Extending the Frisch scheme for errors-in-variables identification to correlated output noise

Torsten Söderström
Division of Systems and Control
Department of Information Technology
Uppsala University, P O Box 337, SE-751 05 Uppsala, Sweden.
Email: ts@it.uu.se

June 19, 2006

Abstract

Several estimation methods have been proposed for identifying errors-in-variables systems, where both input and output measurements are corrupted by noise. One of the promising approaches is the so called Frisch scheme. In its standard form it is designed to handle white measurement noise on the input and output sides. As the output noise comprises both effects of measurement errors and of process disturbances, it is much more realistic to allow correlated output noise. It is described in the paper how the Frisch scheme can be extended to such cases.

1 Introduction

For system identification of linear dynamic systems from noise–corrupted output measurements there are many different methods available, see, for example, [6], [12]. Estimation of the parameters for linear dynamic systems when also the input is affected by noise ('errors–in–variables', or EIV, models) is recognized as a more difficult problem. The class of scientific disciplines which makes use of such representations is very broad, as proved by the several applications collected in [15], [16], such as time series modelling, array signal processing for direction–of–arrival estimation, blind channel equalization, multivariate calibration in analytical chemistry, image processing, astronomical data reduction, etc. In case of static systems, errors–in–variables representations are closely related to other well–known topics such as *latent variables* models and *factor* models [4].

Some comparisons between different approaches for errors-in-variables modelling of dynamic systems are given in [9] and [11] and references therein. The so called Frisch scheme is one of the more interesting approaches for the errors-in-variables identification. It has its roots in a classical algebraic estimation problem, [3], where a regression problem was treated. It has been proposed for identifying dynamic systems in [1] and was further elaborated in [2]. An analysis concerning the accuracy of the estimates obtained using the Frisch scheme was provided in [8].

2 Problem statement and notional setup

2.1 Setup

Consider the system depicted in Figure 1 with noise-corrupted input and output measurements.

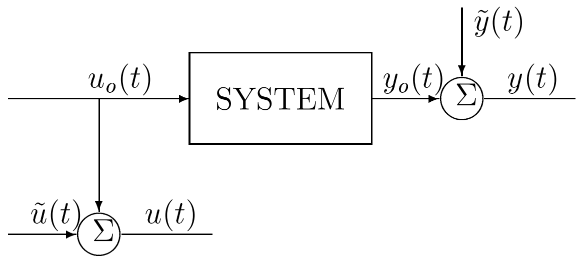

Figure 1: The basic setup for an error-in-variables problem.

The noise–free input is denoted by $u_o(t)$ and the undisturbed output by $y_o(t)$. They are linked through the linear difference equation

$$A(q^{-1})\, y_o(t) = B(q^{-1})\, u_o(t), \tag{2.1}$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the backward shift operator q^{-1}, *i.e.* $q^{-1}\,x(t) = x(t-1)$ *etc.* More precisely,

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na}, \\ B(q^{-1}) &= b_1 q^{-1} + \cdots + b_{nb} q^{-nb}. \end{aligned} \tag{2.2}$$

We assume that the observations are corrupted by additive measurement noises $\tilde{u}(t)$ and $\tilde{y}(t)$ of zero mean. The available signals are thus of the form

$$\begin{aligned} u(t) &= u_o(t) + \tilde{u}(t), \\ y(t) &= y_o(t) + \tilde{y}(t). \end{aligned} \tag{2.3}$$

The general problem is to determine the system characteristics, *i.e.* the transfer function

$$G(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})}. \tag{2.4}$$

In other words, the estimation problem is as follows. Given the noisy input-output data $u(1), y(1), \ldots u(N), y(N)$, determine an estimate of the parameter vector

$$\theta = (a_1 \ldots a_{na}\ b_1 \ldots b_{nb})^\top. \tag{2.5}$$

2.2 Assumptions

In order to proceed, some general assumptions must be introduced.

A1. The dynamic system (1) is asymptotically stable, *i.e.* $A(z)$ has all zeros outside the unit circle.

A2. All the system modes are observable and controllable, *i.e.* $A(z)$ and $B(z)$ have no common factors.

A3. The polynomial degrees na and nb are *a priori* known.

A4. The processes $\tilde{u}(t)$ and $\tilde{y}(t)$ are mutually uncorrelated, and uncorrelated with the noise–free signals $u_o(t)$ and $y_o(t)$.

A5. The sequences $\tilde{u}(t)$ and $\tilde{y}(t)$ are zero–mean white noise sequences with variances λ_u and λ_y, respectively.

A6. The true input $u_o(t)$ is a zero–mean stationary ergodic random signal, that is persistently exciting at least of order $na + nb$.

Remark. The assumptions as listed above are standard for many EIV estimation methods. They apply, in particular, for the Frisch scheme. As $\tilde{y}(t)$ in (2.3) should comprise not only sensor noise but also effects of process disturbances, Assumption **A5** is not very realistic. The purpose of the paper is to examine means to extend the Frisch scheme to situations where $\tilde{y}(t)$ is correlated in an arbitrary way.

2.3 Notations

The following notations will be convenient.

We introduce the regressor vector

$$\varphi(t) = (-y(t-1) \ldots - y(t-na)\ u(t-1) \ldots u(t-nb))^\top, \tag{2.6}$$

which is compatible with the parameter vector (2.5). Further, we will use the conventions:

- θ_o denotes the true parameter vector, and $\hat{\theta}$ its estimate.
- Similarly, we let $A_o(q^{-1}),\ B_o(q^{-1}),\ \lambda_u^o,\ \lambda_y^o$ denote the true values of $A(q^{-1})$, $B(q^{-1}),\ \lambda_u,\ \lambda_y$, respectively.
- $\varphi_o(t)$ denotes the noise-free part of the regressor vector:

$$\varphi_o(t) = (-y_o(t-1) \ldots - y_o(t-na)\ u_o(t-1) \ldots u_o(t-nb))^\top. \tag{2.7}$$

- $\tilde{\varphi}(t)$ denotes the noise-contribution to the regressor vector. This means that

$$\tilde{\varphi}(t) = (-\tilde{y}(t-1) \ldots - \tilde{y}(t-na)\ \tilde{u}(t-1) \ldots \tilde{u}(t-nb))^\top. \tag{2.8}$$

Sometimes it is very convenient to add a leading element to θ and to φ. For this reason we introduce the extended regressor vector as

$$\overline{\varphi}(t) = \begin{pmatrix} -y(t) \\ \varphi(t) \end{pmatrix}, \tag{2.9}$$

and the extended parameter vector

$$\overline{\theta} = \begin{pmatrix} 1 \\ \theta \end{pmatrix}. \tag{2.10}$$

At other times it is useful to work with partitioned parameter and regression vectors. For this reason we introduce also the partitions of θ and $\varphi(t)$ as

$$\theta = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_{na} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_{nb} \end{pmatrix}, \tag{2.11}$$

and

$$\varphi(t) = \begin{pmatrix} \varphi_y(t) \\ \varphi_u(t) \end{pmatrix},$$

where

$$\varphi_y(t) = \begin{pmatrix} -y(t-1) \\ \vdots \\ -y(t-na) \end{pmatrix}, \quad \varphi_u(t) = \begin{pmatrix} u(t-1) \\ \vdots \\ u(t-nb) \end{pmatrix}. \tag{2.12}$$

Extended versions of the partitioned vectors will also be handy:

$$\overline{\theta} = \begin{pmatrix} \overline{\mathbf{a}} \\ \mathbf{b} \end{pmatrix}, \quad \overline{\mathbf{a}} = \begin{pmatrix} 1 \\ \mathbf{a} \end{pmatrix}, \tag{2.13}$$

$$\overline{\varphi}(t) = \begin{pmatrix} \overline{\varphi}_y(t) \\ \varphi_u(t) \end{pmatrix}, \quad \overline{\varphi}_y(t) = \begin{pmatrix} -y(t) \\ \varphi_y(t) \end{pmatrix}. \tag{2.14}$$

Cross-covariance matrices between two vectors $x(t)$ and $y(t)$ are denoted as

$$R_{xy} = \mathsf{E}x(t)y^\top(t), \tag{2.15}$$

and their natural estimates are denoted as

$$\hat{R}_{xy} = \frac{1}{N}\sum_{t=1}^{N} x(t)y^\top(t). \tag{2.16}$$

The covariance matrices are often partitioned in a way compatible with the partitioning of the vectors. For example,

$$\hat{R}_{\overline{\varphi}} = \begin{pmatrix} \hat{R}_{\overline{\varphi}_y} & \hat{R}_{\overline{\varphi}_y\varphi_u} \\ \hat{R}_{\varphi_u\overline{\varphi}_y} & \hat{R}_{\varphi_u} \end{pmatrix}. \tag{2.17}$$

3 The Frisch scheme

As a background we review the Frisch scheme here for the case when $\tilde{y}(t)$ is white noise, and thus satisfies Assumption **A5**.

First note that

$$\overline{\varphi}_o^\top(t)\overline{\theta}_o = -A_o(q^{-1})y_o(t) + B_o(q^{-1})u_o(t) = 0. \tag{3.1}$$

Further it holds that

$$R_\varphi = R_{\varphi_o} + R_{\tilde{\varphi}}, \qquad R_{\overline{\varphi}} = R_{\overline{\varphi}_o} + R_{\tilde{\overline{\varphi}}}. \tag{3.2}$$

It follows from (3.1) that

$$R_{\overline{\varphi}_o}\overline{\theta}_o = \mathsf{E}\overline{\varphi}_o\overline{\varphi}_o^\top\overline{\theta}_o = \mathbf{0}. \tag{3.3}$$

Hence the matrix $R_{\overline{\varphi}_o}$ is singular (positive semidefinite), with at least one eigenvalue equal to zero. The corresponding eigenvector is $\overline{\theta}_o$. One can show that under the general assumptions **A2** and **A6**, the matrix $R_{\overline{\varphi}_o}$ will in fact have only one eigenvalue in the origin.

The noise covariance matrix has a simple structure, as

$$R_{\tilde{\overline{\varphi}}} = \begin{pmatrix} \lambda_y I_{na+1} & \mathbf{0} \\ \mathbf{0} & \lambda_u I_{nb} \end{pmatrix}. \tag{3.4}$$

The relation (3.3) is the basis for the Frisch method. The idea is to have appropriate estimates of the noise variances and then determine the parameter vector θ from

$$\left(\hat{R}_{\overline{\varphi}} - \hat{R}_{\tilde{\overline{\varphi}}}\right)\hat{\overline{\theta}} = \mathbf{0}. \tag{3.5}$$

Assume for the time being that some estimate $\hat{\lambda}_u$ of the input noise variance is available. Then the output noise variance $\hat{\lambda}_y$ is determined so that the matrix appearing in (3.5) is singular. More specifically, see [8]:

$$\hat{\lambda}_y = \lambda_{\min}\left(\hat{R}_{\overline{\varphi}_y} - \hat{R}_{\overline{\varphi}_y\varphi_u}\left(\hat{R}_{\varphi_u} - \hat{\lambda}_u I_{nb}\right)^{-1}\hat{R}_{\varphi_u\overline{\varphi}_y}\right), \tag{3.6}$$

where $\lambda_{\min}(R)$ denotes the smallest eigenvalue of R.

The estimate of the parameter vector θ is determined by solving

$$\left(\hat{R}_\varphi - \begin{pmatrix} \hat{\lambda}_y I_{na} & \mathbf{0} \\ \mathbf{0} & \hat{\lambda}_u I_{nb} \end{pmatrix}\right)\hat{\theta} = \hat{r}_{\varphi y}, \tag{3.7}$$

which is indeed the last $na + nb$ equations in (3.5). By (3.6), $\hat{\lambda}_y$ will be a function of $\hat{\lambda}_u$.

What remains is to determine $\hat{\lambda}_u$. Different alternatives have been proposed:

In [1], the function $\hat{\lambda}_y(\hat{\lambda}_u)$ is evaluated both for the nominal model and for an extended model, adding one A or one B parameter (or both). The two functions correspond to curves in the $(\hat{\lambda}_u, \hat{\lambda}_y)$ plane. The curves will ideally intersect in one unique point, which defines the estimates. In case $\hat{R}_{\overline{\varphi}}$ would be replaced by its true value $R_{\overline{\varphi}}$ a situation as displayed in Figure 2 would be obtained. Curve

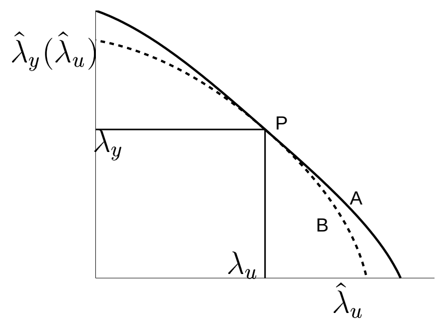

Figure 2: Illustration of the principle for Frisch estimation.

A corresponds to the true model order, while curve B applies for the increased model order. The coordinates of the common point P give precisely the true noise variances $\lambda_u, \ \lambda_y$.

For a finite data set the situation is less ideal, and there is not a distinct point P where the curves A and B share a common point. We refer to [1], [13] for more detailed aspects on how this type of the Frisch scheme can be implemented.

Another alternative is to compute residuals, and compare their statistical properties with what can be predicted from the model. This alternative was proposed in [2], and is presented here in a slightly more general form. Define the residuals

$$\varepsilon(t, \hat{\theta}) = \hat{A}(q^{-1})y(t) - \hat{B}(q^{-1})u(t), \tag{3.8}$$

and compute sample covariance elements

$$\hat{r}_\varepsilon(k) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t, \hat{\theta})\varepsilon(t+k, \hat{\theta}). \tag{3.9}$$

Compute also theoretical covariance elements $\hat{r}_{\varepsilon_o}(k)$ based on the model

$$\varepsilon_o(t) = \hat{A}(q^{-1})\hat{\tilde{y}}(t) - \hat{B}(q^{-1})\hat{\tilde{u}}(t), \tag{3.10}$$

where $\hat{\tilde{y}}(t)$ and $\hat{\tilde{u}}(t)$ are independent white noise sequences with

$$\mathsf{E}\left[\hat{\tilde{y}}^2(t)\right] = \hat{\lambda}_y, \qquad \mathsf{E}\left[\hat{\tilde{u}}^2(t)\right] = \hat{\lambda}_u. \tag{3.11}$$

Next, define a criterion for comparing $\{\hat{r}_\varepsilon(k)\}$ and $\{\hat{r}_{\varepsilon_o}(k)\}$. A fairly general way to do this is to take

$$V_N(\hat{\lambda}_u) = \delta^\top W \delta, \tag{3.12}$$

where W is a user chosen, positive definite weighting matrix, and the vector δ is

$$\delta = \begin{pmatrix} \hat{r}_\varepsilon(1) - \hat{r}_{\varepsilon_o}(1) \\ \vdots \\ \hat{r}_\varepsilon(m) - \hat{r}_{\varepsilon_o}(m) \end{pmatrix}. \tag{3.13}$$

The maximum lag m used in (3.13) is a user choice. The estimate $\hat{\lambda}_u$ is determined as the minimizing element of the criterion

$$\hat{\lambda}_u = \arg\min_{\lambda_u} V_N(\lambda_u), \tag{3.14}$$

so

$$\frac{\partial V_N}{\partial \lambda_u}_{|\lambda_u = \hat{\lambda}_u} = 0. \tag{3.15}$$

In summary the Frisch scheme algorithm consists of the equations (3.6), (3.7) and (3.15). In its implementation, there is an optimization over one variable, $\hat{\lambda}_u$, in (3.14). In the evaluation of the loss function $V_N(\hat{\lambda}_u)$, also (3.6) and (3.7) are used to get $\hat{\lambda}_y$ and $\hat{\theta}$, respectively, as functions of $\hat{\lambda}_u$.

4 Bias-compensating least squares

The Frisch scheme can indeed be interpreted as a bias-compensating least squares method, [9].

When the least squares method is applied to the linear regression model

$$y(t) = \varphi^\top(t)\theta + \varepsilon(t). \tag{4.1}$$

the estimate $\hat{\theta}$ will be biased due to the presence of measurement noises $\tilde{u}(t)$ and $\tilde{y}(t)$ in the data. The least squares (LS) estimate of θ using the model (4.1) is

$$\hat{\theta}_{\mathrm{LS}} = \hat{R}_\varphi^{-1}\hat{r}_{\varphi y} \to R_\varphi^{-1} r_{\varphi y}, \qquad N \to \infty. \tag{4.2}$$

It holds that

$$R_\varphi = R_{\varphi_o} + R_{\tilde{\varphi}}, \quad r_{\varphi y} = r_{\varphi_o y_o} = R_{\varphi_o}\theta_o. \tag{4.3}$$

Therefore the LS estimate $\hat{\theta}_{\mathrm{LS}}$ satisfies

$$R_\varphi \hat{\theta}_{\mathrm{LS}} = [R_\varphi - R_{\tilde{\varphi}}]\theta_o, \tag{4.4}$$

and $\hat{\theta}_{\mathrm{LS}}$ is biased due to the term $R_{\tilde{\varphi}}$.

The principle for bias-compensated least squares (BCLS) is to adjust the least squares estimate for this effect. The adjusted estimate will be

$$\hat{\theta}_{\mathrm{BCLS}} = \left[\hat{R}_\varphi - \hat{R}_{\tilde{\varphi}}\right]^{-1} \hat{r}_{\varphi y}, \tag{4.5}$$

where the noise term $\hat{R}_{\tilde{\varphi}}$ has to be estimated in some way. Note that (4.5) is indeed the same equation as (3.7).

Assuming **A5** applies ($\tilde{u}(t), \tilde{y}(t)$ both white noise), two more equations are needed to determine the unknowns $\theta,\ \lambda_u,\ \lambda_y$. For the Frisch scheme, these two equations are precisely (3.6) and (3.15).

There are many other variants of bias-compensating least squares, see references in [9]. The compensated normal equations (4.5) apply to all variants, but different variants are tied to different sets of the two additional equations.

One of the popular bias-compensating schemes is BELS (bias-eliminating least squares), [17]. Introduce the asymptotic least squares criterion

$$V_{\mathrm{LS}}(\theta) = \mathsf{E}[y(t) - \varphi^\top(t)\theta]^2 \tag{4.6}$$

Then, as shown in [17],

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

From (4.3) it follows that $R_{\varphi_0}\theta_0 = R_\varphi\hat{\theta}_{\mathrm{LS}}$, and hence

$$\begin{aligned} V_{\mathrm{LS}}(\hat{\theta}_{\mathrm{LS}}) &= \lambda_y + \theta_0^\top R_\varphi\hat{\theta}_{\mathrm{LS}} + \theta_0^\top R_{\varphi_0}\hat{\theta}_{\mathrm{LS}} - 2\hat{\theta}_{\mathrm{LS}}^\top R_{\varphi_0}\theta_0 \\ &= \lambda_y + \theta_0^\top R_{\tilde{\varphi}}\hat{\theta}_{\mathrm{LS}} \\ &= \lambda_y + \lambda_y \overline{\mathbf{a}}_o^\top \hat{\overline{\mathbf{a}}}_{\mathrm{LS}} + \lambda_u \mathbf{b}_o^\top \hat{\mathbf{b}}_{\mathrm{LS}}. \end{aligned} \tag{4.7}$$

Note that (4.7) can be seen as a linear equation in λ_y and λ_u.

It is striking to see that the Frisch scheme implies a similar (but not identical!) relation. In this case we have

$$\begin{aligned} V_{\mathrm{LS}}(\hat{\theta}) &= \frac{1}{N}\sum_{t=1}^{N}[\overline{\varphi}^\top(t)\hat{\overline{\theta}}]^2 = \hat{\overline{\theta}}^\top \hat{R}_{\overline{\varphi}}\,\hat{\overline{\theta}} \\ &= \hat{\overline{\theta}}^\top \hat{R}_{\tilde{\overline{\varphi}}}\hat{\overline{\theta}} = \hat{\lambda}_y \hat{\overline{\mathbf{a}}}^\top \hat{\overline{\mathbf{a}}} + \hat{\lambda}_u \hat{\mathbf{b}}^\top \hat{\mathbf{b}}. \end{aligned} \tag{4.8}$$

Equation (3.5) is utilized here. Note the similarity between (4.7) and (4.8).

Remark. The relation (4.8) can be written as

$$\hat{r}_\varepsilon(0) = \hat{r}_{\varepsilon_o}(0), \tag{4.9}$$

and this is the reason why no zero lag element is included in the vector δ, (3.13).

5 Extensions to correlated output noise

When extending the Frisch scheme to correlated output noise we first focus in this section how to modify the constituting equations. The algorithmic details will be dealt with in the following sections.

For correlated output noise we have to modify the unknowns. Needless to say, the parameter vector θ is kept, as well as the input noise variance λ_u. The output noise variance λ_y will be substituted by a covariance sequence:

$$\rho_y = \begin{pmatrix} r_{\tilde{y}}(0) & \ldots & r_{\tilde{y}}(k) \end{pmatrix}^\top. \tag{5.1}$$

We keep it open so far what value of k to choose.

The noise parameters λ_u and ρ_y can be treated in different ways. One pragmatic way is to regard them as nuisance parameters, and for simplicity not impose any constraints. They are then regarded just as intermediate vehicles in order to arrive at an estimator of the parameter vector θ.

A more elaborated way is to impose constraints reflecting the physical meaning of these parameters. As λ_u is a variance, it must hold that $\lambda_u \geq 0$. In the case of a white output noise we would also have $\lambda_y \geq 0$. For the correlated output case, it is more complicated to handle the precise conditions on ρ_y. Some necessary conditions are that the noise covariance matrix

$$R_{\tilde{\varphi}} = \begin{pmatrix} r_{\tilde{y}}(0) & \dots & r_{\tilde{y}}(na-1) & & & \\ \vdots & \ddots & & & & \\ r_{\tilde{y}}(na-1) & \dots & r_{\tilde{y}}(0) & & & \\ & & & \lambda_u & & \\ & & & & \ddots & \\ & & & & & \lambda_u \end{pmatrix} = \begin{pmatrix} R_{\tilde{\varphi}_y} & \mathbf{0} \\ \mathbf{0} & \lambda_u I_{nb} \end{pmatrix} \tag{5.2}$$

must be positive definite, and thus also the matrix $\hat{R}_{\tilde{\varphi}_y}$. For a more stringent treatment, a parameterization that gives precisely necessary and sufficient conditions for ρ_y to be the beginning of a covariance sequence is given in [14]. However, the problem under study then becomes more severely nonlinear, and will be more complicated to handle.

We find directly from (5.1) and (5.2) that a necessary condition on the maximal lag k is

$$k \geq na - 1. \tag{5.3}$$

When extending the Frisch scheme we keep the modified normal equations (4.5) and the 'Frisch equation'

$$\hat{R}_{\overline{\varphi}} - \hat{R}_{\tilde{\overline{\varphi}}} \ \text{ singular.} \tag{5.4}$$

The relation (5.4) will now be rewritten as giving $\hat{\lambda}_u$ as a function of $\hat{\rho}_y$. Similarly to (3.6), the explicit relation will be

$$\hat{\lambda}_u = \lambda_{\min}\left(\hat{R}_{\varphi_u} - \hat{R}_{\varphi_u \overline{\varphi}_y}\left(\hat{R}_{\overline{\varphi}_y} - \hat{R}_{\tilde{\overline{\varphi}}_y}(\hat{\rho}_y)\right)^{-1} \hat{R}_{\overline{\varphi}_y \varphi_u}\right). \tag{5.5}$$

Now consider the modified normal equations. For correlated output noise, they will read

$$\left(\hat{R}_{\varphi} - \hat{R}_{\tilde{\varphi}}\right)\hat{\theta} = \hat{r}_{y\varphi} - \hat{r}_{\tilde{y}\tilde{\varphi}}. \tag{5.6}$$

Here $\hat{R}_{\tilde{\varphi}}$ depends on $r_{\tilde{y}}(0), \dots, r_{\tilde{y}}(na-1)$, while $\hat{r}_{\tilde{y}\tilde{\varphi}}$ depends on $r_{\tilde{y}}(1), \dots, r_{\tilde{y}}(na)$. We thus must always require

$$k \geq na. \tag{5.7}$$

In the next subsections we investigate some approaches for what equations to use in addition to (5.4) and the normal equations.

5.1 Comparing residual functions

A first attempt is to keep the previous approach to compare sample and theoretical covariances of the residuals. More specifically, try to set

$$\delta = 0, \tag{5.8}$$

where δ is given in (3.13).

When computing the covariance element $\hat{r}_{\varepsilon_o}(m)$, one can note that it depends on the covariance elements $r_{\tilde{y}}(0), \ldots, r_{\tilde{y}}(na+m)$. We must therefore choose

$$k \geq na + m. \tag{5.9}$$

However, in this case we have

$$\begin{array}{rl} \#\text{ of unknowns}: & (na+nb)+1+(k+1) \\ \#\text{ of equations}: & (na+nb)+1+m \end{array} \tag{5.10}$$

For this to be compatible, we get

$$m \geq k+1 \geq na+m+1. \tag{5.11}$$

The first inequality in (5.11) follows as we must have at least as many equations as unknowns. The second equality follows after invoking (5.9). Obviously, (5.11) is a contradiction. It means that this attempt to extend the Frisch scheme can never be successful. No matter how we choose m, we will always end up with more unknowns than equations.

5.2 Correlating residuals with past outputs

Another attempt to extend the Frisch scheme is to cross-correlate past outputs with residuals. In this case we have

$$\mathsf{E}\begin{pmatrix} y(t-na-1) \\ \vdots \\ y(t-na-m) \end{pmatrix}\big(A(q^{-1})y(t) - B(q^{-1})u(t)\big) = \ldots \tag{5.12}$$

which gives m equations. The right hand side will have a contribution that depends on ρ_y. The largest lag appears in the last equation of (5.12) and leads to the condition

$$k \geq na + m. \tag{5.13}$$

Similarly to the previous subsection, by analysing the number of equations and the number of unknowns we conclude that also this attempt of extending the Frisch scheme can never work.

5.3 Correlating residuals with past inputs

A third attempt is to cross-correlate past inputs with residuals. This idea leads to the equations

$$\mathsf{E}\begin{pmatrix} u(t-nb-1) \\ \vdots \\ u(t-nb-m) \end{pmatrix}\big(A(q^{-1})y(t)-B(q^{-1})u(t)\big)=0. \tag{5.14}$$

Now we can achieve a more favourable situation compared to the previous attempts, as the output noise covariance vector ρ_y does not appear in (5.14).

A more detailed analysis show that using (5.14) as the third equation for an extended Frisch approach leads to (5.10) and hence $k+1 \leq m$. Combining this with (5.7) we finally make the choices

$$\begin{array}{rcl} k & \geq & na, \\ m & \geq & k+1 \geq na+1. \end{array} \tag{5.15}$$

6 Algorithms

According to the previous section, the relevant equations to use are

- The compensated normal equations, (4.5),
- The 'Frisch equation', (5.5),
- Crosscorrelation of residuals and past inputs, (5.14).

For later purposes, we write (5.14) as

$$\begin{array}{rcl} r_{z\varepsilon} & = & \mathsf{E}z(t)\varepsilon(t,\theta) \\ & = & \mathsf{E}\begin{pmatrix} u(t-nb-1) \\ \dots \\ u(t-nb-m) \end{pmatrix}\big(y(t)-\varphi^\top(t)\theta\big)=0. \end{array} \tag{6.1}$$

Remark. For BELS similar equations apply. The second equation above is to be replaced by (4.7). That the third equation applies for BELS is shown in [5].

We will in what follows present two different algorithms.

6.1 Algorithm 1

In this case we treat the elements of the $(k+1)$-dimensional vector ρ_y as independent variables. Further, note that λ_u and θ are functions of ρ_y by (5.5) and (4.5), respectively. To determine ρ_y we optimize $\| r_{z\varepsilon}(\theta) \|$ over ρ_y, in the sense

$$\hat{\rho}_y = \arg \min_{\rho_y} \| r_{z\varepsilon}(\theta(\rho_y)) \|^2 \tag{6.2}$$

This algorithm thus involves a $(k+1)$-dimensional optimization, with potential difficulties of convergence problems. The optimization can be carried out with or without invoking constraints on λ_u and ρ_y.

6.2 Algorithm 2

The idea here is to reduce the dimensionality of the optimization problem, and instead solving two one-dimensional subproblems. To do so, we split the compensated normal equations into two parts, and write the total set of equations as

$$\begin{aligned}
\left(\hat{R}_{\varphi_y} - \hat{R}_{\tilde{\varphi}_y}(\hat{\rho}_y) \quad -\hat{R}_{\varphi_y\varphi_u} \right) \hat{\theta} &= \hat{r}_{\varphi_y y} - \hat{r}_{\tilde{\varphi}_y \tilde{y}}(\hat{\rho}_y), && (6.3) \\
\left(-\hat{R}_{\varphi_u\varphi_y} \quad \hat{R}_{\varphi_u} - \hat{\lambda}_u I_{nb} \right) \hat{\theta} &= \hat{r}_{\varphi_u y}, && (6.4) \\
\hat{R}_{\overline{\varphi}} - \hat{R}_{\tilde{\overline{\varphi}}} \ &\text{singular} && (6.5) \\
\left(-\hat{R}_{z\varphi_y} \quad \hat{R}_{z\varphi_u} \right) \hat{\theta} &= \hat{r}_{zy}. && (6.6)
\end{aligned}$$

Here, the compensated normal equations (4.5) are rewritten as (6.3), (6.4). Further, (6.5), (6.6) are just the same as (5.4) and (5.14), respectively.

Remark. For the particular case $m = na + nb$, (6.6) is nothing but the standard instrumental variable estimate for errors-in-variables problems, [7].

The number of equations above are na for (6.3), nb for (6.4), 1 for (6.5) and $na+1$ for (6.6).

It is now very useful to see that the vector ρ_y appears only in (6.3) and (6.5). Furthermore, equations (6.4) and (6.6) gives $nb + m = na + nb + 1$ equations for $na + nb + 1$ unknowns ($\hat{\theta}$ and $\hat{\lambda}_u$). This observation leads to the following two steps algorithm.

Step 1. Solve (6.4) and (6.6) with respect to $\hat{\theta}$ and $\hat{\lambda}_u$. Note that these equations can be written more compactly as

$$\left(\boldsymbol{R} - \hat{\lambda}_u \begin{pmatrix} \boldsymbol{0} & I_{nb} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \right) \hat{\theta} = \boldsymbol{r}, \tag{6.7}$$

where

$$\boldsymbol{R} = \begin{pmatrix} -\hat{R}_{\varphi_u\varphi_y} & \hat{R}_{\varphi_u} \\ -\hat{R}_{z\varphi_y} & \hat{R}_{z\varphi_u} \end{pmatrix}, \qquad \boldsymbol{r} = \begin{pmatrix} \hat{r}_{\varphi_u y} \\ \hat{r}_{zy} \end{pmatrix} \tag{6.8}$$

are of dimensions $(na + nb + 1)|(na + nb)$ and $(na + nb + 1)|1$, respectively.

Equation (6.7) is bilinear in $\hat{\theta}$ and $\hat{\lambda}_u$. It can be treated in various ways. Here we propose to use a variable projection algorithm, as in [10]. Set

$$J = \begin{pmatrix} \boldsymbol{0} & I_{nb} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}. \tag{6.9}$$

We characterize the sought solution as

$$\min_{\hat{\theta}, \hat{\lambda}_u} \| \ (\boldsymbol{R} - \hat{\lambda}_u J)\hat{\theta} - \boldsymbol{r} \ \|^2 . \tag{6.10}$$

For given $\hat{\lambda}_u$, solve (6.10) in a least squares sense with respect to $\hat{\theta}$ to get

$$\hat{\theta} = (\boldsymbol{R} - \hat{\lambda}_u J)^{\dagger} \boldsymbol{r}, \tag{6.11}$$

where $R^\dagger$ denotes the pseudoinverse of R. Next $\hat{\lambda}_u$ is found by minimizing the concentrated loss function

$$\begin{aligned} V_2(\hat{\lambda}_u) &= \| (\boldsymbol{R} - \hat{\lambda}_u J)(\boldsymbol{R} - \hat{\lambda}_u J)^\dagger \boldsymbol{r} - \boldsymbol{r} \|^2 \\ &= \boldsymbol{r}^\top \boldsymbol{r} - \boldsymbol{r}^\top (\boldsymbol{R} - \hat{\lambda}_u J) \left[(\boldsymbol{R} - \hat{\lambda}_u J)^\top (\boldsymbol{R} - \hat{\lambda}_u J) \right]^{-1} (\boldsymbol{R} - \hat{\lambda}_u J)^\top \boldsymbol{r}. \end{aligned} \tag{6.12}$$

This is a one-dimensional optimization problem. The sought solution should satisfy $V_2(\hat{\lambda}_u) = 0$ within rounding errors. This gives a way to check if the computed solution is reasonable, or the search routine can be expected to be stuck in a local minimum point.

Remark. If we only care about the estimation of θ, then BELS and the Frisch scheme will be give exactly the same estimate for the case considered here of white input noise and correlated output noise. In both cases Step 2 can be disregarded as long as there is no interest in determining ρ_y explicitly.

Step 2.

As $\hat{\theta}$ and $\hat{\lambda}_u$ are computed in Step 1, only $\hat{\rho}_y$ remains as unknown to be determined in Step 2. Should only the parameter vector θ be of interest, Step 2 can be skipped altogether.

To determine ρ_y, use the remaining equations (6.3) and (6.5). A further inspection reveals that (6.3) is in fact linear (or rather, affine) in ρ_y. We can therefore in principal rewrite (6.3) as

$$F \rho_y = f, \tag{6.13}$$

where F is a matrix of dimension $na|(na+1)$. Using (6.13) we find that

$$\rho_y = \alpha h_0 + F^\dagger f, \tag{6.14}$$

where h_0 is a vector spanning the null space of F. Using (6.14) we have a one-dimensional parameterization of ρ_y using the scalar α. Inserting this into (5.5) we can finally determine the remaining unknown α as

$$\hat{\alpha} = \arg \min_{\alpha} \| \hat{\lambda}_u - \lambda_{\min} \left(\hat{R}_{\varphi_u} - \hat{R}_{\varphi_u \overline{\varphi}_y} \left(\hat{R}_{\overline{\varphi}_y} - \hat{R}_{\tilde{\overline{\varphi}}_y} (\alpha h_0 + F^\dagger f) \right)^{-1} \hat{R}_{\overline{\varphi}_y \varphi_u} \right) \| . \tag{6.15}$$

Hence also Step 2 of the algorithm has as its core a one-dimensional optimization problem.

7 Numerical examples

In this section we discuss by means of some numerical examples what performance to expect of the algorithms previously introduced.

For both algorithms it is possible to include constraints on $\hat{\lambda}_u$ and $\hat{\rho}_y$. Note in particular that there is no guarantee that the vector ρ_y obtained from (6.14) does satisfy the constraints. The largest possible value of the input noise variance $\hat{\lambda}_u$ is obtained when $R_{\tilde{\overline{\varphi}}_y} = 0$, and requiring that the matrix $\hat{R}_{\overline{\varphi}_o}$ is positive definite. This leads to the condition

$$0 < \hat{\lambda}_u < \lambda_{\min}\left(\hat{R}_{\varphi_u} - \hat{R}_{\varphi_u\overline{\varphi}_y}\hat{R}^{-1}_{\overline{\varphi}_y}\hat{R}_{\overline{\varphi}_y\varphi_u}\right) \tag{7.1}$$

The output noise covariance sequence $\hat{\rho}_y$ should similarly be such that

$$0 < \hat{R}_{\tilde{\overline{\varphi}}_y}(\rho_y) < \hat{R}_{\overline{\varphi}_y} - \hat{R}_{\overline{\varphi}_y\varphi_u}\hat{R}^{-1}_{\varphi_u}\hat{R}_{\varphi_u\overline{\varphi}_y} \tag{7.2}$$

In particular for Algorithm 1, where the optimization is over the vector ρ_y, an initial estimate is needed for the iterative search. In the implemented algorithm, first the EIV problem with modelling the output noise to be white was considered, leading to an estimate $\hat{\lambda}_y$. Then $\hat{\rho}_y$ was initiated as

$$\rho_y = \begin{pmatrix} \hat{\lambda}_y & 0 & \dots & 0 \end{pmatrix}^\top . \tag{7.3}$$

The estimate $\hat{\rho}_y$ was then determined by minimizing the criterion (6.2) using a penalty function, i.e. adding a large term to the criterion function when the constraint (7.2) was not satisfied.

Example 1. Occasionally the estimates produced by Algorithm 1 did show a weak performance. For a simple case, it was possible to attribute this performance to convergence to a false minimum point. There is in fact no guarantee that (6.2) even has a unique global minimum. As a simple illustration, consider the system given by

$$\begin{aligned} y_o(t) &= 2u_o(t-1) \\ u_0(t) &= \tfrac{1+0.7q^{-1}}{1-0.5q^{-1}}e(t), \qquad \mathsf{E}e^2(t) = 1, \\ \tilde{y}(t) &= \tfrac{1}{1-0.5q^{-1}}e_y(t), \qquad \mathsf{E}e_y^2(t) = 4. \end{aligned} \tag{7.4}$$

In this case $na = 0$ and ρ_y will be a scalar quantity. To avoid analysing what happens for a particular realization, we can examine the asymptotic case when $N = \infty$. This means that all sample covariance elements tend to their expected values. The loss function in (6.2) is plotted versus the scalar unknown ρ_y in Figure 3 for $m = 1$ and $m = 2$. We see easily from the figure that for the case $m = 1$ there are two (isolated) global minimum points. While the true parameter values correspond to $\rho_y = 5.33$, it turns out that the described initialization gives convergence to the other global minimum point at $\rho_y = 4.2$. Fortunately, if we increase m to 2 (although $m = 1$ should be sufficient according to (5.15)) the situation improves, and there is a unique minimum to the loss function. ■

Example 2. A Monte Carlo simulation study was carried out for a number of systems and algorithms. The implemented algorithms are as follows.

- **A1** Algorithm 1. The constraints on ρ_y, (7.2) are treated using a penalty term on the loss function.
- **A2F** Algorithm 2. The search for λ_u in Step 1 is done within the interval described by (7.1). For Step 2, an analog interval is derived for $r_{\tilde{y}}(0)$ and then

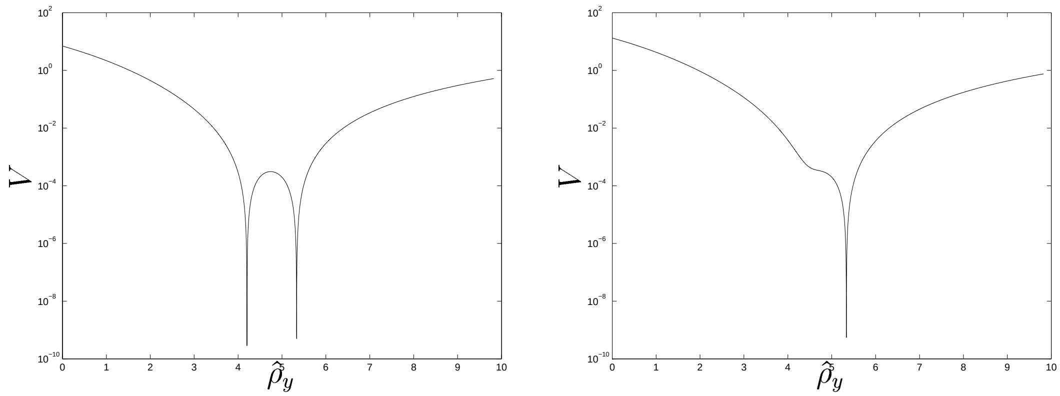

Figure 3: Behaviour of the loss function V in (6.2) versus ρ_y. Left: $m = 1$, right: $m = 2$.

transformed into an interval for α using (6.14). The constraint (7.2) for ρ_y is disregarded.

A2B This is Algorithm 2 with details as described for **A2F**. The difference is in Step 2 where the BELS equation (4.7) was used instead of (6.5). In this way a linear system of equations was used to determine ρ_y. No constraint on ρ_y or even $r_{\tilde{y}}(0)$ was applied.

Six different systems were tried. In all cases 100 realizations, each of length $N = 10000$ were used. In several cases the optimization algorithms produce warnings. As the computed estimates of θ still are reasonable, the results shown below are based on using all realizations.

System 1. This system is given by

$$\begin{aligned} &y_o(t) = \tfrac{2q^{-1}}{1-0.8q^{-1}} u_o(t), \qquad u_o(t) = \tfrac{1+0.7q^{-1}}{1-0.5q^{-1}} e(t), \quad \mathsf{E} e^2(t) = 1, \\ &\mathsf{E}\tilde{u}^2(t) = 1, \qquad \tilde{y}(t) = \tfrac{1}{1-0.7q^{-1}} e_y(t), \qquad \mathsf{E} e_y^2(t) = 1, \end{aligned} \tag{7.5}$$

leading to signal-to-noise ratios on the input and output sides, respectively,

$$\text{SNR}_u = 5.82 \text{ dB}, \qquad \text{SNR}_y = 7.63 \text{ dB}. \tag{7.6}$$

The results obtained are displayed in Table 1.

The results obtained for System 1 are quite good in terms of estimated θ and λ_u. The estimates of ρ_y are very bad, though. For Algorithm 2 it is known that the quality of $\hat{\rho}_y$ does not influence the properties of $\hat{\theta}$. For Algorithm 1, there is no decoupling in the estimation phase: rather, $\hat{\theta}, \hat{\lambda}_u$ and $\hat{\rho}_y$ are computed together as a solution to one joint optimization problem. It is therefore striking that also for Algorithm 1 the estimates $\hat{\theta}$ are very accurate although $\hat{\rho}_y$ is generally bad.

System 2. In this case the system is of second order, and

$$\begin{aligned} &y_o(t) = \tfrac{2q^{-1}+q^{-2}}{1-1.5q^{-1}+0.7q^{-2}} u_o(t), \qquad u_o(t) = \tfrac{1+0.7q^{-1}}{1-0.5q^{-1}} e(t), \quad \mathsf{E} e^2(t) = 1, \\ &\mathsf{E}\tilde{u}^2(t) = 1, \qquad \tilde{y}(t) = \tfrac{1}{1-0.7q^{-1}} e_y(t), \qquad \mathsf{E} e_y^2(t) = 4, \end{aligned} \tag{7.7}$$

Table 1: Results obtained for System 1. Mean and standard deviations over 100 realizations.

Parameter	True value	A1 mean	A1 std	A2F mean	A2F std	A2B mean	A2B std
a_1	-0.8	-0.800	0.003	-0.800	0.005	-0.800	0.005
b_1	2.0	1.997	0.024	2.000	0.033	2.000	0.033
$\rho_y(1)$	1.96	1.08	0.49	1.16	0.86	1.11	0.95
$\rho_y(2)$	1.37	0.08	0.49	0.16	0.94	0.12	1.00
λ_u	1	0.996	0.031	1.000	0.037	1.000	0.037

leading to signal-to-noise ratios on the input and output sides, respectively,

$$\text{SNR}_u = 5.82 \text{ dB}, \qquad \text{SNR}_y = 11.48 \text{ dB}. \tag{7.8}$$

The results obtained are displayed in Table 2.

Table 2: Results obtained for System 2. Mean and standard deviations over 100 realizations.

Parameter	True value	A1 mean	A1 std	A2F mean	A2F std	A2B mean	A2B std
a_1	-1.5	-1.504	0.012	-1.504	0.026	-1.504	0.026
a_2	0.7	0.702	0.010	0.703	0.024	0.703	0.024
b_1	2.0	1.970	0.125	1.980	0.211	1.980	0.211
b_2	1.0	0.997	0.094	0.988	0.127	0.988	0.127
$\rho_y(1)$	7.84	6.51	7.01	207	34	2.44	212
$\rho_y(2)$	5.49	2.25	6.27	180	31	-1.29	188
$\rho_y(3)$	3.84	1.64	4,46	128	23	-0.57	134
λ_u	1	0.956	0.126	0.947	0.270	0.947	0.270

The results for System 2 show much of the same behaviour as the results for System 1. It is noticeable that the estimates of ρ_y are completely unreliable for algorithms **A2F** and **A2B**, although $\hat{\theta}$ is still very good. Algorithm **A1** produces the best results though.

We consider next four more, partly simpler, systems.

System 3. Now

$$\begin{aligned} y_o(t) &= \tfrac{2q^{-1}-q^{-2}}{1-0.8q^{-1}} u_o(t), \qquad u_o(t) = e(t), \quad \mathsf{E}e^2(t) = 10, \\ \mathsf{E}\tilde{u}^2(t) &= 1, \qquad \tilde{y}(t) = \tfrac{1}{1-0.7q^{-1}} e_y(t), \qquad \mathsf{E}e_y^2(t) = 4, \end{aligned} \tag{7.9}$$

leading to signal-to-noise ratios on the input and output sides, respectively,

$$\text{SNR}_u = 10 \text{ dB}, \qquad \text{SNR}_y = 1.55 \text{ dB}. \tag{7.10}$$

Table 3: Results obtained for System 3. Mean and standard deviations over 100 realizations.

Parameter	True value	A1 mean	A1 std	A2F mean	A2F std	A2B mean	A2B std
a_1	-0.8	-0.799	0.008	-0.799	0.010	-0.799	0.010
b_1	2.0	1.951	0.144	2.036	0.210	2.036	0.210
b_2	-1.0	-0.974	0.076	-1.016	0.109	-1.016	0.109
$\rho_y(1)$	7.84	5.23	3.55	4.18	4.13	3.11	5.25
$\rho_y(2)$	5.49	0.52	1.42	0.52	1.27	-0.33	2.13
λ_u	1	0.698	0.740	1.074	1.017	1.074	1.017

Table 4: Results obtained for System 4. Mean and standard deviations over 100 realizations.

Parameter	True value	A1 mean	A1 std	A2F mean	A2F std	A2B mean	A2B std
a_1	-1.5	-1.501	0.014	-1.501	0.032	-1.501	0.032
a_2	0.7	0.701	0.013	0.701	0.033	0.701	0.033
b_1	2.0	1.995	0.070	1.998	0.114	1.998	0.114
$\rho_y(1)$	7.84	4.04	1.26	155	9	28	85
$\rho_y(2)$	5.49	0.04	1.08	133	8	22	75
$\rho_y(3)$	3.84	0.00	0.68	94	7	15	54
λ_u	1	0.992	0.077	0.994	0.105	0.994	0.105

The results obtained are displayed in Table 3.

System 4. In this case the system is of second order, and

$$\begin{aligned} &y_o(t) = \tfrac{2q^{-1}}{1-1.5q^{-1}+0.7q^{-2}}u_o(t), \quad u_o(t) = \tfrac{1+0.7q^{-1}}{1-0.5q^{-1}}e(t), \quad \mathsf{E}e^2(t) = 1, \\ &\mathsf{E}\tilde{u}^2(t) = 1, \quad \tilde{y}(t) = \tfrac{1}{1-0.7q^{-1}}e_y(t), \quad \mathsf{E}e_y^2(t) = 4, \end{aligned} \tag{7.11}$$

leading to signal-to-noise ratios on the input and output sides, respectively,

$$\text{SNR}_u = 5.82 \text{ dB}, \quad \text{SNR}_y = 6.32 \text{ dB}. \tag{7.12}$$

The results obtained are displayed in Table 4.

System 5. Now

$$\begin{aligned} &y_o(t) = 2q^{-1}u_o(t), \quad u_o(t) = e(t), \quad \mathsf{E}e^2(t) = 10, \\ &\mathsf{E}\tilde{u}^2(t) = 1, \quad \tilde{y}(t) = \tfrac{1}{1-0.7q^{-1}}e_y(t), \quad \mathsf{E}e_y^2(t) = 5, \end{aligned} \tag{7.13}$$

leading to signal-to-noise ratios on the input and output sides, respectively,

$$\text{SNR}_u = 10 \text{ dB}, \quad \text{SNR}_y = 0.09 \text{ dB}. \tag{7.14}$$

Table 5: Results obtained for System 5. Mean and standard deviations over 100 realizations.

Parameter	True value	A1 mean	A1 std	A2F mean	A2F std	A2B mean	A2B std
b_1	2.0	1.98	0.19	2.03	0.21	2.03	0.21
$\rho_y(1)$	9.80	5.48	3.74	4.40	4.19	4.40	4.19
λ_u	1	0.79	0.92	1.04	1.02	1.04	1.02

Table 6: Results obtained for System 6. Mean and standard deviations over 100 realizations.

Parameter	True value	A1 mean	A1 std	A2F mean	A2F std	A2B mean	A2B std
b_1	2.0	1.93	0.14	1.95	0.16	1.95	0.16
b_2	1.0	0.97	0.07	0.97	0.08	0.97	0.08
$\rho_y(1)$	7.84	5.69	3.60	5.38	3.91	5.38	3.91
λ_u	1	0.59	0.73	0.65	0.79	0.65	0.79

The results obtained are displayed in Table 5.

System 6. Now

$$\begin{aligned} y_o(t) &= (2q^{-1} + q^{-2})u_o(t), \quad u_o(t) = e(t), \quad \mathsf{E}e^2(t) = 10, \\ \mathsf{E}\tilde{u}^2(t) &= 1, \quad \tilde{y}(t) = \frac{1}{1-0.7q^{-1}}e_y(t), \quad \mathsf{E}e_y^2(t) = 4, \end{aligned} \tag{7.15}$$

leading to signal-to-noise ratios on the input and output sides, respectively,

$$\mathrm{SNR}_u = 10 \text{ dB}, \qquad \mathrm{SNR}_y = 2.30 \text{ dB}. \tag{7.16}$$

The results obtained are displayed in Table 6.

The results reported in Tables 3-6 confirm the previous picture. All methods provide very accurate estimates of θ and Algorithm **A1** seems to give the smallest error. The estimates of λ_u are acceptable, while the estimates of ρ_y are fully unreliable. ■

8 Conclusions

Approaches for how to extend the Frisch method for identifying errors-in-variables systems have been described. The extension concerns the possibility to handle correlated output noise. A set of equations, partly nonlinear, that defines the estimates have been described. Some comparisons to the related bias-eliminating least squares (BELS) algorithm have also been provided. Two algorithms for solving the equations defining the parameter estimates have been derived. One algorithm is based on a multivariable optimization procedure, while the second consists of two one-dimensional optimization subproblems. The first algorithm has somewhat better (numerical) performance in general. The asymptotic statistical properties should though depend on the set of defining nonlinear equations, and not on the algorithm employed to solving these equations. As long as only the system parameters are concerned, the second algorithm turns out to be identical to the bias-eliminating least squares algorithm.

Acknowledgments

This work was supported in part by The Swedish Research Council under contract 621-2005-4207. I also thank Mei Hong for helpful comments on the manuscript.

References

[1] S. Beghelli, R.P. Guidorzi, and U. Soverini. The Frisch scheme in dynamic system identification. *Automatica*, 26:171–176, 1990.

[2] R. Diversi, R. Guidorzi, and U. Soverini. A new criterion in EIV identification and filtering applications. In *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, August 27-29 2003.

[3] R. Frisch. Statistical confluence analysis by means of complete regression systems. Technical Report 5, University of Oslo, Economics Institute, Oslo, Norway, 1934.

[4] W. A. Fuller. *Measurement Error Models*. Wiley, New York, NY, 1987.

[5] M. Hong, T. Söderström, and W. X. Zheng. A simplified form of the bias-eliminating least squares method for errors-in-variables identification. 2006. In preparation.

[6] L. Ljung. *System Identification - Theory for the User, 2nd edition*. Prentice Hall, Upper Saddle River, NJ, USA, 1999.

[7] T. Söderström. Identification of stochastic linear systems in presence of input noise. *Automatica*, 17:713–725, 1981.

[8] T. Söderström. Accuracy analysis of the Frisch estimates for identifying errors-in-variables systems. In *44th IEEE CDC/ECC 2005*, Seville, Spain, December 12-15 2005.

[9] T. Söderström. Errors-in-variables methods in system identification. In *14th IFAC Symposium on System Identification*, Newcastle, Australia, March 29-31 2006. Invited plenary paper.

[10] T. Söderström, M. Hong, and W. X. Zheng. Convergence properties of bias-eliminating algorithms for errors-in-variables identification. *International Journal of Adaptive Control and Signal Processing*, 19(9):703–722, November 2005.

[11] T. Söderström, U. Soverini, and K. Mahata. Perspectives on errors-in-variables estimation for dynamic systems. *Signal Processing*, 82(8):1139–1154, August 2002.

[12] T. Söderström and P. Stoica. *System Identification*. Prentice Hall International, Hemel Hempstead, UK, 1989.

[13] U. Soverini and T. Söderström. Identification methods of dynamic systems in presence of input noise. In *Proc. SYSID 2000, IFAC 12th Symposium on System Identification*, Santa Barbara, California, June 21–23 2000.

[14] P. Stoica and T. Söderström. A useful input parameterization for optimal experiment design. *IEEE Transactions on Automatic Control*, AC-27:736–738, 1982.

[15] S. Van Huffel, editor. *Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modelling.* SIAM, Philadelphia, USA, 1997.

[16] S. Van Huffel and Ph. Lemmerling, editors. *Total Least Squares and Errors-in-Variables Modelling. Analysis, Algorithms and Applications.* Kluwer, Dordrecht, The Netherlands, 2002.

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