Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Convergence of Bias-Eliminating Least Squares Methods for Identification of Dynamic Errors-in-Variables Systems

Torsten Söderström, Mei Hong and Wei Xing Zheng

Abstract— **The problem of dynamic errors-in-variable identification is studied in this paper. We investigate asymptotic convergence properties of the previous bias-eliminating algorithms. We first derive an error dynamic equation for the bias-eliminating parameter estimates. We then show that the asymptotic convergence of the bias-eliminating algorithms is basically determined by the eigenvalue of the largest magnitude of a system matrix in the estimation error dynamic equation. Moreover, the bias-eliminating algorithms possess desired convergence when all the eigenvalues of the system matrix in the estimation error dynamic equation fall strictly inside the unit circle. Given possible divergence of the iteration-type bias-eliminating algorithms under very low SNR (signal-to-noise ratio) values at the system input and output, we re-formulate the bias-elimination problem as a minimization problem associated with a concentrated loss function and develop a variable projection algorithm to efficiently solve the resulting minimization problem. Finally, we illustrate and verify the theoretical results through stochastic simulations.**

I. INTRODUCTION

Identification of dynamic errors-in-variables (EIV) systems where both input and output measurement data are corrupted by noise is a fundamental research problem. It has applications in many practical areas, such as, control engineering, signal processing, image processing, time series analysis, econometrics and so on. A number of approaches have been developed in the fast few decades. For examples, see [1], [2], [3], [4], [5], [6], [7], [8], [9], etc. An overview of different identification methods for the dynamic EIV problem can be found in a recent paper [10].

In this paper we center on the bias-eliminating least squares (BELS) algorithms, which, with a modest computational cost, usually give more accurate estimates than standard instrumental variable (IV) methods for identification of dynamic EIV systems. The BELS algorithms are built upon the bias compensation principle. That is, the variances of the input and output measurement noises are first estimated and the noise-induced bias in the conventional least-squares (LS) parameter estimate is then eliminated so as to achieve estimation consistency. The iterative algorithmic structure of the BELS algorithms lends itself easily to on-line implementation for recursive identification. Since the BELS algorithms involve an iterative estimation procedure, the convergence of such iterative process is necessarily investigated. In [4], [11], attempts were made to analyze the convergence of the BELS algorithms by means of the contraction mapping and fixed point theory. Although the convergence analysis given in [4], [11] is on the track, it is not quite rigorous theoretically. Indeed, it has been observed in practice that when the system input and output are both corrupted by a modest amount of noise, the BELS algorithms are able to converge rapidly. However, the algorithms may be divergent in the presence of high input noise and/or high output noise. Hence, convergence properties of the BELS algorithms are an important yet open question. In this paper we shall investigate asymptotic convergence properties of the BELS algorithms for the case of a large number of data samples.

The outline of the paper is as follows. Section II states the problem of identifying dynamic errors-in-variables systems, and also re-derives a set of key equations utilized by the BELS algorithm in [11]. Section III deduces the error dynamic equation for the BELS parameter estimates. On this basis, a study of the asymptotic convergence properties for the BELS algorithms is presented. Section IV proposes an alternative method, called the variable projection algorithm, for identification of dynamic errors-in-variables systems. Section V presents a numerical simulation study to demonstrate the theoretical analysis. Finally, Section VI concludes the paper with some remarks.

II. THE ERRORS-IN-VARIABLES PROBLEM

Let a linear system to be identified be described by the following model

$$y_0(t) = \frac{B(q^{-1})}{A(q^{-1})} u_0(t), \tag{1}$$

where

$$\begin{aligned} A(q^{-1}) &= 1 + a_1\, q^{-1} + \ldots + a_{na}\, q^{-na} \\ B(q^{-1}) &= b_1\, q^{-1} + \ldots + b_{nb}\, q^{-nb}. \end{aligned} \tag{2}$$

Since the input $u_0(t)$ and the output $y_0(t)$ are usually measured with corrupting noise, the noisy measurements $u(t)$ and $y(t)$ are obtained via

$$\begin{aligned} u(t) &= u_0(t) + \tilde{u}(t) \\ y(t) &= y_0(t) + \tilde{y}(t). \end{aligned} \tag{3}$$

The following general assumptions are adopted:

A1. $A(q^{-1})$ and $B(q^{-1})$ are co-prime polynomials.

This work was supported in part by The Swedish Research Council under contract 621-2002-4671, a Research Grant from the Australian Research Council and a Research Grant from the University of Western Sydney, Australia.

Torsten Söderström and Mei Hong are with the Division of Systems and Control, Department of Information Technology, Uppsala University. P O Box 337, SE-751 05 Uppsala, Sweden. ts@it.uu.se, mhong@it.uu.se

Wei Xing Zheng is with the School of QMMS, University of Western Sydney, Penrith South DC NSW 1797, Australia. w.zheng@uws.edu.au

0-7803-9568-9/05/$20.00 ©2005 IEEE

A2. The system orders na and nb are a priori known.

A3. The noise free input $u_0(t)$ is persistently exciting of sufficient order.

A4. $\tilde{u}(t)$ and $\tilde{y}(t)$ are mutually independent white noise sequences, of zero mean, and variances λ_u and λ_y, respectively. The noise signals are independent of $u_0(t)$.

The aim of the dynamic EIV identification is to estimate the parameter vector

$$\theta = (a_1 \dots a_{na} \quad b_1 \dots b_{nb})^T \tag{4}$$

as well as the noise variances λ_u and λ_y from data records of the noisy signals $\{u(t), y(t)\}_{t=1}^N$.

Let the regressor vector $\varphi(t)$ be defined by

$$\begin{aligned} \varphi(t) &= (-y(t-1) \dots - y(t-na) \\ &\qquad u(t-1) \dots u(t-nb))^T. \end{aligned} \tag{5}$$

It is straightforward to decompose $\varphi(t)$ into a sum of one noise-free term and one noise term:

$$\begin{aligned} \varphi(t) &\stackrel{\Delta}{=} \varphi_0(t) + \tilde{\varphi}(t) \\ &= (-y_0(t-1) \dots - y_0(t-na) \\ &\qquad u_0(t-1) \dots u_0(t-nb))^T \\ &\quad +(-\tilde{y}(t-1) \dots - \tilde{y}(t-na) \\ &\qquad \tilde{u}(t-1) \dots \tilde{u}(t-nb))^T. \end{aligned} \tag{6}$$

We use θ_0 to denote the true value of the parameter vector, and use θ as a general notation for the parameter vector with arbitrary values in what follows. Let the covariance matrix and covariance vector be defined respectively by

$$R_\varphi = E[\varphi(t)\varphi^T(t)], \qquad r_{\varphi y} = E[\varphi(t)y(t)]. \tag{7}$$

For the investigation conducted in this paper, it will suffice to deal with the case of an infinite number of data points (i.e., $N \to \infty$). Hence, there is no need to distinguish between R_φ and its estimate $\frac{1}{N}\sum_{t=1}^{N} \varphi(t)\varphi^T(t)$ from available data points. We will also use the notations R_{φ_0}, $R_{\tilde{\varphi}}$, *etc.* which are compatible and self-explanatory. It is evident that the relation $R_\varphi = R_{\varphi_0} + R_{\tilde{\varphi}}$ holds. Moreover, it follows from Assumption A4 that

$$R_{\tilde{\varphi}} = E[\tilde{\varphi}(t)\tilde{\varphi}^T(t)] = \begin{pmatrix} \lambda_y I_{na} & 0 \\ 0 & \lambda_u I_{nb} \end{pmatrix}. \tag{8}$$

The least squares estimate $\hat{\theta}_{LS}$ of θ can be written as the solution to the normal equations

$$R_\varphi \hat{\theta}_{LS} = r_{\varphi y}. \tag{9}$$

Since

$$\begin{aligned} (R_{\varphi_0} + R_{\tilde{\varphi}})\hat{\theta}_{LS} &= E[\varphi(t)y(t)] \\ &= E[\varphi_0(t)y_0(t)] + E[\tilde{\varphi}(t)\tilde{y}(t)] \\ &= E[\varphi_0(t)\varphi_0^T(t)]\theta_0 \\ &= R_{\varphi_0}\theta_0, \end{aligned} \tag{10}$$

it is easy to see that $\hat{\theta}_{LS}$ is biased. Equation (10) also shows that if the noise variances λ_y and λ_u are known then the bias can be estimated and eliminated:

$$(R_\varphi - R_{\tilde{\varphi}})\hat{\theta}_{BELS} = r_{\varphi y} \tag{11}$$

where $\hat{\theta}_{BELS}$ denotes the bias-eliminated least squares estimate. A number of algorithms have appeared to construct such bias-elimination schemes, see [3], [4], [11], [12]. Here we will examine the algorithm given in [11].

In order to determine λ_y and λ_u (and hence $R_{\tilde{\varphi}}$), two more relations are needed. One such relation can be derived from the minimal value of the least squares criterion:

$$\begin{aligned} V_{LS} &= \min_\theta E[y(t) - \varphi^T(t)\theta]^2 \\ &= E[y(t) - \varphi^T(t)\hat{\theta}_{LS}]^2 \\ &= \lambda_y + E[\varphi_0^T(t)\theta_0 - \varphi^T(t)\hat{\theta}_{LS}]^2 \\ &= \lambda_y + \theta_0^T R_{\varphi_0}\theta_0 + \hat{\theta}_{LS}^T R_\varphi \hat{\theta}_{LS} - 2\hat{\theta}_{LS}^T R_{\varphi_0}\theta_0. \end{aligned}$$

From (9) and (10) it follows that $R_{\varphi_0}\theta_0 = R_\varphi\hat{\theta}_{LS}$, and hence

$$\begin{aligned} V_{LS} &= \lambda_y + \theta_0^T R_\varphi \hat{\theta}_{LS} + \theta_0^T R_{\varphi_0}\hat{\theta}_{LS} - 2\hat{\theta}_{LS}^T R_{\varphi_0}\theta_0 \\ &= \lambda_y + \theta_0^T R_{\bar{\varphi}}\hat{\theta}_{LS}. \end{aligned} \tag{12}$$

Note that (12) can be seen as a linear equation in λ_y and λ_u.

To get also a second relation for λ_y and λ_u, an extended model structure has to be considered. This is similar to the so-called Frisch scheme [3]. For this purpose we introduce extended versions of $\varphi(t)$, θ and θ_0 as

$$\bar{\varphi}(t) = \begin{pmatrix} \varphi(t) \\ \underline{\varphi}(t) \end{pmatrix}, \quad \bar{\theta} = \begin{pmatrix} \theta \\ \underline{\theta} \end{pmatrix}, \quad \bar{\theta}_0 = \begin{pmatrix} \theta_0 \\ 0 \end{pmatrix}. \tag{13}$$

The model extension can, for example, mean that an additional A parameter is appended. In that case,

$$\underline{\varphi}(t) = -y(t-na-1), \qquad \underline{\theta} = a_{na+1}. \tag{14}$$

Another possibility is to append an additional B parameter, leading to

$$\underline{\varphi}(t) = u(t-nb-1), \qquad \underline{\theta} = b_{nb+1}. \tag{15}$$

We stick to the general case here covering (14) and (15) as special cases. We note in passing that although we will for simplicity assume $\underline{\varphi}(t)$ being scalar, it is possible to generalize the analysis to $\underline{\varphi}(t)$ being a vector. The number of new relations derived will be equal to the dimension of $\underline{\varphi}(t)$.

Next consider least squares estimation in the extended linear regression model

$$y(t) = \bar{\varphi}^T(t)\bar{\theta}, \tag{16}$$

which leads to

$$R_{\bar{\varphi}}\hat{\bar{\theta}}_{LS} = r_{\bar{\varphi}y}. \tag{17}$$

Recall that $y(t) = y_0(t) + \tilde{y}(t)$, (3), and $\varphi(t) = \varphi_0(t) + \tilde{\varphi}(t)$, (6). Hence

$$\begin{aligned} R_{\bar{\varphi}}\hat{\bar{\theta}}_{LS} &= r_{\bar{\varphi}_0 y_0} + r_{\tilde{\bar{\varphi}}\tilde{y}} \\ &= R_{\bar{\varphi}_0}\bar{\theta}_0 \\ &= (R_{\bar{\varphi}} - R_{\tilde{\bar{\varphi}}})\bar{\theta}_0. \end{aligned} \tag{18}$$

Note that in the right hand side $R_{\tilde{\bar{\varphi}}}$ varies linearly with λ_y and λ_u. Set

$$H = (0, \dots, 1) \in \mathcal{R}^{na+nb+1}, \tag{19}$$

$$J = \begin{pmatrix} I_{na+nb} \\ 0 \end{pmatrix}, \quad \bar{\theta}_0 = J\theta_0. \tag{20}$$

Recalling that $H\bar{\theta}_0 = 0$, we find that (18) implies

$$\begin{aligned} H\hat{\bar{\theta}}_{LS} &= HR_{\bar{\varphi}}^{-1}(R_{\bar{\varphi}} - R_{\tilde{\bar{\varphi}}})\bar{\theta}_0 \\ &= -HR_{\bar{\varphi}}^{-1}R_{\tilde{\bar{\varphi}}}J\theta_0. \end{aligned} \tag{21}$$

Set

$$\lambda = (\lambda_y \quad \lambda_u)^T. \tag{22}$$

The matrix $R_{\tilde{\varphi}}$ given in (8) may be denoted by $R_{\tilde{\varphi}}(\lambda)$. Summing up so far we have derived the following equations for determining θ and λ:

$$R_\varphi\hat{\theta}_{LS} = [R_\varphi - R_{\tilde{\varphi}}(\lambda)]\theta, \tag{23}$$

$$V_{LS} = \lambda_y + \hat{\theta}_{LS}^T R_{\tilde{\varphi}}(\lambda)\theta, \tag{24}$$

$$H\hat{\bar{\theta}}_{LS} = -HR_{\bar{\varphi}}^{-1}R_{\tilde{\bar{\varphi}}}(\lambda)J\theta. \tag{25}$$

Equations (23)-(25) turn out to be bilinear in the unknowns θ and λ. That is, they are linear in θ and linear in λ. There are different ways to solve these equations. We examine two different algorithms in the next two sections.

In this paper we only consider the white measurement noise cases. For more general colored noise cases, some other BELS algorithms have been proposed, see [5]. The basic idea remains the same except that (24) and (25) need to be changed by other equations.

III. THE BELS ALGORITHM

In order to remove the bias from $\hat{\theta}_{LS}$, the BELS algorithm [11], [12], [4] aims to solve the equations (23)-(25), which consists of the following steps.

Initialization: Set

$$k = 0, \qquad \hat{\theta}^{(0)} = \hat{\theta}_{LS}. \tag{26}$$

Iteration: Solve

$$\begin{cases} V_{LS} &= \lambda_y + \hat{\theta}_{LS}^T R_{\tilde{\varphi}}(\lambda)\hat{\theta}^{(k)}, \\ H\hat{\bar{\theta}}_{LS} &= -HR_{\bar{\varphi}}^{-1}R_{\tilde{\bar{\varphi}}}(\lambda)J\hat{\theta}^{(k)}. \end{cases} \tag{27}$$

with respect to λ. Denote the outcome by $\hat{\lambda}^{(k)}$, then compute

$$\hat{\theta}^{(k+1)} = \hat{\theta}_{LS} + R_\varphi^{-1}R_{\tilde{\varphi}}(\hat{\lambda}^{(k)})\hat{\theta}^{(k)}. \tag{28}$$

Increase k by 1, $k := k + 1$, and iterate until convergence.

It has been observed in practice that this algorithm converges quickly when the amounts of noise on the input and output sides are modest. No convergence may occur if the amount of noise is very high. To analyze the situation we will examine the local convergence property of the algorithm.

Due to the fact that the underlying equations are bilinear, it is possible to write the algorithm in the following compact form

$$\begin{aligned} &\hat{\theta}^{(k+1)} = c + C(\hat{\lambda}^{(k)})\hat{\theta}^{(k)}, \\ &F(\hat{\theta}^{(k)})\hat{\lambda}^{(k)} = f. \end{aligned} \tag{29}$$

Explicit expressions for c, $C(\lambda)$, $F(\theta)$ and f are given in [13]. Furthermore, $C(\lambda)$ and $F(\theta)$ are affine functions, namely

$$\begin{aligned} C(\lambda) &= C_0 + \sum_{i=1}^{2} C_i\lambda_i, \\ F(\theta) &= F_0 + \sum_{j=1}^{na+nb} F_j\theta_j. \end{aligned} \tag{30}$$

Introduce also the matrices

$$\begin{aligned} &\bar{C}(\theta) = (C_1\theta \quad C_2\theta), && (na+nb)|2 \\ &\bar{F}(\lambda) = (F_1\lambda \quad F_2\lambda \quad \dots \quad F_{na+nb}\lambda). && 2|(na+nb) \end{aligned} \tag{31}$$

It follows easily from (30) and (31) that

$$\begin{aligned} C(\lambda)\theta &= C_0\theta + \bar{C}(\theta)\lambda, \\ F(\theta)\lambda &= F_0\lambda + \bar{F}(\lambda)\theta. \end{aligned} \tag{32}$$

The true parameter vectors θ_0 and λ_0 certainly form a stationary solution of the recursion (29). Hence

$$\begin{aligned} &\theta_0 = c + C(\lambda_0)\theta_0, \\ &F(\theta_0)\lambda_0 = f. \end{aligned} \tag{33}$$

Close to this stationary solution we can model the algorithm behavior by linearization. Set

$$\tilde{\theta}^{(k)} = \hat{\theta}^{(k)} - \theta_0, \qquad \tilde{\lambda}^{(k)} = \hat{\lambda}^{(k)} - \lambda_0. \tag{34}$$

We have

$$\tilde{\theta}^{(k+1)} \approx C(\lambda_0)\tilde{\theta}^{(k)} + \bar{C}(\theta_0)\tilde{\lambda}^{(k)}, \tag{35}$$

$$F(\theta_0)\tilde{\lambda}^{(k)} + \bar{F}(\lambda_0)\tilde{\theta}^{(k)} \approx 0, \tag{36}$$

see appendix for details. It implies

$$\tilde{\theta}^{(k+1)} = G\tilde{\theta}^{(k)}, \tag{37}$$

with

$$G = C(\lambda_0) - \bar{C}(\theta_0)F^{-1}(\theta_0)\bar{F}(\lambda_0). \tag{38}$$

In the above analysis, one must of course assume that the matrix $F(\theta_0)$ is nonsingular. If this is not the case one would anyway expect great difficulties when solving the last equation of (29).

Equation (37) represents an error dynamics equation for the BELS parameter estimates, with G being a system matrix. From (37) one can directly see that the algorithm converges locally (that is, for initial values close enough to the true values), precisely when the system matrix G in (37) has all eigenvalues strictly inside the unit circle. The above analysis is based on infinite data size. Since G changes continuously when $N \to \infty$, the conclusion of the analysis holds also as long as N is large enough. In [13] more detailed expressions for the system matrix G are provided. There, we also prove the following lemma.

Lemma 1: The matrix G has always one eigenvalue equal to zero. In case $Eu_0^2(t)$ becomes large, the eigenvalues of G all satisfy

$$\lambda_j(G) = O(1/Eu_0^2(t)). \tag{39}$$

■

The lemma shows that for high signal-to-noise ratios, all eigenvalues of G will be small, and the convergence of the algorithm will be quick.

Numerical examinations, which will partly be presented in Section V, confirm indeed that G has small eigenvalues for large SNR on the input and output sides. Moreover, for moderate SNR, if the maximum absolute eigenvalue of G is smaller than 0.9, then convergence of the BELS algorithm can be achieved. However, there are also cases where for very low SNR values, G can have eigenvalues outside the unit circle. Even if the corresponding SNR values are unrealistically small, this nevertheless shows that the BELS algorithm does not *always* converge. This observation is one of the reasons for examining alternative methods for solving (23)-(25).

IV. A VARIABLE PROJECTION ALGORITHM

We will now derive an alternative algorithm for solving the equations (23)-(25). We write them compactly as

$$M(\lambda)\theta = m(\lambda), \tag{40}$$

where $M(\lambda)$ is an $(na+nb+2)|(na+nb)$ matrix, and $m(\lambda)$ is an $(na+nb+2)$ vector. Both $M(\lambda)$ and $m(\lambda)$ are affine functions of λ. For explicit expressions of $M(\lambda)$ and $m(\lambda)$, please see appendix.

One way to solve (40) is to reformulate this as an optimization problem

$$(\hat{\theta}, \hat{\lambda}) = \arg\min_{\theta,\lambda} \parallel M(\lambda)\theta - m(\lambda) \parallel^2 . \tag{41}$$

Assuming that $M(\lambda)$ has full column rank (equal to $na+nb$), the minimization with respect to θ is simple:

$$\begin{aligned} \hat{\theta} = \hat{\theta}(\hat{\lambda}) &= M^{\dagger}(\lambda)m(\lambda) \\ &= [M^T(\lambda)M(\lambda)]^{-1}M^T(\lambda)m(\lambda), \end{aligned} \tag{42}$$

where $M^{\dagger}$ denotes the pseudo-inverse of M. The full rank assumption is generically fulfilled, see [13]. Inserting the expression (42) into (41) leads to the concentrated loss function

$$\begin{aligned} V(\lambda) &= \parallel m(\lambda) - M(\lambda)M^{\dagger}(\lambda)m(\lambda) \parallel^2 \\ &= \parallel [I - M(\lambda)M^{\dagger}(\lambda)]m(\lambda) \parallel^2 \\ &= m^T(\lambda)[I - M(\lambda)M^{\dagger}(\lambda)]m(\lambda). \end{aligned} \tag{43}$$

In (43), $I - M(\lambda)M^{\dagger}(\lambda)$ is the orthogonal projection onto the null space of $M^T(\lambda)$. To minimize $V(\lambda)$ with respect to λ is often referred to as a variable projection problem, in particular when $m(\lambda)$ does not depend on λ. Such minimization problems occur frequently in sensor array processing as well as in many other applications [14], [15], [16]. It has also been proposed for an errors-in-variables problem using a different setup [17]. Here, the algorithm will run as follows.

1. Choose an initial value of λ, say $\lambda = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

2. Minimize $V(\lambda)$ in (43) to get

$$\hat{\lambda} = \arg\min_{\lambda} V(\lambda) \tag{44}$$

TABLE I

PARAMETERS FOR DIFFERENT CASES

	a_1	a_2	b_1	b_2	d	SNR_u	SNR_y	φ
S_1	-0.8		1.0		-0.5	0.97	4.55	$-y(t-2)$
S_2	-0.8		1.0		-0.5	0.97	4.55	$u(t-2)$
S_3	-0.8		1.0		-0.5	10.97	14.55	$-y(t-2)$
S_4	-0.8		1.0		-0.5	10.97	14.55	$u(t-2)$
S_5	-0.8		1.0		-0.9	2.58	6.44	$-y(t-2)$
S_6	-0.8		1.0		-0.9	2.58	6.44	$u(t-2)$
S_7	-1.5	0.7	1.0	0.5	-0.5	0.97	12.65	$-y(t-3)$
S_8	-1.5	0.7	1.0	0.5	-0.5	0.97	12.65	$u(t-3)$

3. Check that $V(\hat{\lambda}) \approx 0$ (within the rounding error effects), so that the minimization has ended successfully.

4. Compute $\hat{\theta}(\hat{\lambda})$ from (42).

V. NUMERICAL ILLUSTRATION

We now conduct some numerical investigations to further illustrate and support the preceding theoretical analysis.

A. Eigenvalues of G and local convergence of the BELS algorithm

Consider the linear system

$$A(q^{-1})y_0(t) = B(q^{-1})u_0(t), \tag{45}$$

where the undisturbed input is modeled as an AR(1) process

$$u_0(t) + du_0(t-1) = e(t), \tag{46}$$

and $e(t)$ is a zero mean white noise. Further let the noise signals $\tilde{u}(t)$ and $\tilde{y}(t)$ be mutually uncorrelated white noise signals. Let λ_e, λ_y and λ_u be the variances of the signals $e(t)$, $\tilde{y}(t)$ and $\tilde{u}(t)$, respectively. For each simulation 1000 realizations are done.

A number of first and second order systems with different parameters are considered (see Table I). In each case, both input and output noise have unit variances. System parameters are estimated by using the BELS algorithm as described in Section III.

From (37), we get

$$\parallel \tilde{\theta}^{(k+1)} \parallel \;\leq\; \max|\lambda_k(G)| \parallel \tilde{\theta}^{(k)} \parallel, \tag{47}$$

where G is assumed to have precisely one eigenvalue that is the largest in magnitude. When k is large, the equality in (47) almost establishes, which gives

$$\frac{\parallel \tilde{\theta}^{(k+1)} \parallel}{\parallel \tilde{\theta}^{(k)} \parallel} = \frac{\parallel \hat{\theta}^{(k+1)} - \theta_0 \parallel}{\parallel \hat{\theta}^{(k)} - \theta_0 \parallel} \approx \max|\lambda_k(G)|, \tag{48}$$

where $\hat{\theta}^{(k+1)}$ and $\hat{\theta}^{(k)}$ are the estimates of θ at iteration $k+1$ and k, respectively, and θ_0 is the true θ. Considering that N is finite in a realization, then $\hat{\theta}_N^{(k)} \to \theta_N^* \neq \theta_0$. The local convergence speed is calculated by the ratio

$$\alpha(k) = \frac{\parallel \hat{\theta}^{(k+1)} - \theta_N^* \parallel}{\parallel \hat{\theta}^{(k)} - \theta_N^* \parallel}, \tag{49}$$

where θ_N^* is evaluated as $\hat{\theta}_N^{(k)}$ with large k.

TABLE II

LOCAL CONVERGENCE RATE OF BELS ALGORITHMS AND MAXIMUM ABSOLUTE EIGENVALUE OF G FOR DIFFERENT SYSTEMS.

System	$\max \lvert\lambda_k(G)\rvert$	$\lim_{k\to\infty} \alpha(k)$ $(N = 1000)$	$\lim_{k\to\infty} \alpha(k)$ $(N = 500)$
S_1	0.039	0.039± 0.014	0.040± 0.019
S_2	0.155	0.156± 0.049	0.158± 0.076
S_3	0.004	0.004± 0.001	0.004± 0.016
S_4	0.018	0.018± 0.008	0.020± 0.010
S_5	0.041	0.040± 0.006	0.040± 0.008
S_6	0.024	0.029± 0.005	0.030± 0.007
S_7	0.641	0.626± 0.074	0.685± 0.092
S_8	0.652	0.611± 0.171	0.659± 0.346

For comparison we also calculate the system matrix G by formula (38) and its eigenvalues. The maximum absolute value of the eigenvalues of G and the stationary value of the $\alpha(k)$ for different systems are shown in Table II (both the mean and standard deviation of $\lim_{k\to\infty} \alpha(k)$ are listed for the cases $N = 1000$ and $N = 500$). We can see that the values of $\max |\lambda_k(G)|$ and $\lim_{k\to\infty} \alpha(k)$ are very similar for each case, and that there is a larger spread of $\alpha(k)$ between the realizations when N is decreased.

B. Behavior of eigenvalues of G

The results of Table II indicate that the larger SNR on the input and output sides, the smaller the eigenvalues for the system matrix G. In general, for modest SNR, $\max |\lambda(G)| \leq 0.9$. However, for some low SNR cases, $\max |\lambda(G)|$ may be larger than one. For example, consider a second order system with $A(q^{-1}) = 1-1.5q^{-1}+0.7q^{-2}$, $B(q^{-1}) = q^{-1}+0.5q^{-2}$, and the undisturbed input as $u_0(t) = \frac{1}{1-0.5q^{-1}}e(t)$. When the output SNR equals to -4dB, and the input SNR equals 0dB, the relevant $\max |\lambda(G)|$ is 1.308 or 1.625 for adding an additional B parameter or an A parameter, respectively. Both values are larger than one.

In practice, $\max |\lambda(G)| < 0.9$ is needed to make the BELS algorithm converge. In Figure 1, we plot the corresponding output SNR versus input SNR curve to maintain the $\max |\lambda(G)| = 0.9$ for the same second order system as described above. It manifests that the BELS algorithm will keep convergent only when input SNR and output SNR are within the marked area. Further, it is worthy to notice that, for the same system parameters, the converging area for adding extra A parameter is generally larger than that of adding B parameters. This result implies that adding extra A parameter is a better choice when using the proposed algorithms.

C. Use of the variable projection algorithm

The variable projection algorithm and the BELS algorithm were applied to the same realization, i.e. under the same data and noise conditions for eight systems listed in Section V-A. In the variable projection algorithm, equation (44) is solved by Nelder-Mead minimization method using the *fminsearch* function in Matlab. The estimation results display that the same $\hat{\theta}$ is achieved for each realization, and the deviation is the same for the two algorithms.

TABLE III

PARAMETERS FOR DIFFERENT LOW SNR CONDITIONS

System	SNR_u	SNR_y	λ_e	φ
S_9	-6.02	5.66	0.2	$-y(t-3)$
S_{10}	-6.02	5.66	0.2	$u(t-3)$
S_{11}	-9.03	2.65	0.1	$-y(t-3)$
S_{12}	-9.03	2.65	0.1	$u(t-3)$

TABLE IV

ESTIMATION RESULTS BY USING THE BELS AND THE VARIABLE PROJECTION ALGORITHMS FOR DIFFERENT LOW SNR CONDITIONS.

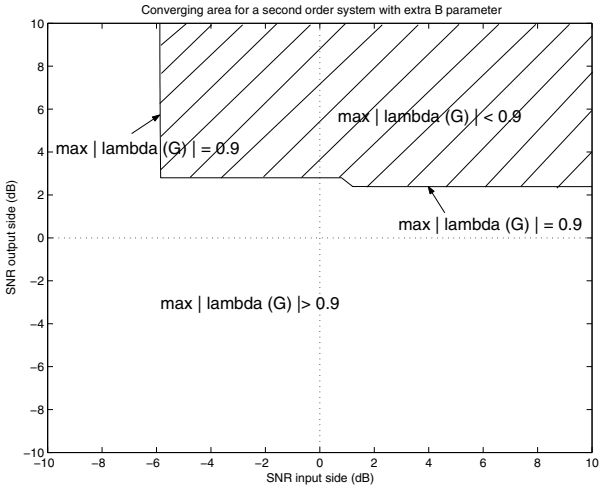
System	$\max \lvert\lambda(G)\rvert$	$\lvert\lvert\hat{\theta} - \theta_0\rvert\rvert$	$\hat{\lambda}_y$	$\hat{\lambda}_u$
S_9(BELS)	0.8991	0.0587	0.9916	1.0408
S_9(VP)	0.8991	0.0447	0.9908	1.0423
S_{10}(BELS)	0.9032	252.14	0.3859	1.0809
S_{10}(VP)	0.9032	0.0451	0.9847	0.9349
S_{11}(BELS)	0.9469	690.38	0.0062	1.0466
S_{11}(VP)	0.9469	0.0126	0.9419	1.0081
S_{12}(BELS)	0.9492	144.32	0.4978	1.0659
S_{12}(VP)	0.9492	0.0804	0.9634	0.9633

Besides, the performances of these two algorithms under the low SNR conditions are analyzed. Consider system S_9-S_{12} that have the same parameters as system S_7-S_8, but with much lower input signal power, see Table III. The simulation results (see Table IV) show that for the variable projection algorithm, the convergence is typically achieved even if $\max |\lambda(G)| > 0.9$. But this is not true for the BELS algorithm which converge only for system S_9 with $\max |\lambda(G)| < 0.9$. In short, the variable projection algorithm is more robust (convergence is more easily achieved) than the BELS algorithm in the low SNR case.

VI. CONCLUSIONS

In this paper asymptotic convergence properties of the previously proposed BELS algorithms for dynamic EIV identification have been investigated for large sample data cases. It has been shown that the convergence of the BELS

Fig. 1. Converging area of input and output SNR for a second order system with parameters $A = [1 \quad -1.5 \quad 0.7]$, $B = [1 \quad 0.5]$, $d = 0.5$ and $\lambda_e = 1$. For points in the shadowed areas $\max |\lambda(G)| \leq 0.9$.



algorithms is essentially determined by eigenvalues of a system matrix in the estimation error dynamic equation. When the system matrix G has all its eigenvalues well inside the unit circle, the BELS algorithms can converge fast. This is the case when there is high SNR on the system input and output sides. For moderate SNR, as long as the maximum absolute eigenvalue of G is smaller than 0.9, the convergence of the BELS algorithm is also achieved. However, in the presence of very low SNR, the system matrix G may have eigenvalues outside the unit circle, leading to the possible divergence of the BELS algorithms. To overcome this issue, the bias-eliminating problem has been re-formulated as a minimization problem associated with a concentrated loss function. Subsequently, a variable projection algorithm has been proposed to perform unbiased parameter estimation for the dynamic EIV problem. The numerical examples have been presented to support the theoretical analysis.

APPENDIX

A. Derivation of (35) and (36)

Using (30)-(34) we have

$$\begin{aligned}\tilde{\theta}^{(k+1)} &= c + C(\lambda_0+\tilde{\lambda}^{(k)})[\theta_0+\tilde{\theta}^{(k)}] - \theta_0 \\ &= c + [C(\lambda_0) + \sum_{i=1}^{2} C_i \tilde{\lambda}_i^{(k)}][\theta_0+\tilde{\theta}^{(k)}] - \theta_0 \\ &= c + C(\lambda_0)\theta_0 - \theta_0 + C(\lambda_0)\tilde{\theta}^{(k)} + \sum_{i=1}^{2} C_i \tilde{\lambda}_i^{(k)}\theta_0 \\ &\quad + \text{second order term} \\ &= C(\lambda_0)\tilde{\theta}^{(k)} + \bar{C}(\theta_0)\tilde{\lambda}^{(k)} + \text{second order term}. \end{aligned} \tag{50}$$

$$\begin{aligned} f &= F(\theta_0+\tilde{\theta}^{(k)})[\lambda_0+\tilde{\lambda}^{(k)}] \\ &= [F(\theta_0) + \sum_{j=1}^{na+nb} F_j \tilde{\theta}_j^{(k)}][\lambda_0+\tilde{\lambda}^{(k)}] \\ &= F(\theta_0)\lambda_0 + F(\theta_0)\tilde{\lambda}^{(k)} + \sum_{j=1}^{na+nb} F_j \tilde{\theta}_j^{(k)}\lambda_0 \\ &\quad + \text{second order term} \\ &= f + F(\theta_0)\tilde{\lambda}^{(k)} + \bar{F}(\lambda_0)\tilde{\theta}^{(k)} + \text{second order term}. \end{aligned} \tag{51}$$

Neglecting the second order terms (which can be done when the iteration is close to the true values), (50) and (51) results in (35) and (36).

B. Expressions for $M(\lambda)$ and $m(\lambda)$

From (23) it holds

$$\theta = \hat{\theta}_{LS} + R_\varphi^{-1} R_{\tilde{\varphi}}(\lambda)\theta. \tag{52}$$

From (25) it follows that

$$H R_{\bar{\varphi}}^{-1}(J_y\lambda_y + J_u\lambda_u)J\theta = -H\hat{\tilde{\theta}}_{LS}, \tag{53}$$

where

$$J_y = \begin{pmatrix} I_{na} & & \\ & 0_{nb} & \\ & & 1 \end{pmatrix} \quad J_u = \begin{pmatrix} 0_{na} & & \\ & I_{nb} & \\ & & 0 \end{pmatrix}.$$

While (24) can be written as

$$\lambda_y + \lambda_y \hat{a}_{LS}^T a + \lambda_u \hat{b}_{LS}^T b = V_{LS}. \tag{54}$$

Putting (52), (53) and (54) together, we arrive at

$$M(\lambda) = \begin{pmatrix} I_{na+nb} - R_\varphi^{-1} R_{\tilde{\varphi}}(\lambda) \\ H R_{\tilde{\varphi}}^{-1}(J_y\lambda_y + J_u\lambda_u)J \\ (\lambda_y \hat{a}_{LS}^T \quad \lambda_u \hat{b}_{LS}^T) \end{pmatrix}, \tag{55}$$

$$m(\lambda) = \begin{pmatrix} \hat{\theta}_{LS} \\ -H\hat{\tilde{\theta}}_{LS} \\ V_{LS} - \lambda_y \end{pmatrix}. \tag{56}$$

REFERENCES

[1] T. Söderström, "Identification of stochastic linear systems in presence of input noise," *Automatica*, vol. 17, pp. 713–725, 1981.

[2] K. V. Fernando and H. Nicholson, "Identification of linear systems with input and output noise: the Koopmans–Levin method," *IEE Proceedings, Part D*, vol. 132, no. 1, pp. 30–36, January 1985.

[3] S. Beghelli, R. Guidorzi, and U. Soverini, "The Frisch scheme in dynamic system identification," *Automatica*, vol. 26, pp. 171–176, 1990.

[4] W. X. Zheng and C. B. Feng, "Unbiased parameter estimation of linear systems in presence of input and output noise," *International Journal of Adaptive Control and Signal Processing*, vol. 3, pp. 231–251, 1989.

[5] ——, "Identification of a class of dynamic errors–in–variables models," *International Journal of Adaptive Control and Signal Processing*, vol. 6, pp. 431–440, 1992.

[6] J. K. Tugnait, "Stochastic system identification with noisy input using cumulant statistics," *IEEE Transactions on Automatic Control*, vol. AC-37, pp. 476–485, 1992.

[7] J. K. Tugnait and Y. Ye, "Stochastic system identification with noisy input-output measurement using polyspectra," *IEEE Transactions on Automatic Control*, vol. AC-40, pp. 670–683, 1995.

[8] P. Guillaume, R. Pintelon, and J. Schoukens, "Robust parametric transfer function estimaton using complex logarithmic frequency response data," *IEEE Transactions on Automatic Control*, vol. 40, pp. 1180–1190, 1995.

[9] P. Stoica, M. Cedervall, and A. Eriksson, "Combined instrumental variable and subspace fitting approach to parameter estimation of noisy input-output systems," *IEEE Transactions on Signal Processing*, vol. 43, pp. 2386–2397, 1995.

[10] T. Söderström, U. Soverini, and K. Mahata, "Perspectives on errors-in-variables estimation for dynamic systems," *Signal Processing*, vol. 82, no. 8, pp. 1139–1154, August 2002.

[11] W. X. Zheng, "Transfer function estimation form noisy input and output data," *International Journal of Adaptive Control and Signal Processing*, vol. 12, pp. 365–380, 1998.

[12] ——, "On least–squares identification of stochastic linear systems with noisy input–output data," *International Journal of Adaptive Control and Signal Processing*, vol. 13, pp. 131–143, 1999.

[13] T. Söderström, M. Hong, and W. Zheng, "Convergence properties of bias-eliminating algorithms for errors-in-variables identification," *International Journal of Adaptive Control and Signal Processing*, 2005, to appear.

[14] G. H. Golub and V. Pereyra, "The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate," *SIAM J. Numerical Analysis*, vol. 10, pp. 413–432, 1973.

[15] ——, "Separable nonlinear least squares: The variable projection method and its applications," *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, April 2003.

[16] M. Viberg and B. Ottersten, "Sensor array processing based on subspace fitting," *IEEE Transactions on Signal Processing*, vol. 39, no. 5, pp. 1110–1121, 1991.

[17] M. Ekman, "Identification of linear systems with errors in variables using separable nonlinear least squares," in *16th IFAC World Congress*, Prague, Czech Republic, July 2005.