TWO-DIMENSIONAL ADAPTIVE PARAMETER ESTIMATION

A. Madady* , M. Shafiee**

***** Electrical Engineering Department- Amir-Kabir University of Technology (Tafresh Branch)
Hafez Ave.,Tehran – IRAN, email: *madady@aut.ac.ir*

****** Electrical Engineering Department- Amir-Kabir University of Technology (Tehran Polytechnic)
Hafez Ave.,Tehran – IRAN, email: *shafiee @aut.ac.ir*

Key words: Adaptive parameter estimation, 2-D systems, Multidimensional systems, 2-D algorithms.

Abstract

This paper gives a formulation of 2-D adaptive parameter estimation (2-DAPE) problem in general case; when both of the 2-D space coordinates are possibly unbounded. A solution procedure is presented for this problem. In this solution procedure, the parameters of the underlying 2-D system are decomposed into two blocks, namely the horizontal and vertical blocks. The estimation of the horizontal block is made by means of a horizontal updating scheme, whereas the vertical block is to be estimated by means of a vertical updating scheme. The 2-D Lyapunov approach is employed to ensure the convergence of the presented 2-D estimation procedure. The computer simulation results are included to illustrate the effectiveness of the proposed procedure.

1 Introduction

Since the introduction of the first state-space model for 2-D systems almost three decades ago in [1], this area is yet undeveloped and offers a rich opportunity for research. Many applications in image processing, 2-D filter design, iterative learning modeling and some industrial processes modeling and control are reported in [2-5]. The evolution of 2-D systems theories from the concepts and results in 1-D systems is well known. Correspondingly, such topics as modeling, stability, stabilization by the state and output feed-back, controllability and observability, pole placement and model matching, model following, optimal control problems, observer and state estimation, transfer function identification and robust control are studied in this category. However, many established results for conventional 1-D systems have not been extended to 2-D systems due to their analytical and structural complexity. Specifically, although the results on adaptive parameter estimation for linear 1-D systems have been well studied [6-8], there has been no known result on the 2-D adaptive parameter estimation apart from some earlier discussions of this problem [9-12].

The aim of this paper is to extend the adaptive parameter estimation approach to 2-D systems when both of the independent variables of the underlying systems are unbounded. To the authors' best knowledge this paper seems to be the first attempt in developing such adaptive parameter estimation procedure to cover the 2-D systems when both of their independent variables are unbounded.

The paper is organized as follows. Section 2 formulates the 2-DAPE problem. Section 3 reviews the previous investigations of the parameter estimation in 2-D case. In Section 4, the interest procedure of the paper is presented. In Section 5, stability (convergence) of the presented procedure is analyzed. In Section 6 we explain the implementation of the proposed algorithm. In Section 7 a simulation example is given. Conclusion is deferred to Section 8.

2 Problem statement

Consider a 2-D discrete time system expressed by regressor model:

$$y(i,j) = \theta z(i,j) \quad i,j = 0,1,... \tag{1}$$

where i and j are non-negative integer-valued horizontal and vertical coordinates respectively, $y \in \Re^m$ and $z \in \Re^n$ are respectively the output and regressor vectors. The $\theta \in \Re^{m\times n}$ includes the parameters of the system, which is called the system parameters matrix.

We define the 2-DAPE problem as follows:

Assume θ *in (1) be constant but unknown,* $y(i,j)$ *and* $z(i,j)$ *are measurable, by utilizing these measurements establish a 2-D estimator for* θ*, as* $\hat{\theta}(i,j)$ *so that:*

$$\begin{gathered}\lim \left(\hat{\theta}(i,j) - \theta\right) = 0 \\ i \text{ and } / \text{or } j \to \infty\end{gathered} \tag{2}$$

3 The Previous investigations

Few efforts are done to estimate the 2-D systems parameters [9-12], that is to estimate θ in (1) utilizing the measurements of $y(i,j)$ and $z(i,j)$. One of the 2-D space coordinates (for example variable i) was assumed to be bounded in all these estimation procedures, unlike the classical 2-D systems theories. For this reason, the general form of the all presented estimation algorithms was as follows:

$$\hat{\theta}(i+1,j) = \hat{\theta}(i,j) + \Delta_1(i,j) \tag{3.a}$$

$$\hat{\theta}(0,j+1) = \hat{\theta}(M,j) + \Delta_2(j) \tag{3.b}$$

$$i = 0,1,...,M-1 \qquad j = 0,1,...$$

where M is a finite known number, which is the upper bound for i, and $\hat{\theta}(i,j)$ is the obtained estimation for θ at point (i,j). The $\Delta_1(i,j)$ and $\Delta_2(j)$ are the modifier terms, which were assigned in some suitable manners.

Just as is inferred from (3.a), the estimation of θ at point i form row j is calculated utilizing the information of the previous point of the same row, that is the information at point $(i-1,j)$. But according to (3.b), the estimation at the beginning point of each row, $\hat{\theta}(0,j)$, is obtained from the information of the end-point of the previous row, that is the information at the point $(M,j-1)$. The difference between these algorithms is in the assignment manner of $\Delta_1(i,j)$ and $\Delta_2(j)$.

Although, the appearance form of these algorithms is 2-D, but their nature is 1-D. To demonstrate this fact we define the variable k as follows:

$$k = i + (M+1)j \quad i = 0,1,\ldots,M-1 \quad j = 0,1,\ldots \tag{4}$$

By this definition, (3.a) and (3.b) can be combined together. The result of this combination is the following 1-D algorithm:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \Delta(k) \quad k = 0,1,\ldots \tag{5}$$

where $\hat{\theta}(k)$ is the estimation of θ at point (i,j) which is associated with k according to (4), and $\Delta(k)$ is as:

$$\Delta(k) = \Delta_1\left(k - \left[\frac{k+1}{M+1}\right](M+1), \left[\frac{k+1}{M+1}\right]\right) \tag{6.a}$$

$$\text{if} \quad k+1 \neq (M+1), 2(M+1), 3(M+1), \ldots$$

and:

$$\Delta(k) = \Delta_2\left(\frac{k+1}{M+1} - 1\right) \tag{6.b}$$

$$\text{if} \quad k+1 = (M+1), 2(M+1), 3(M+1), \ldots$$

The $\left[\frac{k+1}{M+1}\right]$ denotes the integer part of $\frac{k+1}{M+1}$.

Obviously, none of the algorithms with structures that are given in (3.a) and (3.b), will not be usable when both of the independent variables i and j are unbounded.

In the next section a comprehensive procedure will be presented for estimation the 2-D systems parameters. This procedure is truly 2-D that is neither of the variables i and j is bounded. Of course, the presented procedure will be usable when one of these variables is bounded.

4 Solution procedure of the 2-DAPE problem

4.1 The Generic 2-D adjustment law

Consider (1), the following 2-D identification model is presented:

$$\hat{y}(i,j) = \hat{\theta}(i,j) z(i,j) \quad i,j = 0,1,\ldots \tag{7}$$

where $\hat{\theta}(i,j) \in \Re^{m \times n}$ is an adjustable matrix and is considered as the estimation of θ at point (i,j).

The next step is the developing a 2-D adaptive law for adjusting $\hat{\theta}(i,j)$ until (2) can be established. For this purpose, we decompose the components of $\hat{\theta}(i,j)$ into two blocks, namely the *horizontal* and *vertical* blocks. Let $\hat{\theta}^h(i,j)$ and $\hat{\theta}^v(i,j)$ denote the horizontal and vertical blocks respectively. Based on the this decomposition the following 2-D generic law is proposed for adjusting the $\hat{\theta}(i,j)$:

$$\begin{aligned} \hat{\theta}^h(i+1,j) &= \hat{\theta}^h(i,j) + \Delta^h(i,j) \\ \hat{\theta}^v(i,j+1) &= \hat{\theta}^v(i,j) + \Delta^v(i,j) \end{aligned} \quad i,j = 0,1,\ldots \tag{8}$$

where $\Delta^h(i,j)$ and $\Delta^v(i,j)$ are respectively the horizontal and vertical 2-D modifier terms.

The updating law (8) is similar to the Rosser state-space model [1] and according to that the horizontal and vertical blocks of $\hat{\theta}(i,j)$ are modified respectively along the horizontal (i) and vertical (j) directions.

For running (8) the quantities $\hat{\theta}^h(0,j)$ $(j = 0,1,\ldots)$ and $\hat{\theta}^v(i,0)$ $(i = 0,1,\ldots)$ are needed, in addition to the modifier terms $\Delta^h(i,j)$ and $\Delta^v(i,j)$. We call the quantities $\hat{\theta}^h(0,j)$ $(j = 0,1,\ldots)$ and $\hat{\theta}^v(i,0)$ $(i = 0,1,\ldots)$ the boundary conditions of the (8), which must be adjusted according to a suitable manner and submitted to (8). The adjustment of these boundary conditions is a 1-D adjusting problem, and we consider that as:

$$\hat{\theta}^h(0,j+1) = \hat{\theta}^h(0,j) + \Delta_0^h(j) \quad j = 0,1,\ldots \tag{9.h}$$

$$\hat{\theta}^v(i+1,0) = \hat{\theta}^v(i,0) + \Delta_0^v(i) \quad i = 0,1,\ldots \tag{9.v}$$

where $\Delta_0^h(j)$ and $\Delta_0^v(i)$ are appropriate 1-D modifier terms.

Thus, the 2-D adjusting algorithm (8) needs the two minor 1-D adjusting algorithms (9) for its boundary conditions, while in the 1-D case there is not the boundary conditions problem. This is a principal difference between the 1-D and 2-D adjustment algorithms.

4.2 The 2-D adjusting algorithm based on the columnar decomposition

The components of $\hat{\theta}(i,j)$ in (7) must be decomposed into the horizontal and vertical blocks, for constructing the 2-D adjusting algorithm (8), just as is explained in subsection 4-1. Hence similarly, the components of θ in (1) are decomposed into the horizontal and vertical blocks, namely θ^h and

θ^v respectively. So that, the $\hat{\theta}^h(i,j)$ and $\hat{\theta}^v(i,j)$ will be respectively the estimations of θ^h and θ^v at the point (i,j).There exist various methods for accomplishing this decomposition. Here a straightforward method, namely the columnar decomposition method, is considered. If another method is chosen, to the authors' best knowledge seems the stability and asymptotic stability analysis of the obtained 2-D estimation algorithm will become very difficult and even impossible.

Usually, (1) is arisen from 2-D state-space or 2-D ARMA equations. Therefore, without loss of generality, it is assumed that the regressor $z(i,j)$ has more than one component. Hence θ has at least two columns. Thus, the columns of θ and so the components of $z(i,j)$ can be decomposed into the horizontal and vertical blocks. For this reason, (1) is rewritten as below form:

$$y(i,j) = \begin{bmatrix} \theta^h & \theta^v \end{bmatrix} \begin{bmatrix} z^h(i,j) \\ z^v(i,j) \end{bmatrix} \quad i,j = 0,1,... \tag{10}$$

where $\theta^h \in \Re^{m\times n_1}, \theta^v \in \Re^{m\times n_2}$ and $z^h \in \Re^{n_1}, z^v \in \Re^{n_2}$.

The dimensions of this decomposition, that is the number n_1 or n_2 is free, but their sum is n, because we have:

$$\theta = \begin{bmatrix} \theta^h & \theta^v \end{bmatrix}, \; z(i,j) = \begin{bmatrix} z^h(i,j) \\ z^v(i,j) \end{bmatrix} \tag{11}$$

We accomplish similar decomposition for identification model (7). Therefore (7) will be rewritten as:

$$\hat{y}(i,j) = \begin{bmatrix} \hat{\theta}^h(i,j) & \hat{\theta}^v(i,j) \end{bmatrix} \begin{bmatrix} z^h(i,j) \\ z^v(i,j) \end{bmatrix} \tag{12}$$

where $\hat{\theta}^h \in \Re^{m\times n_1}$ and $\hat{\theta}^v \in \Re^{m\times n_2}$.

Therefore, the 2-D adjusting algorithm (8) can be written as following compact form:

$$\begin{bmatrix} \hat{\theta}^h(i+1,j) \vdots \hat{\theta}^v(i,j+1) \end{bmatrix} = \begin{bmatrix} \hat{\theta}^h(i,j) \vdots \hat{\theta}^v(i,j) \end{bmatrix} + \Delta(i,j) \quad i,j = 0,1,... \tag{13}$$

where $\Delta(i,j) = \begin{bmatrix} \Delta^h(i,j) \vdots \Delta^v(i,j) \end{bmatrix}$

Here, we utilize the LMS method for determining the 2-D modifier term $\Delta(i,j)$. For this purpose, the error between the outputs of the system (10) and the identification model (12) will be considered:

$$e(i,j) = \hat{y}(i,j) - y(i,j) \quad i,j = 0,1,... \tag{14}$$

The quadratic index $g(i,j)$ is defined on $e(i,j)$ as:

$$g(i,j) = \frac{1}{2} e^T(i,j) P e(i,j) \quad i,j = 0,1,... \tag{15}$$

where $P \in \Re^{m\times m}$ is a symmetric positive definite matrix.

According to the LMS method, $\Delta(i,j)$ is chosen as follows:

$$\Delta(i,j) = \mu(i,j)\left(-\frac{\nabla g(i,j)}{\nabla\hat{\theta}(i,j)} \right) \tag{16}$$

where $\mu(i,j)$ is a positive scalar, namely the step size.

It is easy to derive that:

$$\frac{\nabla g(i,j)}{\nabla\hat{\theta}(i,j)} = Pe(i,j) z^T(i,j) \tag{17}$$

Thus (13) will become:

$$\begin{aligned} \begin{bmatrix} \hat{\theta}^h(i+1,j) \vdots \hat{\theta}^v(i,j+1) \end{bmatrix} &= \begin{bmatrix} \hat{\theta}^h(i,j) \vdots \hat{\theta}^v(i,j) \end{bmatrix} \\ &-\mu(i,j) Pe(i,j) z^T(i,j) \qquad i,j = 0,1,... \end{aligned} \tag{18}$$

Considering the form of the 2-D algorithm (18), its boundary conditions adjustment manners, that is the 1-D algorithms (9), are offered as follows:

$$\hat{\theta}^h(0,j+1) = \hat{\theta}^h(0,j) - \mu(0,j) Pe(0,j) z^{h^T}(0,j) \tag{19.h}$$

$$\hat{\theta}^v(i+1,0) = \hat{\theta}^v(i,0) - \mu(i,0) Pe(i,0) z^{v^T}(i,0) \tag{19.v}$$

$$i,j = 0,1,...$$

5 Stability and asymptotic stability analysis of the presented procedure

The estimation error at the point (i,j) is defined as follows:

$$\begin{aligned} &\theta^h(i,j) = \hat{\theta}^h(i,j) - \theta^h, \theta^v(i,j) = \hat{\theta}^v(i,j) - \theta^v \\ &\theta(i,j) = \begin{bmatrix} \theta^h(i,j) & \theta^v(i,j) \end{bmatrix} \quad i,j = 0,1,... \end{aligned} \tag{20}$$

Let the line L_k is defined as:

$$L_k = \left\{ (i,j) \middle| 0 \le i, 0 \le j \;,\; i+j = k \right\} \quad k = 0,1,... \tag{21}$$

This line has $k+1$ points; hence we define the *mean square of the estimation error on line* L_k, which is denoted by $e_\theta(k)$, as follows:

$$e_\theta(k) = \frac{1}{k+1} \sum_{(i,j)\in L_k} \sum Trace\left[\theta(i,j)\theta^T(i,j) \right] \tag{22}$$

5.1 Stability and asymptotic stability of the boundary conditions adjusting algorithms

Consider the 1-D algorithms (19.h) and (19.v), the concept of the stability of these algorithms is respectively as:

$$\lim_{j\to\infty} e(0,j) = 0 \;,\; \lim_{i\to\infty} e(i,0) = 0 \tag{23}$$

Also, the concept of their asymptotic stability is respectively as:

$$\lim_{j\to\infty} \theta^h(0,j) = 0 \;,\; \lim_{i\to\infty} \theta^v(i,0) = 0 \tag{24}$$

Lemma 1. *The 1-D algorithm (19.h) is stable, if the step size* $\mu(0,j)$ *is chosen in the following interval:*

$$0<\mu(0,j)<\frac{2}{z^T(0,j)z(0,j)\lambda_{\max}(P)} \quad j=0,1,\ldots \quad (25)$$

where $\lambda_{\max}(P)$ *denotes the maximum eigenvalue of the matrix* P.

For limitation in the paper length the proofs of the Lemmas and Theorems, which are based on the 2-D Lyapunov approach, are not given here and we will give them in the journal version of paper.

Lemma 2. *If 1-D algorithm (19.h) is stable, and the regressor* $z(0,j)$ *is sufficiently general so that there exists the number* n_0^v, *such that the matrix* $\Psi_0^v(j,n_0^v)$, *which is defined below, has full row rank for any beginning point* j, *then this algorithm will be asymptotically stable, too :*

$$\Psi_0^v(j,n_0^v)=\left[z(0,j)\;\; z(0,j+1)\;\cdots\; z(0,j+n_0^v)\right] \quad (26)$$

Lemma 3. *The 1-D algorithm (19.v) is stable, if the step size* $\mu(i,0)$ *is chosen in the following interval:*

$$0<\mu(i,0)<\frac{2}{z^T(i,0)z(i,0)\lambda_{\max}(P)} \quad i=0,1,\ldots \quad (27)$$

Lemma 4. *If 1-D algorithm (19.v) is stable, and the regressor* $z(i,0)$ *is sufficiently general so that there exist the number* n_0^h, *such that the matrix* $\Psi_0^h(i,n_0^h)$ *,which is defined below, has full row rank for any beginning point* i, *then this algorithm will be asymptotically stable, too:*

$$\Psi_0^h(i,n_0^h)=\left[z(i,0)\;\; z(i+1,0)\cdots z(i+n_0^h,0)\right] \quad (28)$$

5.2 Stability and asymptotic stability of the 2-D algorithm

The following stability definitions are presented about the 2-D algorithm (18).

Definition 1. The 2-D algorithm (18) is stable, if the error between the outputs of the system (10) and the identification model (12) vanishes in any manner we go away from the origin $(i,j)=(0,0)$, that is:

$$\lim_{i \text{ and/or } j\to\infty} e(i,j)=0$$

Definition 2. The 2-D algorithm (18) is marginal asymptotically stable if:

$$\lim_{k\to\infty} e_\theta(k)=0$$

where $e_\theta(k)$ is defined in (22).

Definition 3. The 2-D algorithm (18) is asymptotically stable if:

$$\lim_{i \text{ and/or } j\to\infty} \theta(i,j)=0$$

Obviously, considering the above definitions, the asymptotically stability is strong than the marginal asymptotically stability. That is, the asymptotically stability implies the marginal asymptotically stability, but the reverse of this fact is not true in general case.

Theorem 1. *The 2-D algorithm (18) is stable if the following conditions hold:*

(a) The step size $\mu(i,j)$ *is chosen in the following interval:*

$$0<\mu(i,j)<\frac{2}{z^T(i,j)z(i,j)\lambda_{\max}(P)} \quad i,j=0,1,\ldots \quad (29)$$

(b) The 1-D algorithms (19.h) and (19.v) are asymptotically stable , so that the following relation is satisfied:

$$\sum_{k=0}^{\infty}\left\{Trace\left[\theta^h(0,k)\theta^{h^T}(0,k)\right]+Trace\left[\theta^v(k,0)\theta^{v^T}(k,0)\right]\right\}<\infty \quad (30)$$

Theorem 2. *If the conditions of Theorem 1 hold, then the 2-D algorithm (18) is marginal asymptotically stable, too.*

Theorem 3. *The 2-D algorithm (18) will be asymptotically stable, if the following conditions hold:*

(a) This algorithm is stable.

(b) The regressor $z(i,j)$ *is sufficiently general, so that there exist the numbers* n^h *and* n^v, *such that the matrices* $\Psi^h(i,j,n^h)$ *and* $\Psi^v(i,j,n^v)$, *which are given below, have full row rank for any beginning point* (i,j):

$$\Psi^h(i,j,n^h)=\left[z^h(i,j)\vdots z^h(i+1,j)\vdots\cdots\vdots z^h(i+n^h,j)\right] \quad (31.h)$$

$$\Psi^v(i,j,n^v)=\left[z^v(i,j)\vdots z^v(i,j+1)\vdots\cdots\vdots z^v(i,j+n^v)\right] \quad (31.v)$$

6 The algorithm implementation

If one of the independent variables of the underlying 2-D system (for example the horizontal variable i) is bounded, we can use the usual raster scan from left to right and from bottom to top for 2-D data scanning as shown in Fig. 1.

Hence, in this case from (18) and (19) the algorithm implementation will be as follows.

Step 1- *choose the initial conditions* $\hat\theta^h(0,0)$ *and* $\hat\theta^v(0,0)$.

Step 2- for $i=0,1,\ldots,M$ *do:*

$$\hat\theta^h(i+1,0)=\hat\theta^h(i,0)-\mu(i,0)Pe(i,0)z^{h^T}(i,0)$$

$$\hat\theta^v(i+1,0)=\hat\theta^v(i,0)-\mu(i,0)Pe(i,0)z^{v^T}(i,0)$$

end

Step 3- for $j=1,2,3,\ldots$ *do:*

$$\hat\theta^h(0,j)=\hat\theta^h(0,j-1)-\mu(0,j-1)Pe(0,j-1)z^{h^T}(0,j-1)$$

for $i=0,1,\ldots,M$ *do:*

$$\hat{\theta}^h(i+1,j)=\hat{\theta}^h(i,j)-\mu(i,j)Pe(i,j)z^{h^T}(i,j)$$
$$\hat{\theta}^v(i,j)=\hat{\theta}^v(i,j-1)-\mu(i,j-1)Pe(i,j-1)z^{v^T}(i,j-1)$$

end

end

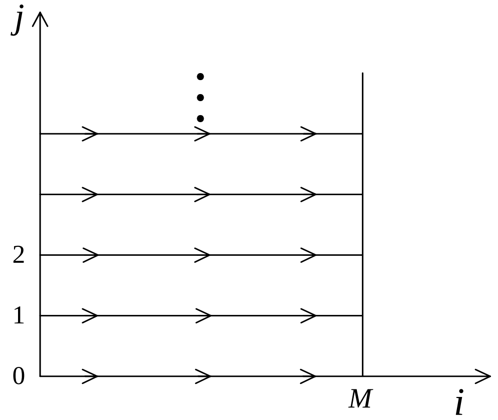

Fig.1: The 2-D data scanning directions (variable i is bounded)

But, if both of the independent variables i and j are unbounded, the usual raster scan, which is shown in Fig. 1, is not usable. In this case, we use the diagonal data scanning as shown in Fig. 2. That is the 2-D data scanning on lines L_k which are defined in (21).

Thus, in this case from (18) and (19) the algorithm implementation will be as follows.

Step 1- *choose the initial conditions* $\hat{\theta}^h(0,0)$ *and* $\hat{\theta}^v(0,0)$.

Step 2- for $k=1$ *do:*

$$\hat{\theta}^h(k,0)=\hat{\theta}^h(k-1,0)-\mu(k-1,0)Pe(k-1,0)z^{h^T}(k-1,0)$$
$$\hat{\theta}^v(k,0)=\hat{\theta}^v(k-1,0)-\mu(k-1,0)Pe(k-1,0)z^{v^T}(k-1,0)$$
$$\hat{\theta}^h(0,k)=\hat{\theta}^h(0,k-1)-\mu(0,k-1)Pe(0,k-1)z^{h^T}(0,k-1)$$
$$\hat{\theta}^v(0,k)=\hat{\theta}^v(0,k-1)-\mu(0,k-1)Pe(0,k-1)z^{v^T}(0,k-1)$$

end

Step 3- for $k=2,3,\ldots$ *do:*

$$\hat{\theta}^h(k,0)=\hat{\theta}^h(k-1,0)-\mu(k-1,0)Pe(k-1,0)z^{h^T}(k-1,0)$$
$$\hat{\theta}^v(k,0)=\hat{\theta}^v(k-1,0)-\mu(k-1,0)Pe(k-1,0)z^{v^T}(k-1,0)$$

for $l=1,\ldots,k-1$ ***do:***

$$\hat{\theta}^h(k-l,l)=\hat{\theta}^h(k-l-1,l)$$
$$-\mu(k-l-1,l)Pe(k-l-1,l)z^{h^T}(k-l-1,l)$$

$$\hat{\theta}^v(k-l,l)=\hat{\theta}^v(k-l,l-1)$$
$$-\mu(k-l,l-1)Pe(k-l,l-1)z^{v^T}(k-l,l-1)$$

end

$$\hat{\theta}^h(0,k)=\hat{\theta}^h(0,k-1)-\mu(0,k-1)Pe(0,k-1)z^{h^T}(0,k-1)$$
$$\hat{\theta}^v(0,k)=\hat{\theta}^v(0,k-1)-\mu(0,k-1)Pe(0,k-1)z^{v^T}(0,k-1)$$

end

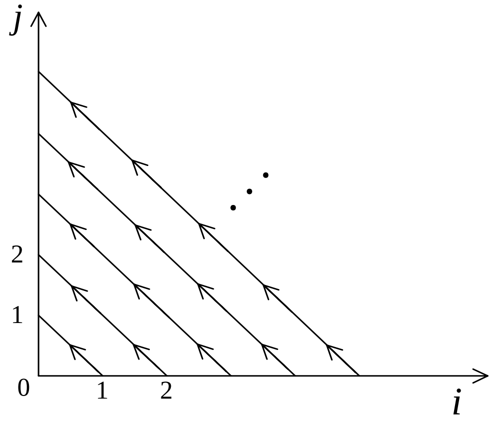

Fig.2: The 2-D data scanning directions (both variables i and j are unbounded)

7 Simulation results

In order to illustrate the performance of the proposed 2-DAPE procedure an example is considered. Suppose a 2-D system, which has five unknown parameters, is described by following ARMA model:

$$y(i,j)=ay(i,j-1)+by(i-1,j)+cy(i-1,j-1)$$
$$+du(i,j)+eu(i-1,j)\qquad i,j=0,1,\ldots$$

The aim is to estimate the system parameters as two-dimensionally, utilizing the measurements of the input signal $u(i,j)$ and the output signal $y(i,j)$.

The system model is converted to the regressor form (1), by the following definitions:

$$\theta=\begin{bmatrix}a & b & c & d & e\end{bmatrix}$$

$$z(i,j)=\left[y(i,j-1)\vdots y(i-1,j)\vdots y(i-1,j-1)\vdots u(i,j)\vdots u(i-1,j)\right]^T$$

We choose the following columnar decomposition:

$$\theta^h=\begin{bmatrix}a & b & c\end{bmatrix},\ \theta^v=\begin{bmatrix}d & e\end{bmatrix}$$
$$z^h(i,j)=\left[y(i,j-1)\vdots y(i-1,j)\vdots y(i-1,j-1)\right]^T$$
$$z^v(i,j)=\left[u(i,j)\vdots u(i-1,j)\right]^T$$

The input signal and the boundary conditions of the system are chosen as:

$$u(i,j)=(-1)^j\sin(\tfrac{\pi i}{15})+(-1)^i\cos(\tfrac{\pi j}{35})\qquad i,j=0,1,\ldots$$
$$y(-1,j)=\sin(\tfrac{\pi j}{20}),\ y(i,-1)=\cos(\tfrac{\pi i}{7}),\ y(-1,-1)=0$$

The true values of system parameters are assumed as follows:

$$a=0.8,\ b=-0.5,\ c=0.4,\ d=2,\ e=-1$$

Considering the Theorem 1, P and $\mu(i,j)$ are selected as:

$$P=1,\ \mu(i,j)=\frac{1}{z^T(i,j)z(i,j)}\qquad i,j=0,1,\ldots$$

It is assumed that the initial conditions of the adjusting algorithms, that is $\hat{\theta}^h(0,0)$ and $\hat{\theta}^v(0,0)$, are as zero.

The obtained simulation results are shown in Figs. 3 and 4 for two among five parameters. Also, the Fig. 5 represents the mean square of the estimation error on the line L_k . It is seen, that the obtained estimation for any parameter is converged to the true value of that parameter, in any manner we go away from the origin $(i, j) = (0,0)$.

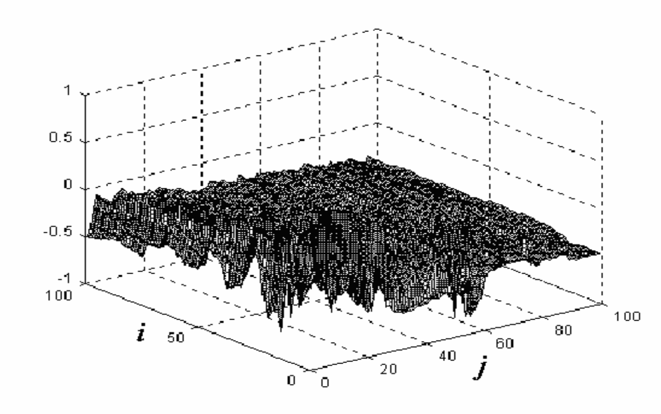

Fig. 3: The obtained estimation for b

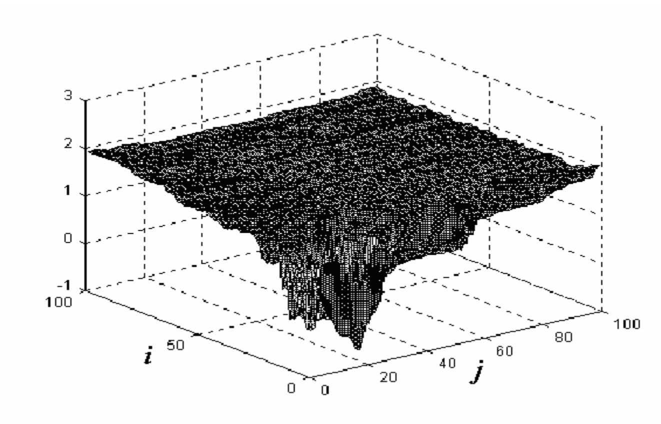

Fig. 4: The obtained estimation for d

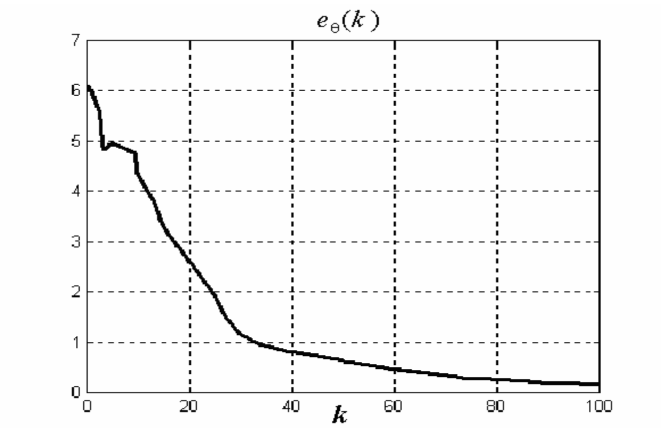

Fig. 5: The mean square of the estimation error on line L_k

8 Conclusions

This paper has extended the adaptive parameter estimation procedure to 2-D systems. The 2-DAPE problem is formulated in a general case, when both of the 2-D space coordinates are possibly unbounded, and the corresponding solution is presented. The solution procedure is based on the identification model and the decomposition of columns of the matrix of system parameters into two blocks.

If θ in (1) has more than two columns, then the various methods will be possible to decompose its columns to the horizontal and vertical blocks. To the authors' experiences, which are obtained from numerous simulations, in this case the optimum decomposition is achieved when the difference between the numbers of columns of the horizontal and vertical blocks is the minimum. Because with this option the convergence rate of the estimation is faster than others.

One can combine the presented 2-D estimation procedure in the paper with any kind of the traditional 2-D controller in order to construct a 2-D adaptive self-tuning controller.

Summarizing, the advantages of the presented procedure in comparison with those already presented in the literature ([9-12]) are as follows:

1- In the presented procedure both of the 2-D space coordinates are possibly unbounded, which is an important improvement in 2-D estimation category.

2- Since the presented estimation method is based on the identification model, if the parameters of the underlying 2-D system to be variant then the parameters of the identification model, which are the estimations of system parameters, can adapt themselves with the system parameters variations and follow their variations.

References

[1] R. P. Roesser, "A Discrete State- Space Model for Linear Image Processing", *IEEE Trans. Automat. Contr.,* **vol. AC-20,** pp. 1-10, (1975).

[2] N. K. Bose, Applied Multidimensional Systems Theory, New York: Van Nostrand Reinhold, (1982).

[3] T. Kaczorek, Two Dimensional Linear Systems, Lecture Notes in Control and Information Science, **vol.68**, Berlin: Springer-Verlag, (1985).

[4] Z. Geng, R. Carroll, J. Xie, "Tow-Dimensional Model and Algorithm for A Class of Iterative Learning Control Systems", *Int. J. Contr.,* **vol. 52**, pp. 833-862, (1990).

[5] D. H. Owens, N. Amann, E. Rogers, M. French, "Analysis of Linear Iterative Learning Control Schemes: a 2-D Repetitive Systems Approach", *Multidimensional Systems and Signal Processing,* **vol. 11**, pp. 125-177, (2000).

[6] C. Richard Johnson, Lectures on Adaptive Parameter Estimation, Prentice-Hall, Englewood Cliffs, New Jersey, (1988).

[7] K. S. Narendra, Stable Adaptive Systems, Prentice-Hall, Englewood Cliffs, New Jersey, (1989).

[8] K. J. Astrom , B. Wittenmark, Adaptive Control, second edition, Addison-Wesely, New York, (1995).

[9] W. P. Heath, P. E. Wellsted, "Self-tuning Prediction and Control for Two-dimensional Processes -Part 2:Parameter Estimation, Set-point Tracking and Offset Handling", Int. J. Control, **vol. 62**, pp. 239-269,(1995).

[10] J. R. Caldas-Pinto, Self-Tuning Filters for Two Dimensions, Ph.D. thesis, Control Systems Center, UMIST, U.K., (1983).

[11] G. R. Wagner, Self-Tuning Algorithms for Two Dimensions Signal Processing, Ph.D. thesis, Control Systems Center, UMIST, U.K., (1987).

[12] W. P. Heat, P. E. Wellsted and M. B. Zarrop, "Forgetting Factors for Two-Dimensional Recursive Estimation", Control System Center, Report no. 724, UMIST, U.K., (May 1990).