model (12) can be written as follows:

$$\sum_{i=0}^{N} (\mathbf{Y}_m(n-i)a_i) + \sum_{i=0}^{N}(\mathbf{U}_m(n-i)b_i) = \mathbf{r}(n) \quad n \in Z \qquad (14)$$

where

$$\mathbf{Y}_m(n) = \mathbf{Y}_n(n) + \boldsymbol{\varepsilon}_n$$
$$\mathbf{U}_m(n) = \mathbf{U}_n(n) + \xi_n$$

are the measured values of the input-output data.

Now one equation of the system (12) is:

$$\sum_{i=0}^{N} (y_n(n-i)+\varepsilon_n(n-i))a_i + \sum_{i=0}^{N}(u_n(n-i)+\xi_n(n-i))b_i = r(n) \qquad (15)$$

The model (15) can be written as follows:

$$(\mathbf{y}_n(n)+\boldsymbol{\varepsilon}_n(n))^T\mathbf{a} + (\mathbf{u}_n(n)+\xi_n(n))^T\mathbf{b} = r(n) \qquad (16)$$

where

$$\mathbf{y_n}(n)=[y_n(n), y_n(n-1),..., y_n(n-N)]^T$$
$$\mathbf{u_n}(n)=[u_n(n), u_n(n-1),..., u_n(n-N)]^T$$

By using the following notation:

$$\mathbf{y}_m(n) = \mathbf{y}_n(n) + \boldsymbol{\varepsilon}_n$$
$$\mathbf{u}_m(n) = \mathbf{u}_n(n) + \xi_n$$

the model (15) can be written as:

$$\mathbf{y}_m^T(n)\mathbf{a} + \mathbf{u}_m^T(n)\mathbf{b} = r(n) \qquad (17)$$

The aim is to calculate the values of **a** and **b** parameters from the values of input-output data of a generical black box. It will be assumed that *r(n)* is a random variable with independent values and gaussian distribution as follows:

$$E[r(n)] = c; \ \ Var[r(n)] = \sigma_r^2$$

The *r(n)* gaussian distribution function is defined as [9]:

$$f[r(n)] = \frac{1}{\sqrt{2\pi\sigma_r}}\exp\left(-\frac{(r(n)-c)^2}{2\sigma_r^2}\right) = \frac{1}{\sqrt{2\pi\sigma_r}}\exp\left(-\frac{(\mathbf{y}_m^T(n)\mathbf{a}+\mathbf{u}_m^T(n)\mathbf{b}-c)^2}{2\sigma_r^2}\right)$$

This notation is used:

$$f(r(n)) = \frac{1}{\sqrt{2\pi\sigma_r}}\exp\left(-\frac{(\mathbf{z}_n^T\mathbf{l}-c)^2}{2\sigma_r^2}\right) \qquad (18)$$

where:

$$\mathbf{z}_n = \left[\mathbf{y_m}(n)|\mathbf{u_m}(n)\right] \qquad (19)$$

is the matrix of the measured input-output data and

$$\mathbf{l} = \left[\mathbf{a}\ \ \mathbf{b}\right]^T$$

is the vector of the parameters.

The probability function is defined as:

$$f[r(1), r(2),..., r(N)] = \prod_{n=1}^{N} f(r(n)) \qquad (20)$$

The logarithmic Likelihood function is given by:

$$\log\{f[r(1), r(2),..., r(N)]\} = \sum_{n=1}^{N}\left[-\log(2\pi) - \frac{1}{2}\log(\sigma_r^2) - \frac{(\mathbf{z}_n^T\mathbf{l}-c)^2}{2\sigma_r^2}\right] \qquad (21)$$

The estimation problem of **a** and **b** parameters can be formulated.

Problem. Consider a set of experimental data (19). The goal is to estimate **a** and **b** parameters of the model (15) to minimize the following performance index [9]:

$$J = \left[\frac{1}{2}N\log\sigma_r^2 + \sum_{n=1}^{N}\frac{(\mathbf{z}_n^T\mathbf{l}-c)^2}{2\sigma_r^2} - \frac{1}{2}\lambda(\mathbf{l^T l}-1)\right] \qquad (22)$$

where λ is the Lagrange multiplier.

Solution. Differentiating the performance index (22) by calculus of $\partial J / \partial c$ it results:

$$\frac{\partial J}{\partial c} = -2\frac{1}{2\sigma_r^2}\sum_{n=1}^{N}(\mathbf{z}_n^T\mathbf{l}-c) \qquad (23)$$

It yields:

$$\frac{\partial J}{\partial c} = 0 \rightarrow c = \frac{1}{N}\sum_{n=1}^{N}\mathbf{z}_n^T\mathbf{l} = \mathbf{Z}^T\mathbf{l} \qquad (24)$$

where

$$\mathbf{Z} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{z}_n \,. \qquad (25)$$

Substituting (24) into performance index (22), it yields:

$$J = \frac{1}{2}N\log\sigma_r^2 + \sum_{n=1}^{N}\left[\frac{\mathbf{z}_n^T\mathbf{l} - c}{2\sigma_r^2}\right] -$$

$$-\frac{1}{2}\lambda(\mathbf{l}^T\mathbf{l} - 1) = \qquad (26)$$

$$= \frac{1}{2}N\log\sigma_r^2 + \frac{1}{\sigma_r^2}\mathbf{l}^T\mathbf{A}\mathbf{l} - \frac{1}{2}\lambda(\mathbf{l}^T\mathbf{l} - 1)$$

where:

$$\mathbf{A} = \sum_{n=1}^{N}(\mathbf{z_n} - \mathbf{Z})(\mathbf{z}_n - \mathbf{Z})^T = \sum_{n=1}^{N}\overline{\mathbf{z}}_n\overline{\mathbf{z}}_n^T \qquad (27)$$

Minimizing (26) with respect to **l**, we can write:

$$\frac{\partial J}{\partial \mathbf{l}} = \frac{1}{\sigma_r^2}\mathbf{A}\mathbf{l} - \lambda\mathbf{l} = 0 \qquad (28)$$

From (28) it results:

$$\mathbf{A}\mathbf{l} = \lambda\sigma_r^2\mathbf{l} \qquad (29)$$

Now it is interesting to note that $\lambda\sigma_r^2$ is an eigenvalue of **A** matrix and **l** is the correspondent eigenvector.

All the eigenvectors of **A** are the Principal Component and we observe that **A** depends opportunely on the I/O data of a generic black box. Now the second derivative of performance index (26) can be calculated:

$$\frac{\partial^2 J}{\partial \mathbf{l}^2} = \frac{1}{\sigma_r^2}\mathbf{A} - \lambda\mathbf{I} \qquad (30)$$

where **I** is the identity matrix. To solve the problem, the (30) must be positive semidefinite. Therefore we have:

$$\forall \mathbf{x}:\quad \mathbf{x}^{\mathbf{T}}\mathbf{x} = 1 \rightarrow \mathbf{x}^{\mathbf{T}}\frac{\partial^2 J}{\partial \mathbf{l}^2}\mathbf{x} \geq 0 \qquad (31)$$

From (29) and (30) it yields:

$$\mathbf{x}^{\mathbf{T}}\mathbf{A}\mathbf{x} \geq \lambda\sigma_r^2 \qquad (32)$$

The (32) is validated if only if $\lambda\sigma_r^2$ is the smallest eigenvalue of the matrix given by (27). The eigenvector **l** which corresponds to smallest eigenvalue is the Last Principal Component. In this case the last principal component solves the mentioned problem. The parameters **a** and **b** of the LPC gaussian model are estimated and the model (17) is identified. In *z* domain the identified model (cf. eq. 17) is:

$$Y(z)\left[a_0 + \sum_{j=1}^{N}a_j z^{-j}\right] = U(z)\left[b_0 + \sum_{j=1}^{N}b_j z^{-j}\right] \qquad (33)$$

Therefore it yields:

$$\frac{Y}{U} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_N z^{-N}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_N z^{-N}} \qquad (34)$$

Note that the stability of the gaussian model depends on a_j parameters. The block diagram of the mathematical model (33) is shown in Fig. 1.

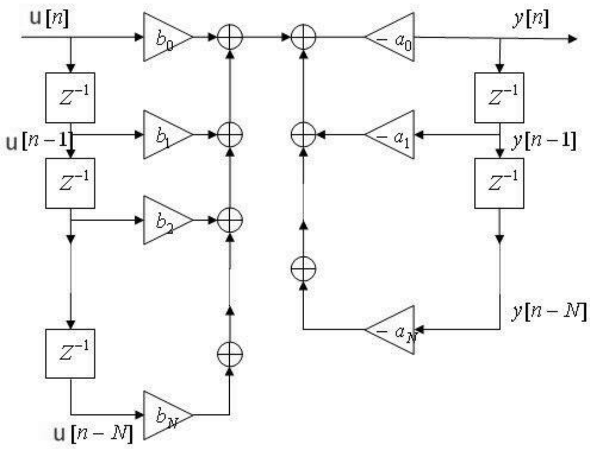

Fig. 1. Block diagram of an LPC gaussian model.

Remark 1. Referring to the above model (15) this remark is essential. The choice of *N* is very important in the process identification. In particular an iterative technique can be adopted and from N=1, the value of N can be increased to reduce the value of the index (22) as much as possible.

IV. LPC WITHOUT CLUSTERS ALGORITHM FOR MIMO MODELS

With reference to the multi-input multi-output (MIMO) black box of Fig. 2, this is a system with *q* measured input values $(\mathbf{U}_{mj}(n),\ j = 1 \ldots q)$ and *q* measured output values $(\mathbf{Y}_{mj}(n), j = 1 \ldots q)$. Note that there are *q* equations systems here (cf. eq. 12):

$$a_{oj}\mathbf{Y}_{mj}(n) + a_{1j}\mathbf{Y}_{mj}(n-1) + \ldots + a_{Nj}\mathbf{Y}_{mj}(n-N) =$$

$$-b_{oj}\mathbf{U}_{mj}(n) - b_{1j}\mathbf{U}_{mj}(n-1) - \ldots - b_{Nj}\mathbf{U}_{mj}(n-N) + \mathbf{r}(n)$$

$$j = 1 \ldots q \qquad (35)$$

Each of system (cf. eq. 14) is of *r* equations and *N+1* unknown quantities $(r \leq N)$, where *N* is the order of the identified model.

Now the steps of the statistical LPC algorithm for the identification model of a black box without cluster can be formulated with reference to all the equations of the system (35) and solution of the problem mentioned in previous subsection (cf. eq. 22).

1) define the input output data matrices of MIMO black box:

$$\mathbf{M}_{nj}^{T}=\begin{bmatrix} Y_{mj}(n) & \dots & Y_{mj}(n-N) & U_{mj}(n) & \dots & U_{mj}(n-N) \\ . & . & . & . & . & . \\ Y_{mj}(n-r) & \dots & Y_{mj}(n-r-N) & U_{mj}(n-r) & \dots & U_{mj}(n-r-N) \end{bmatrix}$$

$j=1....q$

(36)

2) compute mean values of the single lines of the $\mathbf{M}_{nj}$ matrices and define: $\mathbf{Z}_j = \frac{1}{N}\sum_{n=1}^{N}\mathbf{M}_{nj}$;

3) calculate the following matrices:

$$\mathbf{A}_j = \sum_{n=1}^{N}(\mathbf{M}_{nj}-\mathbf{Z}_j)(\mathbf{M}_{nj}-\mathbf{Z}_j)^T =$$

$$= \sum_{n=1}^{N}\overline{\mathbf{M}}_{nj}\overline{\mathbf{M}}_{nj}^{T} \quad (37)$$

$j=1...q$

4) calculate the minimum eigenvalues of $\mathbf{A}_j$ matrices (*j=1…q*).

From the eigenvectors the following parameters are achieved:

$$\mathbf{a}_j^T = [a_{0j}, a_{1j}, ..., a_{Nj}]$$

$$\mathbf{b}_j^T = [b_{0j}, b_{1j}, ..., b_{Nj}] \quad (38)$$

$j=1...q$

Remark 2. Referring to the above MIMO identification model method it is important to make the following remark. The model (35) is linear and, as it is well known, the last principal component indicates the possible linear relations between the input-output data of the $\mathbf{M}_{nj}$ (j=1…q) matrix . For non linear data set the above approach must be changed. For this reason in the next section a cluster approach is developed.

V. LPC WITH CLUSTER (CLPC) FOR MIMO NONLINEAR MODELS IDENTIFICATION

In this section a statistical Cluster LPC (CLPC) method to identify nonlinear models is presented. In fact the sets of the input output data from generical black boxes can be very large and non-linearities are possible. Therefore the input output data of the black box will be divided in clusters. In every cluster the data must be sufficiently linear.The problem of degree of linearity in a cluster is explained by means the LPC algorithm. Also, since the data are time varying, the clusters will be organized in time windows and for each window a finite state machine will update the parameters of a LPC dynamical gaussian model via an optimality design criterion that maximises the likelihood function of the data. So the steady state LPC gaussian model will be the identified model of the black box.

In many systems are in very large sets with substantial non-linearities. Therefore we cannot use the LPC model in a large sets of data (see Fig. 2.a). The linearization of the data (see Fig. 2.b) before to apply the previous steps is not convenient for approximation problems and we must apply a clustering approach (see Fig. 2.c). In this work each cluster of data is on opportunely window time. Therefore the CLPC algorithm is developed. Fig. 3 shows the block diagram of the statistical CLPC algorithm. For every window the parameters of the model (35) must be uploaded and an appraiser which estimates the $\mathbf{a}_j$ and $\mathbf{b}_j$ (*j=1…q*) parameters in every time is developed. This appraiser is called "*finite state machine*". A finite state machine is a system of discrete inputs-outputs and in particular is defined by a set of *inputs*, a set of *outputs*, a set of *states*, a set of *maps* from states and input into states and outputs, and an initial state. The inputs of the finite state machine are the input-output data clusters from the black box and the data are organized in the $\mathbf{M}_n$ matrix (cf. eq. 36) for every cluster. For each cluster the outputs are the time varying parameters (cf eq. 38) of the LPC gaussian mathematical model (cf. eq. 35). As it is well known the concept of *state* in control theory means capturing information about operation of the system in a set of variable. The state provides the task with information indicating what action is required at each scan. The parameters of the finite state machine of this work are as follows:

1) *parameters_sizes:* they are the sizes of $\mathbf{a}_j, \mathbf{b}_j$ (*j=1…q*) vectors;

2) *window_size:* it is the amplitude of the window-time for executing the LPC algorithm;

3) *data_ cluster_ dimension:* it is the updating interval of $\mathbf{a}_j, \mathbf{b}_j$ (*j=1…q*) values;

4) *sample_time:* it is the desired sample time.

The state components of the finite state machine are:

1) last window_sizes of input-output data for model identification;

2) counter;

3) last updating of $\mathbf{a}_j$ and $\mathbf{b}_j$ parameters.

In every time depending on choice of sample time the finite state machine performs the following tasks:

1) buffering of a new state vector from the previous state and from new input (state transition);

2) output calculus only from new state vector.

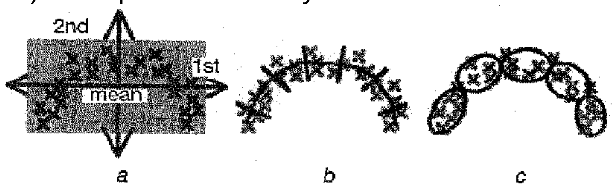

Fig. 2. (a) large set of data; (b) linearization of data; (c) clusters data approach.

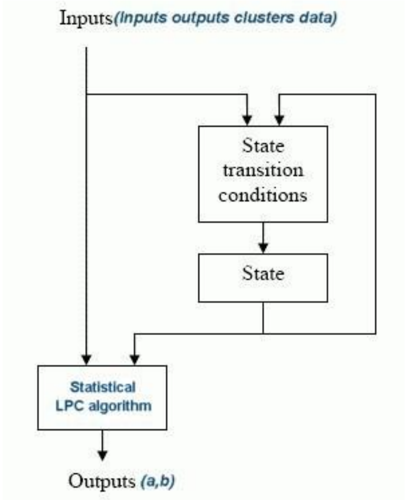

Fig. 3. CLPC algorithm.

Step by step the appraiser updates the state. The window of input-output samples is translated and there is an increment of the value of a counter. If this value is maximum (*data_cluster_dimension*), $\mathbf{a}_j, \mathbf{b}_j$ are updated using LPC which maximises the likelihood function (cf eqs. 21-22). In other words the CLPC algorithm is designed to be operated through repeated execution of statistical LPC algorithm depending on *data_clusters_ dimension* and on *window_size.* For every cluster the minimum or null eigenvalue computed by the fourth step of the statistical LPC algorithm of the previous subsection indicates possible linear relations between the data of the cluster. In particular from LPC algorithm applied to data of a cluster, if there is many difference between the maximum and the minimum eigenvalue, then linear relations between the data of the cluster are guaranteed. In fact, if the minimum eigenvalue is very small, then the disequality (32) is strictly verified and the likelihood function is maximum. We observe that the new state vector of the finite state machine depends on the previous state and on the new input. Therefore in this case the algorithm converges after a few clusters analysis if there is many difference between the maximum and minimum eigenvalue and this difference is quasi constant for each remaining cluster. So the outputs of the CLPC algorithm are quasi constant after a few clusters analysis. The choice of the *data_ cluster_ dimension* must guarantee the convergence of the algorithm and it depends on the nonlinearity degree of the data. The problem of the convergence is shown in the simulation experiments of the next section.

VI. APPLICATION OF THE CLPC ALGORITHM TO THE MODEL OF THE CONTROL SYSTEM OF A BRUSHLESS MOTOR.

The control system of the brushless motor consists of a power amplifier, resolver interface, and digital motor control circuits. This provides everything that is needed to control the motor's torque, velocity or position. Fig. 4 shows the servo block diagram of the motor in position control mode. There are a position loop proportional controller (*position gain*), a velocity loop proportional controller (*velocity gain*), a velocity loop integrator (*velocity integrator*). There are low-pass filters and notch filter to reduce mechanical vibrations. The input and output signals of the control system are the position command and the position signal respetively. The input-output of the motor are the torque and the position signal respectively.

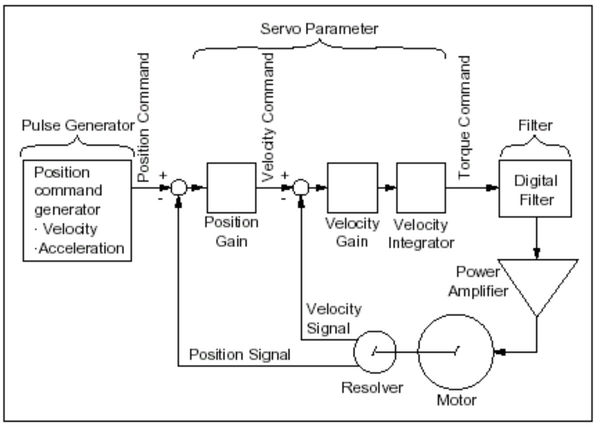

Fig. 4. Control position system of the brushless

A single-input, single-output (SISO) system is presented. Therefore the index *j* (cf. eq. 36) is equal to *j=1*. The input data $\mathbf{U}_m$ of the matrix $\mathbf{M}_n$ (cf. eqs. 35-36) are the position commands in every time (see Fig. 5), while the output data $\mathbf{Y}_m$ (cf. eqs. 35-36) are the position signals in every time. As regards the cluster approach, Table 1 reasumes the parameters of the CLPC algorithm.

Sample Time	10 ms
Parameter_size	3
Window_size	201 samples
Window_time	2 s
Data_ cluster_dimension	12 samples

Table 1. CLPC parameters.

Since the *parameter_size* is equal to 3 and *j=1*, it yields (cf. eq. 38):

$$\mathbf{a}_1^T = \left[a_{01}, a_{11}, a_{2j}\right] \quad \mathbf{b}_1^T = \left[b_{01}, b_{11}, b_{21}\right] \qquad (39)$$

In this case is *N=2.* A specific type of position command is used: ramp with saturation to implement the nonlinearity (see Fig. 5). This signal is opportunely implemented using hardware and software of the brushless motor. For each data cluster, Fig. 6 summarises the difference *D* between the maximum and the minimum eigenvalue. Fig. 6 shows that *D* increseases for each cluster data analysis and after a few clusters analysis it is quasi constant.

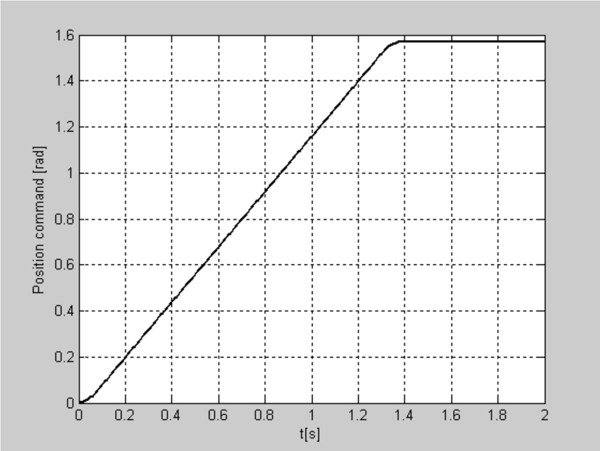

Fig. 5. Position command [rad].

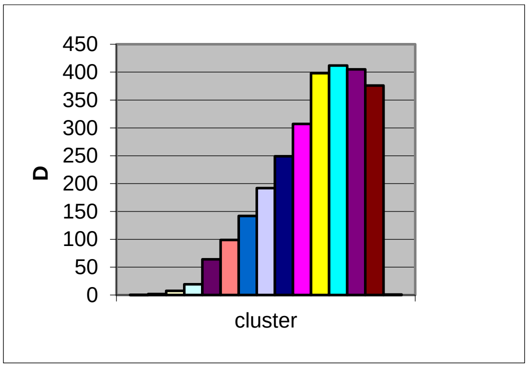

Fig. 6. D for each cluster.

Figs. 7-12 show the outputs of the CLPC algorithm Since after a few clusters analysis the parameter *D* is high and it is quasi constant for each remaining cluster, the parameters (39) are quasi constant after a few clusters data analysis.

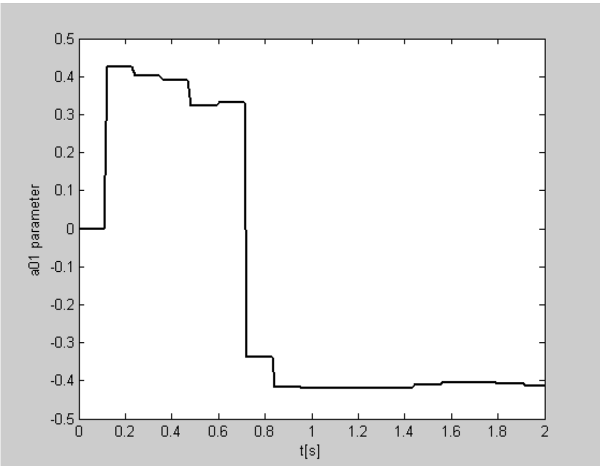

Fig. 7. a_{01} parameter.

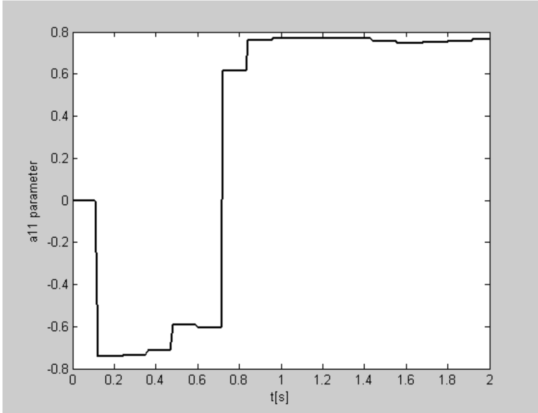

Fig. 8. a_{11} parameter.

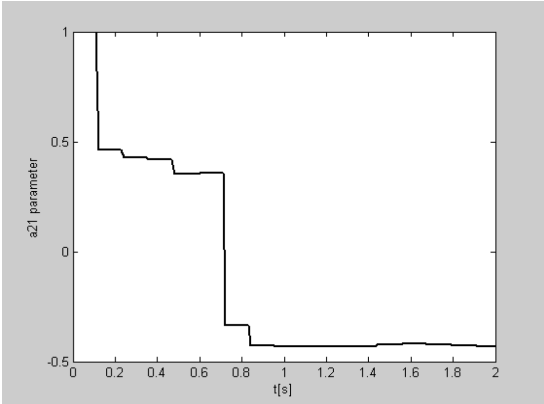

Fig. 9 a_{21} parameter

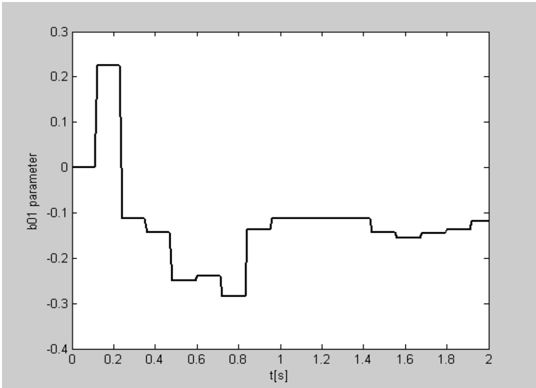

Fig. 10. b_{01} parameter.

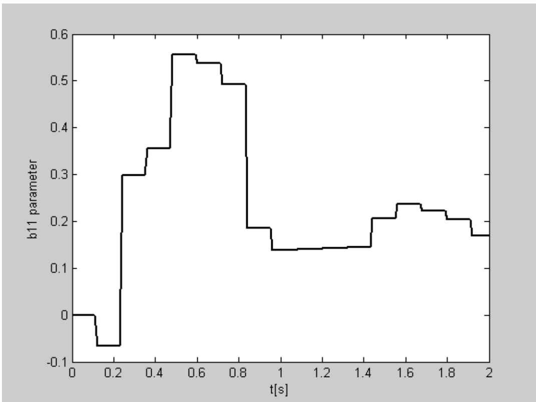

Fig. 11. b_{11} parameter.

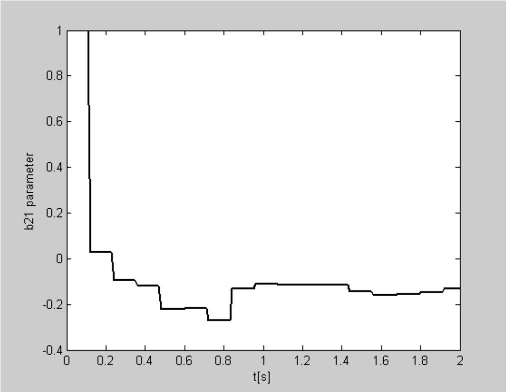

Fig. 12. b_{21} parameter.

As regards the brushless motor as actuator of the control system shown in Fig. 5, the inputs are the torque commands while the outputs are the position signals. Using the quasi constant outputs of the CLPC algorithm, the input-output model of the control system has been identified. Figs 13-16 show the good efficiency of the CLPC identification algorithm. In particular in Figs. 13 and 14 the experimental and estimated torque commands and the identification error of the control torque as difference between the experimental and estimated control torques have been shown.

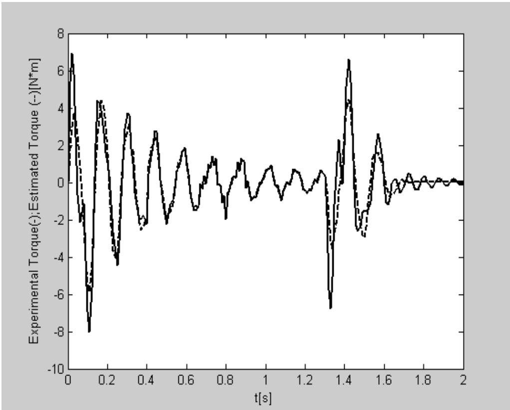

Fig. 13 Experimental torque command (-), estimated torque command (--) [Nm]

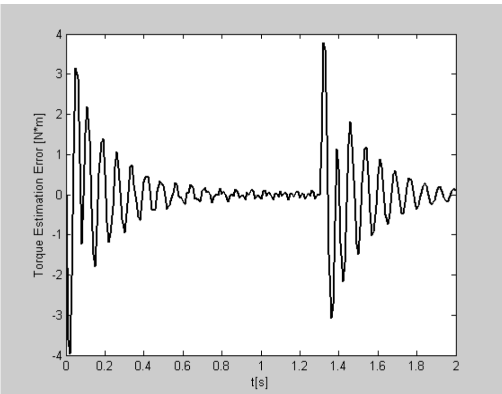

Fig. 14 Torque Command identification error

In Figs. 15, 16 and 17 the experimental values of the position signal, the estimated position signal and the identification error as difference between the experimental and estimated values of the same signal have been sketched respectively.

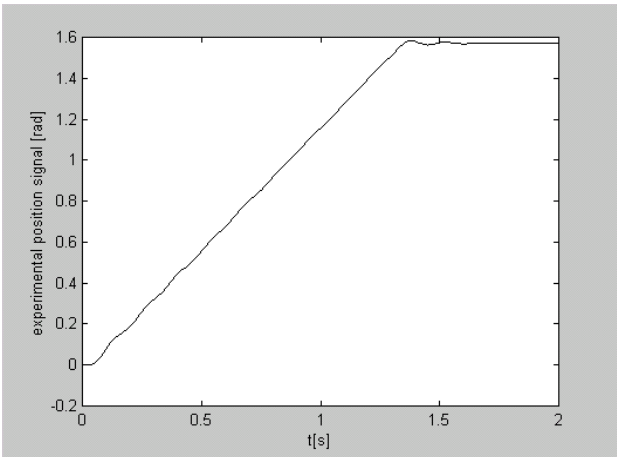

Fig. 15 Experimental angular position of the brushless motor

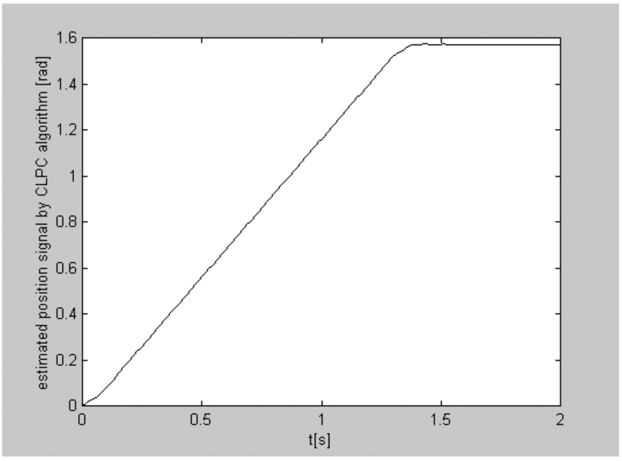

Fig. 16 Estimated angular position signal of the brushless motor

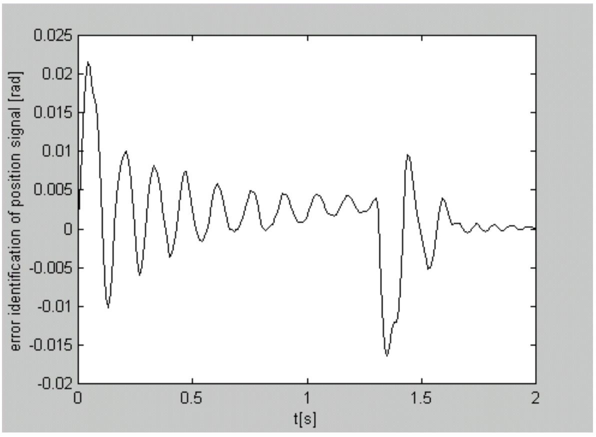

Fig. 17 Error identification of the position signal

Since the *N* order of the CLPC algorithm is equal to 2, the quadratic nonlinearity are well tracked at the most. Therefore from Figs. 15-17 we observe a large identification error at t=1.4s. However the stability of the estimated model by using of the CLPC algorithm ensures the convergence to zero of the identification errors.

Conclusions

An algorithm of identification method for nonlinear SISO and MIMO model based on CLPC algorithm from experimental input-output data of a black box has been developed in this work. The approach is based on statistical LPC algorithm. The LPC algorithm is formulated within a maximum likelihood framework using a Gaussian varying time model. But using the LPC approach, the data input-output of the black box must be sufficiently linear. Therefore a cluster approach of the LPC algorithm is proposed. The global set of the input-output data can be nonlinear, but in a cluster the data are sufficiently linear. From the difference between the minimum and the maximum principal component, the degree of linearity of the data in a cluster can be evaluated. The CLPC algorithm is designed to be operated through repeated execution of statistical LPC algorithm depending on parameters of a finite state machine. The LPC algorithm is executed for every data cluster and the parameters are uploaded by using the finite state machine, where the new state vector depends on previous state and on new input. The convergency of the algorithm is ensured if, after a few clusters analysis, the difference between the maximum and the minimum eigenvalue is high (this assure the linearity) and it is quasi constant for each remaining cluster. An application to identify the nonlinear model of a brushless motor as actuator of a control system is developed, where the identification errors of the torque command and of the position signal converge to zero.

References

[1] B.S Atal “A Model of LPC in Terms of Eigenvectors of the Autocorrelation Matrix of the Impulse Response of the LPC Filter”, Acoustic Research Department AT&T Bell Laboratories, Murrai Hill, 1989.

[2] T.W.S. Chow and S.Y.Cho, “An Accelerated Recurrent Network Training Algorithm using IIR Filter Model and Recursive Least Squares Method”, IEEE Transactions on Circuits and Systems, 44(11), 1997, pp.1082-1086.

[3] G.Cirrincione, M.Cirrincione, Chuan Lu and S. Van Huffel, “A Novel Neural Approach to Inverse Problems with Discontinuities (the GMR Neural Network)”, Proceedings of the IEEE International Joint Conference, Vol. 4, 2003, pp. 3106-3111

[4] M.Collins, S. Dasgupta and R. Schapire, “A Generalization of Principal Component Analysis to the Exponential Family. Advanced in Neural Information Processing Systems (NIPS), Vol.14, 2001.

[5] P.De Groen “An Introduction to Total Least Squares”, Nieuw Archief voor Wiskunde, Vierde serie, Vol. 14, 1996, pp. 237-253.

[6] I.Jollife, “Principal Component Analysis”, Springer-Verlag, New York, 1986.

[7] M. J. Piovoso and K.A. Kosanovich, “The Use of Multivariate Statistics in Process Control” The Control Handbook, CRC PRESS, IEEE PRESS, 1996, pp. 561-573.

[8] T. Poggio and F. Girosi, “Networks for Approximation and Learning”, Proceedings of the IEEE Conference, 78(9), 1990, pp.1481-1497.

[9] F.M. Raimondi, T. Raimondi, Maurizio Melluso, “Stochastic Estimate PCA Algorithm for Planar Industrial Manipulator Control”, Proceedings of the IEE Control 2004 Conference, Bath (U.K.), 2004.

[10] Magda Ruiz, J. Colomer, J. Colprim and J. Melendez, “Multivariate Statistical Process Control for Situation Assessment of a Sequencing Batch Reactor”, Proceedings of the UKAC-IEE Conference, Bath(U.K.), 2004

[11] M. Tipping and C. Bishop, “Probabilistic Principal Component Analysis”, Journal of the Royal Statistical Society, 61(3) , 1999, pp.611-622.

[12] Liao Y.C, “Order Recursive Algorithm for ARMA Identification”, Proceedings of IEEE International Conference on Systems Engineering, 1989, pp-209-211.

[13] Ananth I., “Introduction to Automated Speaker Verification”, Department of Electrical Engineering Pennsylvania State University, 2003.

[14] Magda Ruiz, Gurkan Sin, Xavier Berjaga, Jesus Colprim, Sebastià Puig, Joan Colomer, “Multivariate Principal Component Analysis and case-Based Reasoning for monitoring, fault detection and diagnosis in a WWTP”, Water Science and Technology, Vol 64, Issue 8, 2011, pp.1661-1667.