# Stochastical Real Time Finite State Machine LPC for Planar Manipulator Control System Model Estimation

FRANCESCO M. RAIMONDI, MAURIZIO MELLUSO
Dipartimento di Ingegneria dell'Automazione e dei Sistemi
University of Palermo
Viale delle Scienze- Post Code 90128
PALERMO (ITALY)

*Abstract:* - This paper presents a new stochastical real-time LPC (Last Principal Component) algorithm to estimate single-input-single-output (SISO) and multiple-input-multiple-output (MIMO) varying time models from input output data clusters of non stationary black boxes. Each of data clusters is on a time window. An application to estimate the control system model of a planar manipulator is developed. In fact many mathematical models of physical systems are non stationary such as industrial manipulator model. A real time estimation algorithm via stochastical LPC algorithm and an appraiser called "*finite state machine"* is then described. For every data cluster the finite state machine updates the parameters of a Gaussian varying time model via an optimality design criterion that maximises the Likelihood function. The estimated steady-state parameters are constant values. By applying to two links planar manipulator, numerical tests of simulation in Matlab 6.5 demonstrate the effectiveness of this algorithm.

*Key-Words:* - LPC, digital filters, finite state machine, maximum likelihood, manipulator control, estimation.

## 1 Introduction

Fundamental objective of Principal Component Analysis (PCA) [6,7,10,11] is projecting a set of input output data of a system into a lower dimensional space that accurately characterizes the state of the process. This can be developed from synthesis of data which are contained in a matrix by linear combinations of observed variables (principal components) which are able to reproduce variability of observed variables. In [4], [13], PCA is formulated within a maximum-likelihood framework and the subspace is estimated in Maximum Likelihood sense using a probabilistic generative model. Considering only the last principal components, PCA algorithm is called "Last Principal Components" (LPC). "Total Least Square" (TLS) algorithm [1], [5] is starting point of LPC algorithm. While TLS algorithm can be only applied to identification of static systems, LPC algorithm can be also applied to on-line identification of dynamic systems. In the world of industrial robotic applications [12] a Proportional-Derivative (PD) controller is often applied; a PD controller doesn't feel the effect of possible changes of load. In [9] an PD control for planar manipulators with two link and non-flexible joints is proposed. In every case, the problems consist in uncertainties of the model [9] as for example the friction torques. Therefore in literature [2,3], [8] there are techniques of identification based on training and adaptive-training algorithms using neural networks. In [11] a combination between multivariate statistical process control based on PCA algorithm and an automatic classification algorithm is developed to application in wast water treatment plant and have been created a model that describes the batch direction. In [10] is proposed a stochastical estimate algorithm based on PCA/LPC to estimate the nonlinear model of a planar manipulator. But in [10] the algorithm is off-line and the mathematical model of a planar industrial manipulator is non-stationary. Therefore an real time algorithm LPC must be developed.

In this paper is developed a new real-time stochastical LPC algorithm applied to estimate SISO and MIMO varying time models from data clusters of non stationary black boxes. Each of data cluters is on a window time. The paper is organized as follows. In Section 2 the details of the stochastical real time LPC algorithm will be pointed out. For every data cluster this algorithm updates the parameters of an LPC filter (Gaussian model) according to an optimality criterion that maximises the likelihood function and using an appraiser called "*finite state machine*". In Section 3 an application to two joints and two degree of freedom planar manipulator will be presented. The inputs of the appraiser are input-output data cluters of an experimental PD MIMO control system of the manipulator. The outputs are varying-time parameters of bidimensional LPC filter and the steady-state outputs are parameters of a diagonal matrix (2x2) of transfer functions in 'z' domain. This matrix is the estimated closed loop model of system control PD of the planar manipulator. Trajectories for testing real time LPC algorithm have been implemented in C language using software, hardware and graphic interface of an industrial manipulator. Efficiency of this algorithm is validated by numerical simulation tests using Matlab 6.5. In steady state the real time LPC of this paper is compared with off-line version [10] by use of Integral Absolute Error (IAE) performance index.

## 2 SISO and MIMO model estimation using real time LPC algorithm

This section presents the real time LPC algorithm and shows how a new real time version of this algorithm can be applied for estimating varying time SISO and MIMO models.

The dynamical linear-in-the-parameters-model of a LPC digital filter is:

$$(\mathbf{Y}_n+\boldsymbol{\varepsilon}_n)^T\mathbf{a}+(\mathbf{U}_n+\xi_n)^T\mathbf{b}=r(n)\quad n\in Z \qquad (1)$$

where $\mathbf{Y_n}$ is the vector of nominal output values, $\boldsymbol{\varepsilon}_n$ is the output noise vector, $\mathbf{U_n}$ is the vector of nominal input values, $\xi_n$ is input noise vector and $n$ is the discrete time. The *r(n)* term is a random noise. It will be assumed that *r(n)* is a random variable with independent values and with gaussian distribution as follows:

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

The *r(n)* gaussian distribution function is defined as [10]:

$$f[r(n)]=\frac{1}{\sqrt{2\pi\sigma_r}}\exp\left(-\frac{(r(n)-c)^2}{2\sigma_r^2}\right)=$$
$$\frac{1}{\sqrt{2\pi\sigma_r}}\exp\left(-\frac{(\mathbf{Y_m}(n)\mathbf{a}+\mathbf{U_m}(n)\mathbf{b}-c)^2}{2\sigma_r^2}\right)= \qquad (3)$$
$$=\frac{1}{\sqrt{2\pi\sigma_r}}\exp\left(-\frac{(\mathbf{z}_n^T\mathbf{l}-c)^2}{2\sigma_r^2}\right)$$

where $\mathbf{z_n}$ components are the measured values. These values can be written as a sum of the input output nominal values and of the noise, that is:

$$\mathbf{z}_n^T=\left[\mathbf{Y_m}(n)\middle|\mathbf{U_m}(n)\right]=\left[\mathbf{Y_n}(n)+\boldsymbol{\varepsilon}_n\middle|\mathbf{U_n}(n)+\xi_n\right] \qquad (4)$$

and

$$\mathbf{Y_n}(n)=[y_n(n),y_n(n-1),...,y_n(n-r)]^T$$
$$\mathbf{U_n}(n)=[u_n(n),u_n(n-1),...,u_n(n-r)]^T$$
$$\boldsymbol{\varepsilon}_n(n)=[\varepsilon_n(n),\varepsilon_n(n-1),...,\varepsilon_n(n-r)]^T \qquad (5)$$
$$\xi_n(n)=[\xi_n(n),\xi_n(n-1),...,\xi_n(n-r)]^T$$

The probability function is defined as:

$$f[r(1),r(2),...,r(N)]=\prod_{n=1}^{N}f(r(n)) \qquad (6)$$

The logarithmic Likelihood function is given by:

$$\log\{f[r(1),r(2),...,r(N)]\}=$$
$$\sum_{n=1}^{N}\left[-\log(2\pi)-\frac{1}{2}\log(\sigma_r^2)-\frac{(\mathbf{z}_n^T\mathbf{l}-c)^2}{2\sigma_r^2}\right] \qquad (7)$$

where:

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

An efficient technique for parametric estimate is the estimation in Maximum Likelihood sense. As is well known the maximum of (7) is equivalent to [10]:

$$\min(J)=\min\left[\frac{1}{2}N\log\sigma_r^2+\sum_{n=1}^{N}\frac{(\mathbf{z}_n^T\mathbf{l}-c)^2}{2\sigma_r^2}-\frac{1}{2}\lambda(\mathbf{l^T l}-1)\right] \qquad (8)$$

The λ term is Lagrange multiplier. Differentiating the performance index (8) by calculus of $\partial J/\partial c$ and $\partial J/\partial \mathbf{l}$, yields:

$$\mathbf{Al}=\lambda\sigma_r^2\mathbf{l} \qquad (9)$$

where

$$\mathbf{A}=\sum_{n=1}^{N}(\mathbf{z}_n-\mathbf{Z})(\mathbf{z}_n-\mathbf{Z})^T=\sum_{n=1}^{N}\overline{\mathbf{z}}_n\overline{\mathbf{z}}_n^T \qquad (10)$$

and

$$\mathbf{Z}=\frac{1}{N}\sum_{n=1}^{N}\mathbf{z}_n\quad . \qquad (11)$$

All the eigenvalues of **A** matrix are the principal components. The solution of optimization problem (8) is the minimum eigenvalue (*last principal components*) of **A** matrix (cf. eqs. 10-11) [10]. From the eigenvector which corresponds to last principal components, **a** and **b** parameters of the model (1) are evaluated. Therefore the steps of off-line LPC algorithm for SISO model estimation are as follows:

1) define the the input output data matrix of SISO black box:

$$\mathbf{z}_n^T=\begin{bmatrix} Y_m(n) & ... & Y_m(n-N) & U_m(n) & ... & U_m(n-N) \\ . & . & . & . & . & . \\ Y_m(n-r) & ... & Y_m(n-r-N) & U_m(n-r) & ... & U_m(n-r-N) \end{bmatrix} \qquad (12)$$

2) compute mean value of the single lines of the $\mathbf{z_n}$ matrix and define **Z** vector (cf eq. 11);
3) subtract the elements of **Z** from lines of $\mathbf{z_n}$ matrix for obtaining $\overline{\mathbf{z}}_\mathbf{n}$ matrix (cf eq. 10);
4) compute $\mathbf{A}=\overline{\mathbf{z}}_N\overline{\mathbf{z}}_N^T$ ;
5) calculate the eigenvalues and the eigenvectors of **A** matrix and evaluate the minimum eigenvalue.

The LPC algorithm is also applied for MIMO models estimation. Consider a system with $q$ measured input values $(\mathbf{U}_{mj}(n)\ \ j=1...q)$ and $q$

measured output values $(\mathbf{Y}_{mj}(n)\, j=1...q)$. Note that there are $q$ equations systems here:

$$a_{oj}\mathbf{Y}_{mj}(n)+a_{1j}\mathbf{Y}_{mj}(n-1)+...+a_{Nj}\mathbf{Y}_{mj}(n-N)=$$
$$=-b_{oj}\mathbf{U}_{mj}(n)-b_{1j}\mathbf{U}_{mj}(n-1)-...-b_{Nj}\mathbf{U}_{mj}(n-N)+r(n)$$
$$j=1...q \tag{13}$$

Each system (cf. eqs. 5) is of $r$ equations and $N$ unknown quantities $(r \le N)$, where $N$ is the order of the estimated model. LPC algorithm is as follows:

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

$$\mathbf{z}_{nj}^T=\begin{bmatrix} Y_{mj}(n) & \ldots & Y_{mj}(n-N) & U_{mj}(n) & \ldots & U_{mj}(n-N) \\ . & . & . & . & . & . \\ Y_{mj}(n-r) & \ldots & Y_{mj}(n-r-N) & U_{mj}(n-r) & \ldots & U_{mj}(n-r-N) \end{bmatrix}$$
$$j=1...q \tag{14}$$

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

$$\mathbf{A}_j=\sum_{n=1}^{N}(\mathbf{z}_{nj}-\mathbf{Z}_j)(\mathbf{z}_{nj}-\mathbf{Z}_j)^T=\sum_{n=1}^{N}\bar{\mathbf{z}}_{nj}\bar{\mathbf{z}}_{nj}^T \tag{15}$$
$$j=1...q$$

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

From the eigenvectors the following parameters are achieved (cf. eqs 13):

$$\mathbf{a}_j^T=\left[a_{0j},a_{1j},...,a_{Nj}\right] \quad \mathbf{b}_j^T=\left[b_{0j},b_{1j},...,b_{Nj}\right] \quad j=1...q$$

But in many systems the parameters can change time after time because the model is non stationary. Therefore an appraiser wich must estimate $\mathbf{a}_j$ 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. This system can have a possible configuration called "*state*". As 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 (cf. eq. 15);
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 estimating;
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;
2) output calculus only from new state vector.

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. 7-8). In other words the real time algorithm is designed to be operated trough repeated execution of LPC algorithm depending on data clusters dimension.

# 3 Real Time LPC algorithm for planar manipulator control system model estimation: simulation results

This section shows how the new real time version of LPC algorithm can be applied for estimating the varying time model of the PD control system of a planar industrial manipulator with two link and non flexible joints. Let us define the following notation: $\theta_{1r},\theta_{1a},\theta_{2r}$ and $\theta_{2a}$ are reference and actual angular position of joint 1 and joint 2 respectively while the work-space coordinates are *(x,y)* [9], [12].

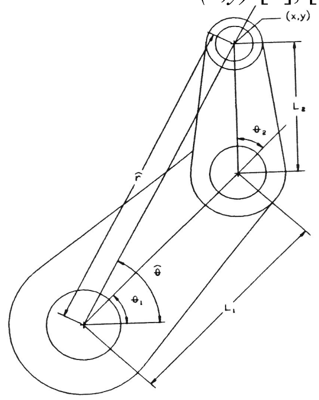

Fig. 1. Planar Manipulator.

Based on this notation and according to LPC algorithm for MIMO systems, the following matrices and vectors can be written (cf eqs 13-15) :

$$\mathbf{Y_m}(n)=\mathbf{Y_n}(n)+\boldsymbol{\varepsilon}_n(n)=\begin{bmatrix}\mathbf{Y}_{m1}(n)\\ \mathbf{Y}_{m2}(n)\end{bmatrix}=\begin{bmatrix}\boldsymbol{\theta}_{1a}(n)\\ \boldsymbol{\theta}_{2a}(n)\end{bmatrix} \tag{16}$$
$$\mathbf{U_m}(n)=\mathbf{U_n}(n)+\boldsymbol{\xi}_n(n)=\begin{bmatrix}\mathbf{U}_{m1}\\ \mathbf{U}_{m2}\end{bmatrix}=\begin{bmatrix}\boldsymbol{\theta}_{1r}(n)\\ \boldsymbol{\theta}_{2r}(n)\end{bmatrix}$$

$$\boldsymbol{\theta}_{1a}^T(n) = [\theta_{1a}(n)..\theta_{1a}(n-r)]$$
$$\boldsymbol{\theta}_{2a}^T(n) = [\theta_{2a}(n)..\theta_{2a}(n-r)] \qquad (17)$$

$$\boldsymbol{\theta}_{1r}^T(n) = [\theta_{1r}(n)..\theta_{1r}(n-r)]$$
$$\boldsymbol{\theta}_{2r}^T(n) = [\theta_{2r}(n)..\theta_{2r}(n-r)] \qquad (18)$$

$$\boldsymbol{\varepsilon}_n(n) = \begin{bmatrix} \varepsilon_{n1}(n), \varepsilon_{n1}(n-1), ..., \varepsilon_{N1}(n-r) \\ \varepsilon_{n2}(n), \varepsilon_{n2}(n-1), ..., \varepsilon_{n2}(n-r) \end{bmatrix}$$
$$\boldsymbol{\xi}_n(n) = \begin{bmatrix} \xi_{n1}(n), \xi_{n1}(n-1), ..., \xi_{n1}(n-r) \\ \xi_{n2}(n), \xi_{n2}(n-1), ..., \xi_{n2}(n-r) \end{bmatrix} \qquad (19)$$

and also is:

$$\mathbf{z}_{N1}^T = \begin{bmatrix} \theta_{1a}(n) & ... & \theta_{1a}(n-N) & \theta_{1r}(n) & ... & \theta_{1r}(n-N) \\ . & . & . & . & . & . \\ \theta_{1a}(n-r) & ... & \theta_{1a}(n-r-N) & \theta_{1r}(n-r) & ... & \theta_{1r}(n-r-N) \end{bmatrix} \qquad (20)$$

$$\mathbf{z}_{N2}^T = \begin{bmatrix} \theta_{2a}(n) & ... & \theta_{2a}(n-N) & \theta_{2r}(n) & ... & \theta_{2r}(n-N) \\ . & . & . & . & . & . \\ \theta_{2a}(n-r) & ... & \theta_{2a}(n-r-N) & \theta_{2r}(n-r) & ... & \theta_{2r}(n-r-N) \end{bmatrix} \qquad (21)$$

$$\mathbf{Z}_j = \frac{1}{N}\sum_{n=1}^{N} \mathbf{z}_{nj} \quad \mathrm{j}=1,2 \qquad (22)$$

$$\mathbf{A}_j = \sum_{n=1}^{N} (\mathbf{z}_{nj} - \mathbf{Z}_j)(\mathbf{z}_{nj} - \mathbf{Z}_j)^T = \sum_{n=1}^{N} \overline{\mathbf{z}}_{nj}\overline{\mathbf{z}}_{nj}^T \qquad (23)$$
$j = 1,2$

Note that there are two equations systems here, one for the joint 1 and one for the joint 2 (cf.fig.1):

$$a_{o1}\boldsymbol{\theta}_{1a}(n) + a_{11}\boldsymbol{\theta}_{1a}(n-1) + ... + a_{N1}\boldsymbol{\theta}_{1a}(n-N) =$$
$$= -b_{o1}\boldsymbol{\theta}_{1r}(n) - b_{11}\boldsymbol{\theta}_{1r}(n-1) - ... - b_{N1}\boldsymbol{\theta}_{1r}(n-N) + r(n)$$
$$a_{o2}\boldsymbol{\theta}_{2a}(n) + a_{12}\boldsymbol{\theta}_{2a}(n-1) + ... + a_{N2}\boldsymbol{\theta}_{2a}(n-N) =$$
$$= -b_{o2}\boldsymbol{\theta}_{2r}(n) - b_{12}\boldsymbol{\theta}_{2r}(n-1) - ... - b_{N2}\boldsymbol{\theta}_{2r}(n-N) + r(n) \qquad (24)$$

The parameters of the manipulator can change time after time because the manipulator model is non stationary. Therefore the appraiser must estimate the following parameters in every time:

$$\mathbf{a}_1^T = [a_{01}...a_{N1}] \quad \mathbf{b}_1^T = [b_{01}...b_{N1}]$$
$$\mathbf{a}_2^T = [a_{02}...a_{N2}] \quad \mathbf{b}_2^T = [b_{02}...b_{N2}] \qquad (25)$$

In this case the parameters sizes of the finite state machine are the sizes of parameters (25) and the data cluster dimension is the updating interval of parameters values (25). Step by step the appraiser updates the state. The data $\mathbf{a_1}, \mathbf{b_1}, \mathbf{a_2}, \mathbf{b_2}$ are updated using LPC algorithm for MIMO systems depending on data cluster dimension. In this case LPC algorithm for MIMO systems is applied for *j=1,2 (q=2)* (cf. eqs. 13-15). The steady-state solutions of $\mathbf{a_1}, \mathbf{b_1}, \mathbf{a_2}, \mathbf{b_2}$ are constant values and therefore the estimated model of the control system of the manipulator can be expressed in the form as:

$$\mathbf{W}(z) = \mathbf{diag}(W_1(z) \;\; W_2(z)) \qquad (26)$$

where:

$$W_1(z) = \frac{\theta_{1a}}{\theta_{1r}} = \frac{b_{01} + b_{11}z^{-1} + ... + b_{N1}z^{-N}}{a_{01} + a_{11}z^{-1} + ... + a_{N1}z^{-N}};$$
$$W_2(z) = \frac{\theta_{2a}}{\theta_{2r}} = \frac{b_{02} + b_{12}z^{-1} + ... + b_{N2}z^{-N}}{a_{02} + a_{12}z^{-1} + ... + a_{N2}z^{-N}} \qquad (27)$$

As regards the numerical simulation tests, after an opportune planning proceeding an input reference trajectory has been introduced in the work space. Subsequently the inverse kinematics [9],[12] converts each cartesian workspace point (*x,y*) along a straight line path into joint angles $(\theta_{1r}, \theta_{2r})$ (see fig. 1) as follows:

$$\theta_{1r} = \arctan\left(\frac{y}{x}\right) - \arcsin\left[\frac{L_2 \sin\theta_{2r}}{\sqrt{x^2 + y^2}}\right] \qquad (28)$$

$$\theta_{2r} = \pm\arccos\left[\frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}\right] \qquad (29)$$

where $L_1=0.359m$ and $L_2=0.241m$ are the lenghts of the links. The experimental control system of manipulator is an PD control [9], [12] and it is implemented in C language. Figure 2 shows the referred trajectories in work space and the actual trajectories in joint space from PD control system. The initial conditions in work-space are (*x(0)=0.6m y(0)=0m*), that is the homing position [9]. For obtaining data set for the appraiser, the maximum velocity of each joint is equal to 2 rad/s.

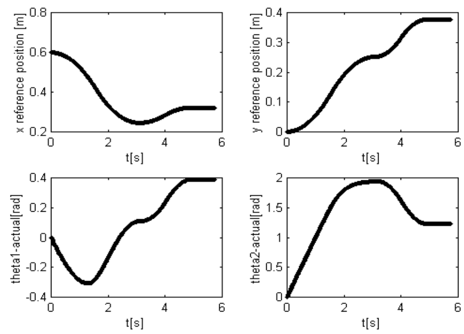


Fig. 2. Input-output trajectories – test data set x[m], y[m], $\theta_{1a}$[rad], $\theta_{2a}$[rad].

For each of data clusters the referred and actual trajectories in joint space are input and output test data set of the finite state machine LPC which is suitably simulated in Matlab 6.5 envinronment. The *N* order of estimated model is equal to 2. Table 1 resumes the parameter values of real time LPC.

| | |
|---|---|
| Sample Time | 10 ms |
| Parameter size | 3 |
| Window-size | 572 samples |
| Window-time | 5.72s |
| Data-cluster | 16 samples |

Table 1. Parameters of real time LPC.

It is obvius that the window time of each of data clusters is equal to 0.016s. Figures 3-6 show the varying time parameters of bidimensional LPC filter (cf. eqs. 24-25).

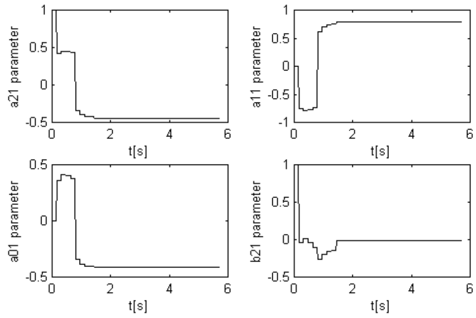


Fig. 3. Estimation of varying-time $a_{21}$, $a_{11}$, $a_{01}$, $b_{21}$.

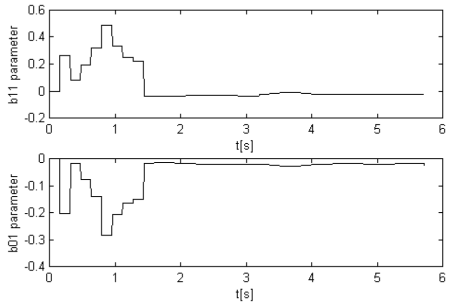


Fig. 4. Estimation of varying-time $b_{11}$ and $b_{01}$.

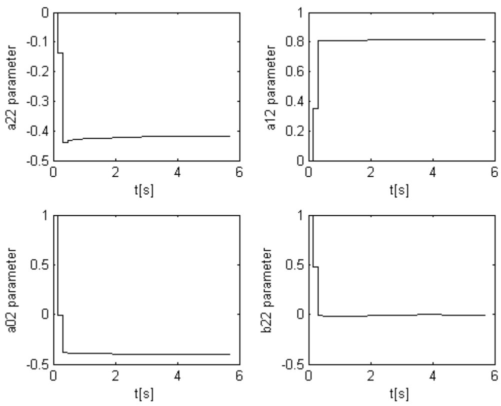


Fig. 5. Estimation of varying-time $a_{22}$, $a_{12}$, $a_{02}$, $b_{22}$ .

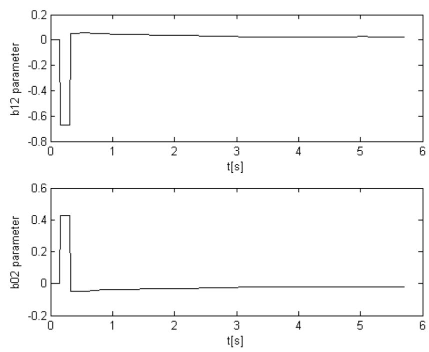


Fig. 6. Estimation of varying-time $b_{12}$ and $b_{02}$.

Note that the steady state solutions of parametric estimation are quasi-constant values. Therefore Tables 2 and 3 resume the steady-state mean values solutions of the components of **a**$_1$, **b**$_1$ and **a**$_2$, **b**$_2$ vectors respectively (cf. eq. 27).

| | $a_{21}$ | $a_{11}$ | $a_{01}$ | $b_{21}$ | $b_{11}$ | $b_{01}$ |
|---|---|---|---|---|---|---|
| **a**$_1$ | -0.43 | 0.79 | -0.41 | - | - | - |
| **b**$_1$ | - | - | - | 0.0118 | -0.0259 | -0.0155 |

Table 2. **a**$_1$ and **b**$_1$ steady-state values.

| | $a_{22}$ | $a_{12}$ | $a_{02}$ | $b_{22}$ | $b_{12}$ | $b_{02}$ |
|---|---|---|---|---|---|---|
| **a**$_2$ | -0.41 | 0.81 | -0.39 | - | - | - |
| **b**$_2$ | - | - | - | -0.0058 | 0.025 | -0.022 |

Table 3. **a**$_2$ and **b**$_2$ steady-state values.

The results of estimation process using the parameters of tables 2-3 are shown in Figures 7-10. Real time LPC of this work is also compared with off line algorithm [10] by use of Integral Absolute Error (IAE) perfomance index:

$$I.A.E. = \int_0^t |\mathbf{e}| dt \, , \qquad (30)$$

where $\mathbf{e}^T = [e_1, e_2]$ and $e_j$ is the difference between the actual position of *j-th* joint using PD control and the estimated actual position of *j-th* joint using LPC algorithm, that is the estimation error of *j-th* joint position( *j=1,2*).

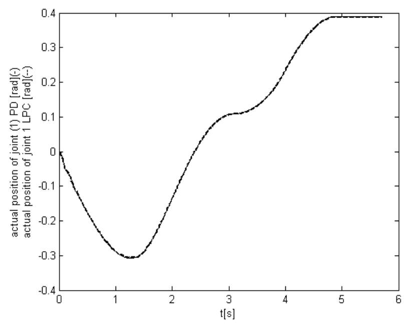


Fig.7. Actual joint 1 position by PD[rad](-); estimated joint 1 position by real time LPC[rad](--).

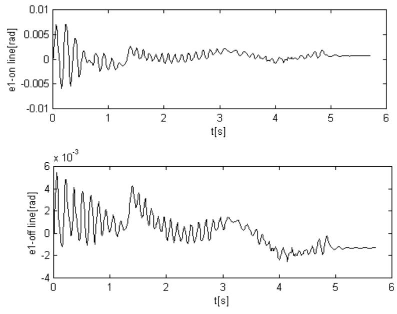


Fig.8. $e_1$ using real time and off line LPC[rad].

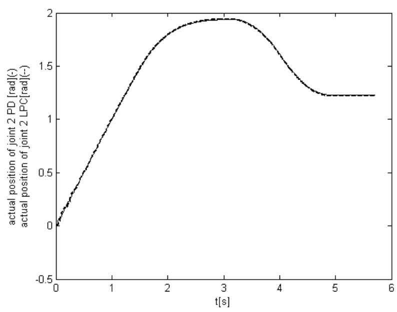


Fig.9. Actual joint 2 position by PD[rad](-); estimated joint 2 position by real time LPC [rad](--).

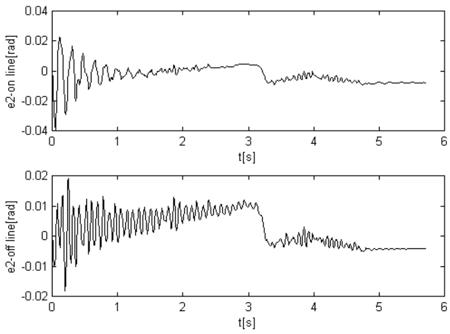


Fig.10. $e_2$ using real time and off-line LPC[rad].

Table 4 resumes the IAE values.

| IAE | Off-line | Real-Time |
|---|---|---|
| Joint 1 | 0.01 | 0.007652 |
| Joint 2 | 0.028 | 0.032 |

Table 4. IAE Performance Index.

By examining the simulation Figures 7-10 and Table 4 the following remarks can be made. The model estimation errors are much smaller than the effectively angular displacements of the joints. The estimation errors achieved from real time LPC are less rapidly varying time than those achieved from off-line LPC. Also IAE performance index of real time algorithm for joint 1 position estimation error is lower than off line version. Therefore when implemented with finite state machine, it turns out that LPC algorithm has a better performance.

# 4 Conclusion

This paper has introduced a new real-time stochastical LPC to estimate SISO and MIMO varying time models from input output data clusters of black boxes. An application to estimate the PD MIMO control system model of a two links planar manipulator is developed. The approach is based on LPC using a finite state machine. LPC is formulated within a maximum likelihood framework using a Gaussian varying time model. In comparison with off-line version there are many advantages. In fact the test data set in all the instants can be unknown, because the data processing depends on data cluster dimension in a time window. Furthermore this algorithm can be utilized when the plant parameters change and a manipulator is a non-stationary system. Numerical simulations show acceptable values of the model estimation errors in steady state.

*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 Transaction on Circuits and Systems,* Vol.44, No.11, 1997.

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

[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,* deel 14, pp 237-253, 1996.

[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,* pp.561-573, 1996.

[8] T. Poggio and F. Girosi, Networks for Approximation and Learning, *Proceeding of the IEEE Conference,* vol.78, No.9, 1990.

[9] F.M.Raimondi, T. Raimondi and M.Melluso, Modelling and Control Strategies for Planar Two-Link Industrial Robot, *Acta of D.I..A.S.-Automation and Systems Engineering Department,* University of Palermo, 2004.

[10] F.M. Raimondi, T. Raimondi and M.Melluso, Stochastic Estimate PCA Algorithm for Planar Industrial Manipulator Control, *Proceeding of the UKAC-IEE Control 2004 Conference,* Bath (U.K.), 2004.

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

[12] Lorenzo Sciavicco and Bruno Siciliano, *Modelling and Control of Robot Manipulators*, 2nd-Edition, Verlag Advanced Textbooks in Control and Signal Processing Series,UK,2000.

[13] M. Tipping and C. Bishop, Probabilistic Principal Component Analysis, *Journal of the Royal Statistical Society,* Vol.61, No.3, pp 611-622, 1999.