Refined Instrumental Variable Estimation: Maximum Likelihood Optimization of a Unified Box-Jenkins Model

Peter C. Young

Systems and Control Group, Lancaster Environment Centre, Lancaster University, UK
Integrated Catchment Assessment and Management Centre, Australian National University College of Medicine, Biology & Environment, Canberra, ACT

Abstract

For many years, various methods for the identification and estimation of parameters in linear, discrete-time transfer functions have been available and implemented in widely available Toolboxes for Matlab^{TM}. This paper considers a unified *Refined Instrumental Variable* (RIV) approach to the estimation of discrete *and* continuous-time transfer functions characterized by a unified operator that can be interpreted in terms of backward shift, derivative or delta operators. The estimation is based on the formulation of a pseudo-linear regression relationship involving optimal prefilters that is derived from an appropriately unified Box-Jenkins transfer function model. The paper shows that, contrary to apparently widely held beliefs, the iterative RIV algorithm provides a reliable solution to the maximum likelihood optimization equations for this class of Box-Jenkins transfer function models and so its *en bloc* or recursive parameter estimates are optimal in maximum likelihood, prediction error minimization and instrumental variable terms.

Key words: System identification, Box-Jenkins model, maximum likelihood, optimal instrumental variable, en-bloc estimation, recursive estimation.

1 Introduction

Instrumental Variable (IV) methods of parameter estimation have a long history in the statistical and control engineering literature. IV estimation has its roots in statistics and econometrics [32] and is discussed in some detail by Kendall and Stuart [19]. Some early publications in the control engineering literature include [43,27] and [44]. Comprehensive treatments of ordinary and optimal IV methods applied to the estimation of parameters in discrete-time transfer function models then appeared almost simultaneously in two early books [35,46]. More recent papers in this general field include [12,8,42,21,48,22,40,34]. The present paper concerns the optimal *Refined Instrumental Variable* (RIV) approach to the unified estimation of parameters in both discrete and continuous-time transfer function models. The basic RIV algorithm was first suggested by the present author [45] for discrete-time models and then thoroughly evaluated and extended to multivariable and continuous-time models [52,16,53,46]. Over the subsequent years, it has been developed in various ways, with recent publications on this topic including [9,51,47,48,11,10]. The present paper follows the above references and considers estimation in the time domain. Alternative IV approaches formulated in the frequency domain (see e.g. [30]) have received much less attention, although recent research [13] is moving in this direction.

Unlike standard IV algorithms, the RIV approach is not based on an IV modification of a linear least-squares solution to the estimation problem, or an approximate approach to prediction error minimization. Rather, as this paper will show, it is an iterative *Pseudo-Linear Regression* (PLR) algorithm that is derived directly from the conditions required for optimization of the *Maximum Likelihood* (ML) function associated with a unified *Box-Jenkins* (BJ) transfer function model. Upon convergence of this iterative procedure, therefore, its parameter estimates are optimal in maximum likelihood, prediction error minimization and instrumental variable terms. This is a rather elegant solution because it not only provides *en-bloc* estimates that maximize the likelihood function but it can also produce *recursive* estimates that are identical to the repeated, stage-wise *en bloc* estimates, as in linear least squares estimation. Indeed, this was one of the primary motivations for developing the algorithm in

⋆ This paper was not presented at any IFAC meeting.

Email address: p.young@lancaster.ac.uk (Peter C. Young).

the iterative pseudo-linear form where, at each iteration, the estimates are obtained from a linear least squares solution that can be either *en-bloc* or recursive. Normally, however, the recursive estimates are only required by the user at the final iteration, where they relate to the converged, *en-bloc* parameter estimates. Here, the recursive estimates are useful for visually appraising the nature of the convergence and associated uncertainty at the final iteration, as illustrated later in the example of Section 6.1.

The first aim of the paper is to emphasize the inherent ML derivation of the RIV parameter estimation algorithm and so heighten awareness of this derivation. The second is to show how this solution is a unified one that can be applied to discrete-time *and* continuous-time transfer function models that includes models defined in terms of the backward shift, δ and derivative operators. The acronym RIV will be used to refer to the general, unified algorithm, while RIVD, RIVC and RIVδ will refer to the specific operator versions. For simplicity, the associated analysis will be presented for the case of a single input, single output, stochastic system. This is, of course, easily extended to a multiple input system where the transfer functions share a common denominator; and RIV algorithms for multiple input models with different denominators in each input channel have been developed for discrete [15] and continuous-time [9] models: these are described fully in these references, so they will not be considered here.

A third aim is to show that the iterative optimization procedure used in the standard implementation of the RIV algorithm can be considered in an entirely equivalent 'iterative updating' form of the *Gauss-Newton* (GN) kind, demonstrating again that it is based on an *implicit* prediction error minimization procedure. This GN interpretation demonstrates how the iterative optimization strategy is seeking out a local maximum of the log-likelihood cost function via prediction error minimization. As such, it provides an alternative to standard iterative prediction error minimization that is both robust under difficult estimation conditions and, as mentioned above, yields inherent recursive estimates of the model parameters because of the pseudo-linear nature of the estimation model. The paper will argue that these recursive parameter estimates can provide a useful diagnostic tool for evaluating both the identifiability of the model and the quality of the associated parameter estimates, as well as providing an obvious link with real-time recursive RIV estimation of time-variable parameters [24,48].

The next Section 2 of the paper introduces the unified BJ model; while the maximum likelihood estimation of the parameters in this model is considered in Section 3. Section 4 shows how maximum likelihood estimation of the unified BJ model can be accomplished by transforming the system and noise sub-models into pseudo-linear regression models, whose iterative estimation within the RIV framework yields maximum likelihood estimates of the full model parameters. Section 5 outlines the main aspects of the RIV algorithm and discusses its initiation, convergence and optimality in instrumental variable terms. Finally, Section 6 presents two simulation studies that reasonably exemplify the performance of the unified RIV algorithm when applied to backward shift, derivative and δ operator transfer function models.

2 The Unified Box-Jenkins Model

This paper is concerned with the estimation of the parameters that characterize a *Single-Input, Single-Output* (SISO), linear, time-invariant and stable transfer function model from uniformly sampled input-output data $\{u(k),\ y(k)\}$, $k = 1, 2, \ \ldots, N$, where the argument k denotes the kth sample from an underlying continuous-time system. In particular, let us consider the stochastic SISO transfer function model first conceived and promoted by Box and Jenkins [5,6] for discrete-time systems, which can be unified and written, at any sampling instant k, in the following decomposed form [1]:

$$\text{System TF Model}: \quad x(k) = \frac{B(\mu^{-1})}{A(\mu^{-1})} u(k-\tau) \tag{1a}$$

$$\text{Noise TF Model}: \quad \xi(k) = \frac{D(\mu^{-1})}{C(\mu^{-1})} e(k); \quad e(k) = \mathcal{N}(0, \sigma^2) \tag{1b}$$

$$\text{Output Observation}: \quad y(k) = x(k) + \xi(k) \tag{1c}$$

where τ is a pure time delay and μ is a unified operator that, in the present paper, can be interpreted as the forward shift operator, denoted here by z; the derivative operator, denoted here by s; or the delta operator, δ. The 'noise-free' output $x(k)$ plays an important part in the subsequent analysis and establishes the link between maximum likelihood and instrumental variable estimation. Given the possible interpretations of the unified operator μ, it is important to note that this model is informal and represents a 'snapshot' of the system at the kth sampling instant.

The model polynomials in μ that characterize the model (1) are defined as follows,

$$\begin{aligned} A(\mu^{-1}) &= 1 + a_1\mu^{-1} + a_2\mu^{-2} + \ \ldots \ + a_n\mu^{-n} \\ B(\mu^{-1}) &= b_0 + b_1\mu^{-1} + b_2\mu^{-2} + \ \ldots \ + b_m\mu^{-m} \\ C(\mu^{-1}) &= 1 + c_1\mu^{-1} + c_2\mu^{-2} + \ \ldots \ + c_p\mu^{-p} \\ D(\mu^{-1}) &= 1 + d_1\mu^{-1} + d_2\mu^{-2} + \ \ldots \ + d_q\mu^{-q} \end{aligned} \tag{2}$$

[1] Note that the nomenclature used for transfer functions here is that used for RIV estimation since 1976 [45,48]; in this unified context, models intended for PEM estimation in Matlab$^{\text{TM}}$ would use $C(\mu^{-1})/D(\mu^{-1})$ for the ARMA noise model; and/or $B(\mu^{-1})/F(\mu^{-1})$ for the system model.

Although these definitions are required for the development of the unified results and apply directly to the polynomials of the backward shift operator model, where $\mu^{-1} = z^{-1}$, the polynomials used in the subsequent development of RIVC and RIVδ algorithms, are defined in terms of μ (see later, Section 4.3.1), i.e.,

$$\begin{aligned} A(\mu) &= \mu^n + a_1\mu^{n-1} + a_2\mu^{n-2} + \ \ldots\ + a_n \\ B(\mu) &= b_0\mu^m + b_1\mu^{m-1} + b_2\mu^{m-2} + \ \ldots\ + b_m \end{aligned} \tag{3}$$

which does not, of course, change the model. Also, for reference in the next Section,

$$\mathbf{e}(k) = [e(1)\ e(2)\ \ldots,\ e(N)]^T; \quad \mathbf{e}(k) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{4}$$

where N is the number of samples available for estimation. The structure of the above model will be denoted by the pentad $[n\ m\ \tau\ p\ q]$ and, for simplicity of exposition, the time delay τ will be set initially to zero, without any loss of generality; and the μ^{-1} argument will be dropped from the polynomials.

Finally, note that, while this unified Box-Jenkins (BJ) model assumes the stochastic white noise source $e(k)$ is normally distributed, this is not an essential requirement for the *application* of the resultant RIV algorithms, although it is essential to the optimality of the ML approach used in the derivation of the RIV algorithm that follows below in the next two Sections.

3 Maximum Likelihood Estimation

Following the ML approach [3,24,30], as considered in [45,46], the log-likelihood function for the N observations $\{y(k),\ u(k)\},\ k = 1, 2,\ \ldots, N$, associated with the model (1), can be written as follows:

$$\begin{aligned} &\mathcal{L}(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \sigma^2, \mathbf{y}, \mathbf{u}) = -\frac{N}{2}\log_e(2\pi) - \\ &\frac{N}{2}\log_e\sigma^2 - \frac{1}{2\sigma^2}\left[\frac{C}{D}\mathbf{y} - \frac{BC}{AD}\mathbf{u}\right]^T\left[\frac{C}{D}\mathbf{y} - \frac{BC}{AD}\mathbf{u}\right] \end{aligned} \tag{5}$$

where,

$$\begin{aligned} &\mathbf{a} = [a_1\ a_2 \ldots,\ a_n]^T;\ \mathbf{b} = [b_0\ b_1 \ldots, b_m]^T \\ &\mathbf{c} = [c_1\ c_2 \ldots,\ c_p]^T;\ \ \mathbf{d} = [d_1\ d_2 \ldots, d_q]^T \\ &\mathbf{y} = [y(1)\ y(2) \ldots, y(N)]^T;\ \mathbf{u} = [u(1)\ u(2)\ \ldots,\ u(N)]^T \end{aligned}$$

Maximization of this log-likelihood function clearly requires the minimization of the final term on the right hand side of (5), which will be recognized as simply the sum of the squares of the prediction errors $e(k)$ where, considered from a control theoretic standpoint, these can be defined as follows:

$$e(k) = \frac{C}{D}\left[y(k) - \frac{B}{A}u(k)\right],\ k = 1, 2,\ \ldots, N. \tag{6}$$

or, alternatively,

$$e(k) = \frac{C}{DA}\left[Ay(k) - Bu(k)\right],\ k = 1, 2,\ \ldots, N. \tag{7}$$

which is important in the subsequent pseudo-linear regression analysis.

In the present context, the ML and PEM optimization problems are identical: it is necessary to find those estimates of the parameters in the polynomials (2), together with the variance σ^2, which minimize the third term in the log-likelihood function (5), namely, the sum of squares of the prediction errors, which is a classical but clearly nonlinear least squares problem. More formally this optimization requires:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}\ \mathcal{J}(\boldsymbol{\theta}\,,\,\mathbf{y}\,,\,\mathbf{u}) \tag{8}$$

where

$$\mathcal{J}(\boldsymbol{\theta}\,,\,\mathbf{y}\,,\,\mathbf{u}) = \sum_{k=1}^{N} e^2(k) \tag{9}$$

and

$$\boldsymbol{\theta} = [\mathbf{a}^T\ \mathbf{b}^T\ \mathbf{c}^T\ \mathbf{d}^T\ \sigma^2]^T \tag{10}$$

is the vector of unknown parameters in the model. The conditions for this are obtained in the usual manner by partially differentiating the log-likelihhood function with respect to all the parameters, in turn, and setting these derivatives to zero. This yields the following five equations:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial a_i} = \frac{1}{\sigma^2}\sum_{k=1}^{N}&\left[\frac{C}{D}y(k) - \frac{BC}{AD}u(k)\right]\times \\ &\frac{BC}{A^2D}\mu^{-i}u(k) = 0;\ \ i = 1, 2, ...,\ n \end{aligned} \tag{11a}$$

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial b_i} = \frac{1}{\sigma^2}\sum_{k=1}^{N}&\left[\frac{C}{D}y(k) - \frac{BC}{AD}u(k)\right]\times \\ &\frac{C}{AD}\mu^{-i}u(k) = 0;\ \ i = 0, 1, ...,\ m \end{aligned} \tag{11b}$$

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial c_i} = \frac{1}{\sigma^2}\sum_{k=1}^{N}&\left[\frac{C}{D}y(k) - \frac{BC}{AD}u(k)\right]\times \\ &\left[\frac{1}{D}\mu^{-i}y(k) - \frac{B}{AD}\mu^{-i}u(k)\right] = 0;\ i = 1, ...,\ p \end{aligned} \tag{11c}$$

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial d_i} = \frac{1}{\sigma^2}\sum_{k=1}^{N}&\left[\frac{C}{D}y(k) - \frac{BC}{AD}u(k)\right]\times \\ &\left[-\frac{C}{D^2}\mu^{-i}y(k) + \frac{BC}{AD^2}\mu^{-i}u(k)\right] = 0; \quad i = 1, ...,\ q \end{aligned} \tag{11d}$$

$$\frac{\partial\mathcal{L}}{\partial\sigma^2} = -\frac{N}{\sigma^2} + \frac{1}{\sigma^4}\sum_{k=1}^{N}\left[\frac{C}{D}y(k) - \frac{BC}{AD}u(k)\right]^2 = 0 \tag{11e}$$

Most published work on this topic has considered purely discrete-time models and the most common method of obtaining estimates of the BJ model parameters that satisfy these conditions is to search the parameter space in some manner. The best known and most computationally efficient approach is to use gradient-based schemes that follow from those originally employed by Box and Jenkins [5] for the BJ model, or Aström and Bohlin [4] for the alternative *AutoRegressive Moving Average eXogenous variables* (ARMAX) model.

A very interesting property of the ML estimates for the BJ model was revealed by the analysis of Pierce [29], who showed that the ML estimates of the parameters in the discrete-time model polynomials $A(z^{-1})$ and $B(z^{-1})$ are asymptotically independent of the estimates of the parameters in the ARMA noise model polynomials $C(z^{-1})$ and $D(z^{-1})$, so that the associated error covariance matrix for $\boldsymbol{\theta}$ is block diagonal, with the off-diagonal blocks zero. This is particularly important in the development of the pseudo-linear regression approach to estimation discussed in subsequent Sections because it is exploited to facilitate the iterative optimization approach used in the RIV algorithm.

4 The Iterative Pseudo-Linear Regression Approach to Maximum Likelihood Optimization

The RIV algorithm is a pseudo-linear regression approach to the maximum likelihood estimation of the unified BJ model, as defined in Sections 2 and 3, that involves separate but linked sub-algorithms for estimating the parameters in the system and noise models defined by equations (1a) and (1b), respectively. Before describing the RIV algorithm, however, it is necessary to consider the development of the PLR models that are exploited in these sub-algorithms.

4.1 The System Model

The pseudo-linear regression estimation procedure for the system model parameters becomes apparent if we consider equation (7) and define the following discrete-time 'prefiltered' variables

$$\begin{aligned} y_{f_\rho}(k) &= \frac{C}{DA}y(k); \quad u_{f_\rho}(k) = \frac{C}{DA}u(k) \\ x_{f_\rho}(k) &= \frac{B}{A}u_{f_\rho}(k) \end{aligned} \tag{12}$$

where the subscript f_ρ indicates that these are prefilters required for the estimation of the system parameter vector

$$\boldsymbol{\rho} = [a_1\ a_2\ \ldots,\ a_n\ b_0\ b_1\ \ldots,\ b_m]^T \tag{13}$$

Here, the subscript f_ρ is necessary because a different prefilter is required for estimating the noise model: see subsequent sub-Section 4.2.

With these definitions, note that equations (11a) and (11b) can be rewritten conveniently in terms of the prefiltered variables:

$$\sum_{k=1}^{N} [Ay_{f_\rho}(k) - Bu_{f_\rho}(k)]\mu^{-i}x_{f_\rho}(k) = 0 \tag{14a}$$

$$\sum_{k=1}^{N} [Ay_{f_\rho}(k) - Bu_{f_\rho}(k)]\mu^{-i}u_{f_\rho}(k) = 0 \tag{14b}$$

These equations are linear in the parameters $a_i,\ i = 1, 2,\ \ldots,\ n$ and $b_i,\ i = 0, 1,\ \ldots,\ m$, provided we assume knowledge of the prefiltered variables $y_{f_\rho}(k)$, $u_{f_\rho}(k)$ and $x_{f_\rho}(k)$ in (12).

Now, if we combine the three equations in (1) and manipulate the model to the form,

$$\frac{C}{DA}Ay(k) = \frac{C}{DA}Bu(k) + e(k) \tag{15}$$

then it can be written as follows in terms of the prefiltered variables,

$$Ay_{f_\rho}(k) = Bu_{f_\rho}(k) + e(k) \tag{16}$$

This can then be represented conveniently as the following regression-like model,

$$y_{f_\rho}(k) = \boldsymbol{\phi}_{f_\rho}^T(k)\boldsymbol{\rho} + e(k) \tag{17}$$

where,

$$\begin{aligned} \boldsymbol{\phi}_{f_\rho}^T(k) = [-\mu^{-1}y_{f_\rho}(k) \cdots - \mu^{-n}y_{f_\rho}(k) \\ u_{f_\rho}(k) \cdots\ \mu^{-m}u_{f_\rho}(k)] \end{aligned} \tag{18}$$

is a data vector defined in terms of the prefiltered variables. Now, equations (14) can be written in the alternative vector-matrix form:

$$\left[\sum_{k=1}^{N} \mathring{\boldsymbol{\phi}}_{f_\rho}(k)\boldsymbol{\phi}_{f_\rho}^T(k)\right]\boldsymbol{\rho} - \left[\sum_{k=1}^{N} \mathring{\boldsymbol{\phi}}_{f_\rho}(k)y_{f_\rho}(k)\right] = 0 \tag{19}$$

where

$$\begin{aligned} \mathring{\boldsymbol{\phi}}_{f_\rho}(k) = [-\mu^{-1}x_{f_\rho}(k) \cdots - \mu^{-n}x_{f_\rho}(k) \\ u_{f_\rho}(k) \cdots\ \mu^{-m}u_{f_\rho}(k)]^T \end{aligned} \tag{20}$$

is a data vector defined in the same way as $\boldsymbol{\phi}_{f_\rho}(k)$ but with the prefiltered output variables replaced by the similarly prefiltered *but unobserved*, noise-free output variables $x_{f_\rho}(k)$ defined in (12).

Of course, the regression-like model (17) cannot be considered as a basis for direct estimation because it is not a

true regression relationship; it is a pseudo-linear regression equation involving prefiltered variables *that depend on the parameters to be estimated*. Before considering how this problem is circumvented, however, the development of a similarly motivated PLR equation for the noise model is considered in the next sub-Section.

4.2 The Noise Model

In all three operational transfer functions considered in this paper, the ARMA noise model is considered in purely discrete-time terms because of the well known problems of directly estimating the parameters in a purely continuous-time ARMA process (see e.g. [20]).

There are various methods available for the estimation of parameters in such a discrete-time ARMA model (see e.g. [31], page 359 *et seq*) but, in order to achieve uniformity in RIV estimation, we require an approach that is based on pseudo-linear regression and so can yield recursive estimates if these are required. The pseudo-linear regression estimation method utilized in the RIV algorithm is the *Instrumental Variable ARMA* (IVARMA) algorithm [48], which is actually applied to the *inverse* noise model,

$$e(k) = \frac{C}{D}\xi(k), \tag{21}$$

under the initial assumption that both $e(k)$ and $\xi(k)$ are available for measurement. Then the following prefiltered variables are introduced:

$$e_{f_\eta}(k) \;=\; \frac{1}{D}e(k); \quad \xi_{f_\eta}(k) \;=\; \frac{1}{D}\xi(k) \tag{22}$$

where the subscript f_η indicates that these filters are required for the estimation of the noise model parameter vector $\boldsymbol{\eta}$ defined later in equation (26). Now, after some manipulation of the ML equations [46], (11c) and (11d) can be rewritten conveniently in terms of the prefiltered variables:

$$\sum_{k=1}^{N} \left[C\xi(k) \;-\; D^*e(k)\right] \mu^{-i}\xi_{f_\eta} \;=\; 0 \tag{23a}$$

$$\sum_{k=1}^{N} \left[C\xi(k) \;-\; D^*e(k)\right] \mu^{-i}e_{f_\eta} \;=\; 0 \tag{23b}$$

for $i = 1, 2, \ldots, p$, where here μ^{-i} is the backward shift operator z^{-i}

$$D^* \;=\; D^*(z^{-1}) \;=\; d_1z^{-1} \;+\; d_2z^{-2} \;+\; \ldots \;+ d_qz^{-q}$$

These equations are linear in the c_i and d_i parameters, again *provided we assume knowledge of the prefiltered variables in* (22).

Noting that the pseudo-linear relationship between the noise model parameters can be obtained as follows [48]:

$$\begin{aligned}\frac{C(z^{-1})}{D(z^{-1})}\xi(k) - e(k) = \\ C(z^{-1})\xi_{f_\eta}(k) - D(z^{-1})e_{f_\eta}(k)\end{aligned} \tag{24}$$

The underlying estimation model in this case is then:

$$e_{f_\eta}(k) = \mathring{\boldsymbol{\psi}}^T_{f_\eta}(k)\boldsymbol{\eta} \tag{25}$$

in which,

$$\begin{aligned}\mathring{\boldsymbol{\psi}}^T_{f_\eta}(k) = [\xi_{f_\eta}(k) \;\cdots\; \xi_{f_\eta}(k-p) - \\ e_{f_\eta}(k-1)\cdots - e_{f_\eta}(k-q)] \\ \boldsymbol{\eta} = [1\; c_1 \cdots\; c_p\; d_1 \cdots d_q]^T\end{aligned} \tag{26}$$

Now, if indeed $\xi(k)$ and $e(k)$ were available for measurement, the equations (23a) can be written in the alternative vector-matrix form:

$$\left[\sum_{k=1}^{N} \mathring{\boldsymbol{\psi}}_{f_\eta}(k)\mathring{\boldsymbol{\psi}}^T_{f_\eta}(k)\right]\boldsymbol{\eta} - \left[\sum_{k=1}^{N} \mathring{\boldsymbol{\psi}}_{f_\eta}(k)e_{f_\eta}(k)\right] = 0 \tag{27}$$

Of course, $\xi(k)$ and $e(k)$ are not directly available for measurement and so, as in the case of the system model, they must be replaced by estimated variables in the iterative pseudo-linear regression solution.

4.3 Refined Instrumental Variable Estimation

The RIV estimation algorithm is an iterative method of jointly solving the optimization equations for the system and noise model parameters derived in the previous Sections. Referring to the ideal normal equations in (19), upon convergence of the iterations, the final RIV estimate $\hat{\boldsymbol{\rho}}$ is given by,

$$\hat{\boldsymbol{\rho}} = \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_{f_\rho}(k)\boldsymbol{\phi}^T_{f_\rho}(k)\right]^{-1} \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_{f_\rho}(k)y_{f_\rho}(k)\right] \tag{28}$$

where $\boldsymbol{\phi}_{f_\rho}(k)$ is the data vector in equation (18) and $\hat{\boldsymbol{\phi}}_{f_\rho}(k)$ is an iteratively updated estimate of the 'noise-free' vector $\mathring{\boldsymbol{\phi}}_{f_\rho}$ in equation (20), in which $\hat{x}_{f_\rho}(k)$ is an iteratively updated estimate of $x_{f_\rho}(k)$. In instrumental variable terms, $\hat{\boldsymbol{\phi}}_{f_\rho}(k)$ is the IV vector,

$$\begin{aligned}\hat{\boldsymbol{\phi}}_{f_\rho}(k) = [-\mu^{-1}\hat{x}_{f_\rho}(k)\cdots - \mu^{-n}\hat{x}_{f_\rho}(k) \\ u_{f_\rho}(k)\cdots\; \mu^{-m}u_{f_\rho}(k)]^T\end{aligned} \tag{29}$$

where $\hat{x}_{f_\rho}(k)$ is generated as the output of the *auxiliary model* equation in the RIV algorithm, i.e.,

$$\hat{x}_{f_\rho}(k) \;=\; \frac{\hat{B}}{\hat{A}} u_{f_\rho}(k) \tag{30}$$

in which $\hat{A}$ and $\hat{B}$ are the iteratively updated estimates of the A and B polynomials.

Similarly, referring to the ideal normal equations (27) of the noise model, the final converged estimate of the ARMA noise model parameter vector $\hat{\boldsymbol{\eta}}$ is given by,

$$\hat{\boldsymbol{\eta}} = \left[\sum_{k=1}^{N} \hat{\boldsymbol{\psi}}_{f_\eta}(k)\boldsymbol{\psi}_{f_\eta}^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \hat{\boldsymbol{\psi}}_{f_\eta}(k)\tilde{e}_{f_\eta}(k)\right] \tag{31}$$

where $\boldsymbol{\psi}_{f_\eta}(k)$ is the data vector in equation (26) and $\hat{\boldsymbol{\psi}}_{f_\eta}(k)$ is the IV vector,

$$\begin{aligned}\hat{\boldsymbol{\psi}}_{f_\eta}(k) = [\hat{\xi}_{f_\eta}(k) \;\cdots\; \hat{\xi}_{f_\eta}(k-p) - \\ \hat{e}_{f_\eta}(k-1)\cdots - \hat{e}_{f_\eta}(k-q)]^T\end{aligned} \tag{32}$$

in which

$$\hat{\xi}_{f_\eta}(k) = y_{f_\rho}(k) - \hat{x}_{f_\rho}(k) \tag{33}$$

and $\hat{e}_{f_\eta}(k)$ is generated as the output of the 'inverse noise' auxiliary model equation

$$\hat{e}_{f_\eta}(k) \;=\; \frac{\hat{C}}{\hat{D}} \hat{\xi}_{f_\eta}(k) \tag{34}$$

where $\hat{C}$ and $\hat{D}$ are the iteratively updated estimates of the C and D polynomials. Finally, $\tilde{e}_{f_\eta}(k)$ in (31) is generated from an independent estimate of $e(k)$ obtained as the residual of high order AR estimation, as discussed below.

In more specific terms, the iterative RIV algorithm blends the separate algorithms for the estimation of the system and noise parameters using a 'bootstrap' approach. Here, at the j^{th} iteration, the system model parameter estimate $\hat{\boldsymbol{\rho}}^{j-1}$ obtained at the previous iteration by the solution of the RIV normal equations (28), or their recursive equivalent, provides the information required to form the auxiliary model. This is then used both to generate the instrumental variable $\hat{x}(k)$ from $u(k)$ (*cf* the third equation in (22)) and to provide an estimate $\hat{\xi}(k)$ of the noise $\xi(k)$, i.e.,

$$\begin{aligned}&\text{Estimation of } x(k): \quad \hat{x}(k) = \frac{\hat{B}(\hat{\boldsymbol{\rho}}^{j-1})}{\hat{A}(\hat{\boldsymbol{\rho}}^{j-1})} u(k) \\ &\text{Estimation of } \xi(k): \qquad \hat{\xi}(k) = y(k) - \hat{x}(k);\end{aligned} \tag{35}$$

Obtaining an estimate of $e(k)$ is more difficult but it is well known that a high order AR model of $\xi(k)$, as estimated by *en bloc* or recursive linear least squares, yields residuals that provide a good estimate $\tilde{e}(k)$ of $e(k)$ (see e.g. [7,14]). The estimates $\hat{\xi}(k)$ and $\tilde{e}(k)$ obtained in this manner are then prefiltered by $f_\eta = 1/\hat{D}$, where $\hat{D}$ is the iteratively updated estimate of MA polynomial D, and these prefiltered variables are used to construct the following vector,

$$\begin{aligned}\tilde{\boldsymbol{\psi}}_{f_\eta}(k) = [\hat{\xi}_{f_\eta}(k) \;\cdots\; \hat{\xi}_{f_\eta}(k-p) - \\ \tilde{e}_{f_\eta}(k-1)\cdots - \tilde{e}_{f_\eta}(k-q)]^T\end{aligned} \tag{36}$$

which provides an iteratively updated estimate of the unobservable vector $\mathring{\boldsymbol{\psi}}(k)$ in equation (26). The estimate of the ARMA noise model parameter vector is then given by reference to equation (31): i.e.,

$$\hat{\boldsymbol{\eta}} = \left[\sum_{k=1}^{N} \tilde{\boldsymbol{\psi}}_{f_\eta}(k)\boldsymbol{\psi}_{f_\eta}^T(k)\right]^{-1} \left[\sum_{k=1}^{N} \tilde{\boldsymbol{\psi}}_{f_\eta}(k)\tilde{e}_{f_\eta}(k)\right] \tag{37}$$

with $\tilde{\boldsymbol{\psi}}_{f_\eta}(k)$ replacing $\hat{\boldsymbol{\psi}}_{f_\eta}(k)$. This is the basis of the IVARMA algorithm.

4.3.1 Special aspects of the RIVC and RIVδ estimation

As pointed out previously in Section 2, in the implementation of the hybrid continuous-time (RIVC) and delta operator (RIVδ) versions of the above RIV algorithm, the A and B polynomials are made functions of μ, rather than μ^{-1}. This is because, in these cases, the pseudo-linear regression estimation equation (17) is considered in the alternative form,

$$\mu^n y_{f_\rho}(k) = \boldsymbol{\phi}_{f_\rho}^T(k)\boldsymbol{\rho} + e(k), \tag{38}$$

as obtained by multiplying through the equation by μ^n, where now,

$$\begin{aligned}\boldsymbol{\phi}_{f_\rho}^T(k) = [-\mu^{n-1}y_{f_\rho}(k)\cdots - y_{f_\rho}(k) \\ \mu^m u_{f_\rho}(k)\cdots \; u_{f_\rho}(k)]\end{aligned} \tag{39}$$

In other words, the 'dependent' variable in the pseudo-linear regression is the filtered nth order derivative of the measured output, in continuous or discrete-time. In the same way, the instrumental variable vector in (29) is defined as

$$\begin{aligned}\hat{\boldsymbol{\phi}}_{f_\rho}(k) = [-\mu^{n-1}\hat{x}_{f_\rho}(k)\cdots - \hat{x}_{f_\rho}(k) \\ \mu^m u_{f_\rho}(k)\cdots \; u_{f_\rho}(k)]^T\end{aligned} \tag{40}$$

5 The Unified RIV Algorithm

The major steps in the current standard implementation of the RIV algorithms are summarized below. For completeness, the time delay τ and appropriate arguments

are re-introduced here. The RIV algorithm is a unified one in the sense that its basic operations are defined in terms of the unified operator μ^{-1}. However, there are small differences, depending on the specific operator model being considered. These specific algorithms will be referred to by the acronyms RIVD, RIVδ and RIVC for, respectively, backward shift, δ and derivative operator transfer function models. Also, it is necessary to initiate the algorithm and this initiation also depends to some extent on the specific operator model concerned. In order to not obscure the nature of the basic unified algorithm, therefore, these are discussed in the next sub-Section 5.1. A diagram of the unified RIV algorithm is given in Figure 1, where the estimated parameter vector $\hat{\boldsymbol{\theta}}$ (see Section 3), with σ^2 removed because it is estimated separately (see below), is shown in partitioned form as the combination of the system and noise model parameter vectors, i.e.

$$\hat{\boldsymbol{\theta}} = \left[\hat{\mathbf{a}}^T \ \hat{\mathbf{b}}^T \ \middle| \ \hat{\mathbf{c}}^T \ \hat{\mathbf{d}}^T \right]^T = \left[\hat{\boldsymbol{\rho}}^T \ \middle| \ \hat{\boldsymbol{\eta}}^T \right]^T \qquad (41)$$

Major Steps in the Unified RIV Algorithm

Step 1. Initialization: This provides an initial estimate of the TF system model parameter vector $\hat{\boldsymbol{\rho}}^0$ (see sub-Section 5.1).

Step 2. Iterative or recursive-iterative IV estimation with adaptive prefilters:

for $j = 1$: convergence

(1) Generate the IV series $\hat{x}(k)$ from the system 'auxiliary model':

$$\hat{x}(k) = \frac{\hat{B}(\mu^{-1}, \hat{\boldsymbol{\rho}}^{j-1})}{\hat{A}(\mu^{-1}, \hat{\boldsymbol{\rho}}^{j-1})} u(k-\tau)$$

with the polynomials based on the estimated parameter vector $\hat{\boldsymbol{\rho}}^{j-1}$ obtained at the previous iteration of the algorithm; for $j = 1$, $\hat{\boldsymbol{\rho}}^0$ is the estimate obtained in Step 1.

(2) Obtain the latest estimate $\hat{\boldsymbol{\eta}}^j$ of the noise model parameter vector based on the estimated noise sequence $\hat{\xi}(k) = y(k) - \hat{x}(k)$ using the *en bloc* IVARMA estimation algorithm (37) or the recursive equivalent of this.

(3) Prefilter the input $u(k)$, output $y(k)$ and instrumental variable $\hat{x}(k)$ signals by the filter

$$f_\rho(z^{-1}, \hat{\boldsymbol{\rho}}^{j-1}, \hat{\boldsymbol{\eta}}^j) = \frac{\hat{C}(z^{-1}, \hat{\boldsymbol{\eta}}^j)}{\hat{D}(z^{-1}, \hat{\boldsymbol{\eta}}^j)\hat{A}(\mu^{-1}, \hat{\boldsymbol{\rho}}^{j-1})}$$

with the polynomials based on the estimated parameter vector $\hat{\boldsymbol{\rho}}^{j-1}$ obtained at the previous iteration of the algorithm and $\hat{\boldsymbol{\eta}}^j$ obtained in Step (2); for $j = 1$, $\hat{\boldsymbol{\rho}}^0$ is the estimate obtained in Step 1.

(4) Based on these prefiltered data, compute the estimate $\hat{\boldsymbol{\rho}}^j$ of the TF system model parameter vector using the *en bloc* solution (28) or the recursive IV equivalent of this.

end

Step 3. Error covariance matrix evaluation: After convergence, compute the estimated parametric error covariance matrices associated with the parameter estimates, from the following expressions that follow from the theorem by Pierce [29,18].

$$\begin{aligned} \mathbf{P}_\rho^* &= \hat{\sigma}^2 \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_{f_\rho}(k) \hat{\boldsymbol{\phi}}_{f_\rho}^T(k) \right]^{-1} \\ \mathbf{P}_\eta^* &= \hat{\sigma}^2 \left[\sum_{k=1}^{N} \tilde{\boldsymbol{\psi}}_{f_\eta}(k) \tilde{\boldsymbol{\psi}}_{f_\eta}^T(k) \right]^{-1} \end{aligned} \qquad (42)$$

where $\hat{\sigma}^2$ is estimated by reference to condition (11e) of equation (11) as

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{k=1}^{N} \hat{e}^2(k) \qquad (43)$$

and $\hat{e}(k)$ are the estimated ARMA model residuals. $\mathbf{P}_\rho^*$ is also generated naturally from the 'symmetric' versions of these algorithms, so that they can be accessed straightforwardly by adding such a symmetric update subsequent to convergence (see page 183 in [46] and later Section 5.2). Note that the definitions (42) assume that the off-diagonal blocks in the full error covariance matrix for $\boldsymbol{\theta}$ are zero, following the Pierce theorem (see Section 3).

The *Simplified Refined Instrumental Variable* (SRIV) algorithm is a very useful, reliable and computationally efficient version of the RIV-type algorithm obtained in the case where the additive noise is purely white (i.e. $C = D = 1.0$), so removing completely the need to estimate the ARMA noise model. The resulting SRIVD, SRIVC and SRIVδ algorithms are optimal under these special conditions but they also function well in practice,

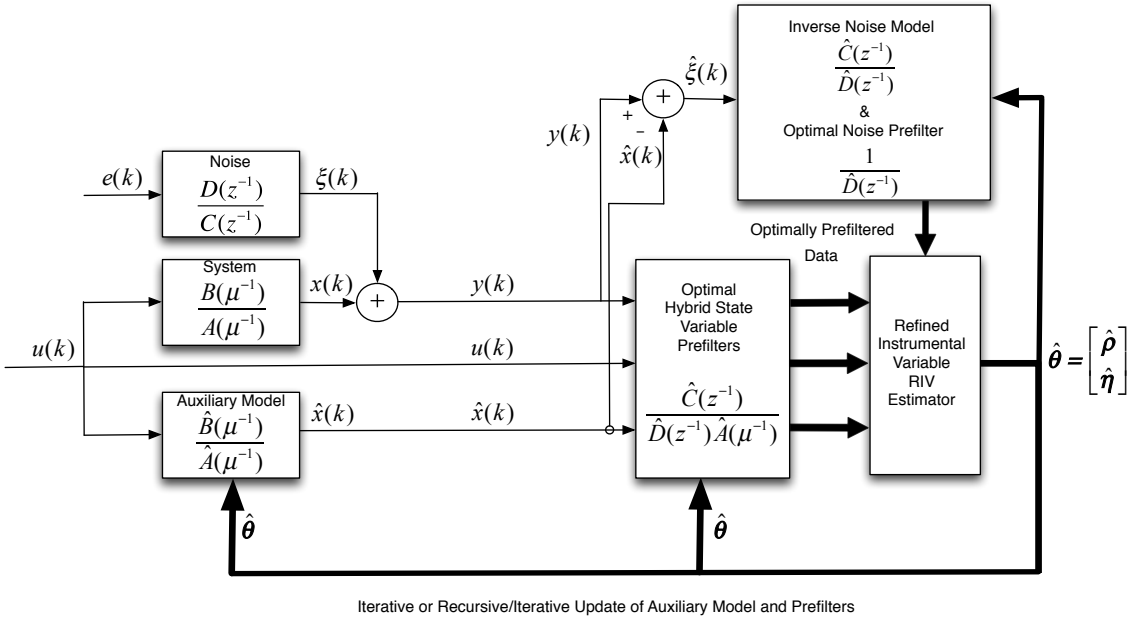

Fig. 1. Block diagram of the unified RIV estimation algorithm.

maintaining consistency even if the noise is coloured because of the IV-type solution; or even if the noise does not conform to the ARMA type description. In effect, the SRIV algorithms are the pseudo-linear regression equivalent of standard 'Output Error' (OE) estimation but they can sometimes produce superior estimation results (e.g. [9,25]). Note also that if the denominator of the ARMA noise model is constrained equal to the system transfer function denominator, the RIV algorithm can be used for the estimation of ARMAX models (see [18] which also discusses RIV estimation of other transfer function model forms).

Finally, the recursive equivalent of the *en bloc* solution (28) is well known [48] and will not be discussed here. It is sufficient to note that the unified recursive equations take the form:

$$\begin{aligned} \hat{\boldsymbol{\rho}}(k) &= \hat{\boldsymbol{\rho}}(k-1) + \mathbf{g}(k)[y_{f\rho}(k) - \boldsymbol{\phi}_{f_\rho}^T(k)\hat{\boldsymbol{\rho}}(k-1)\,] \\ \mathbf{g}(k) &= \hat{\mathbf{P}}_\rho(k-1)\hat{\boldsymbol{\phi}}_{f_\rho}(k)\times \\ &\qquad [1 + \boldsymbol{\phi}_{f_\rho}^T(k)\hat{\mathbf{P}}_\rho(k-1)\hat{\boldsymbol{\phi}}_{f_\rho}(k)]^{-1} \\ \hat{\mathbf{P}}_\rho(k) &= \hat{\mathbf{P}}_\rho(k-1) - \mathbf{g}(k)\boldsymbol{\phi}_{f_\rho}^T(k)\hat{\mathbf{P}}_\rho(k-1) \end{aligned} \tag{44}$$

where the third equation is a recursive update of the inverse IV cross-product matrix

$$\hat{\mathbf{P}}_\rho(k) = \left[\sum_{i=1}^{k} \hat{\boldsymbol{\phi}}_{f_\rho}(i)\boldsymbol{\phi}_{f_\rho}^T(i)\right]^{-1} \tag{45}$$

that avoids matrix inversion. Recursive estimation takes a little longer than *en-bloc* estimation so, although it could be used at each iteration, it is computationally more efficient to use it only at the final iteration, as mentioned previously in Section 1 and illustrated later in the example of Section 6.1.

5.1 Initiation

The initiation of the RIV/SRIV algorithms requires the specification of $A(z^{-1}, \hat{\boldsymbol{\rho}}^0)$, $A(s, \hat{\boldsymbol{\rho}}^0)$ or $A(\delta, \hat{\boldsymbol{\rho}}^0)$. In the case of the RIVD/SRIVD algorithm, it is standard practice to use initial linear least squares estimation of an ARX model. While the parameters of this model are normally biased, they are nearly always sufficient to define an initial estimate that induces subsequent convergence of the RIV algorithm. Note that if the estimated ARX model is unstable, then the unstable eigenvalues of the estimated denominator polynomial are reflected into the stable region of the complex z^{-1} plane (e.g. using the Matlab routine polystab). Normally, this estimate is employed to initiate the simpler SRIVD algorithm (see above) and then the resulting estimated parameter vector is used to define the initial $A(z^{-1}, \hat{\boldsymbol{\rho}}^0)$ for final, full RIVD estimation. Of course, a facility could be added that allows the user to specify $A(z^{-1}, \hat{\boldsymbol{\rho}}^0)$ on the basis of prior knowledge; or to employ the 'state variable filter' approach that is used in the RIVC algorithm.

The initiation of the RIVC algorithm is discussed fully in

[51,48] but the two simplest approaches are: (i) estimate an ARX model in discrete-time, as in the RIVD case, and convert this to continuous-time form in the conventional manner, so providing an initial $A(s, \hat{\boldsymbol{\rho}}_c^0)$; or (ii) select a single breakpoint parameter λ (breakpoint frequency in radians/time unit) of the *State Variable Filter* (SVF):

$$f(s, \boldsymbol{\rho}_c^0) = \frac{1}{A(s, \boldsymbol{\rho}_c^0)} = \frac{1}{(s+\lambda)^n} \tag{46}$$

This latter SVF approach, which is normally superior, was suggested a long while ago [44] but has proven popular ever since. When used in practice, it is necessary to find a λ value that induces satisfactory convergence. This is chosen so that it is equal to, or larger than, the bandwidth of the system to be identified. However, the choice is not critical since convergence occurs over a fairly wide range of values (see example 2, Section 6.2). These same two approaches can also be used for initiation of the RIVδ algorithm but, naturally, in relation to the specification of the initial $A(\delta, \hat{\boldsymbol{\rho}}^0)$ polynomial.

5.2 Convergence

A recent, quite comprehensive analysis of convergence for RIVC algorithm is given in [23], who refer to the earlier analysis of Stoica and Söderström [39,36] in relation to the Steiglitz-McBride algorithm [38]. Here, we will consider the local convergence of the unified RIV algorithm in rather simpler terms using an alternative 'incremental' implementation of the algorithm *that is completely equivalent to the standard RIV iterative update.*

The standard RIV algorithm outlined in Section 5 is one way of implementing the pseudo-linear regression type of optimization based on the ideas presented in Sections 4.1 to 4.3. But there are other possibilities provided that the resulting RIV algorithm is based on the PLR approach with the basic prefiltering carried out as described. One possibility is to consider how the parameter vector is updated at the jth iteration based on its estimate at the previous $(j-1)$th iteration. In order to develop this incremental iterative updating, equation (28) is written in the following vector-matrix terms at iteration j:

$$\hat{\boldsymbol{\rho}}_N^j = \left[\hat{\boldsymbol{\Phi}}_{j-1}^T \boldsymbol{\Phi}_{j-1}\right]^{-1} \hat{\boldsymbol{\Phi}}_{j-1}^T \mathbf{y}_{j-1}, \tag{47}$$

Here, $\hat{\boldsymbol{\rho}}_N^j$ is the estimate based on all N samples in the data set at the jth iteration; $\boldsymbol{\Phi}_{j-1}$ is a $N \times m+n+1$ data matrix with rows defined in (18) or (39), depending of the specific form of the algorithm, with the prefilters defined at iteration $j-1$; $\hat{\boldsymbol{\Phi}}_{j-1}$ is a similarly formed and prefiltered instrumental variable product matrix, with rows defined in (29) or (40); and $\mathbf{y}_{j-1}$ is the $N \times 1$ prefiltered output vector with elements $y_{f_{\rho,j-1}}(k)$ or $\mu^n y_{f_{\rho,j-1}}(k)$, $k = 1, 2, \ldots, N$, where the subscript $f_{\rho,j-1}$ denotes the prefiltering operation at iteration $j-1$. However, $\mathbf{y}_{j-1}$ can be defined in terms of the previous iteration estimates, i.e.,

$$\mathbf{y}_{j-1} = \boldsymbol{\Phi}_{j-1} \hat{\boldsymbol{\rho}}_N^{j-1} + \mathbf{e}_{j-1} \tag{48}$$

where $\mathbf{e}_{j-1}$ is the $N \times 1$ vector of recursive residuals $\hat{e}(k)$, $k = 1, 2, \ldots, N$. Combining these equations shows that

$$\hat{\boldsymbol{\rho}}_N^j = \hat{\boldsymbol{\rho}}_N^{j-1} + \left[\hat{\boldsymbol{\Phi}}_{j-1}^T \boldsymbol{\Phi}_{j-1}\right]^{-1} \hat{\boldsymbol{\Phi}}_{j-1}^T \mathbf{e}_{j-1} \tag{49}$$

When considering these updating equations, it is important to emphasize that the updating algorithm (49) is *entirely equivalent* to the standard RIV iterative update (47) and so it can be used to evaluate the convergence of the standard RIV iterations. Also, this equation is of the *Gauss-Newton* (GN) type (see e.g. [24]), except that the cross-product matrix in the square brackets is not symmetric because of the IV formulation (cf. the equivalent symmetric Hessian matrix within a standard GN context). Nevertheless, it will be very close to symmetric because the source of the instrumental variables (the output of the auxiliary model) is an estimate of the noise-free output $x(k)$ and it can be shown [48] that,

$$\text{p.lim } \frac{1}{N} \left[\hat{\boldsymbol{\Phi}}^T \boldsymbol{\Phi}\right] = \frac{1}{N} \left[\hat{\boldsymbol{\Phi}}^T \hat{\boldsymbol{\Phi}}\right] \tag{50}$$

where p.lim is the probability in the limit as $N \to \infty$. Moreover, in RIV estimation $\hat{x}(k) \to x(k)$, so that

$$\text{p.lim } \frac{1}{N} \left[\hat{\boldsymbol{\Phi}}^T \boldsymbol{\Phi}\right] = \frac{1}{N} \left[\mathring{\boldsymbol{\Phi}}^T \mathring{\boldsymbol{\Phi}}\right]$$

where $\mathring{\boldsymbol{\Phi}}(k)$ is a $N \times n+m+1$ matrix with rows $\mathring{\boldsymbol{\phi}}^T(i)$, $i = 1, 2, \ldots, N$, as defined previously in terms of the noise-free output and input variables by equation (20). Consequently, we see that, in these probabilistic terms, the symmetry is implicit in the RIV algorithm. The GN nature of the RIV algorithm becomes even more obvious if a 'symmetric gain' version of the RIV algorithm (see above, Step 3 in Section 5) is utilized, i.e.,

$$\hat{\boldsymbol{\rho}}_N^j = \hat{\boldsymbol{\rho}}_N^{j-1} + \left[\hat{\boldsymbol{\Phi}}_{j-1}^T \hat{\boldsymbol{\Phi}}_{j-1}\right]^{-1} \hat{\boldsymbol{\Phi}}_{j-1}^T \mathbf{e}_{j-1} \tag{51}$$

where the Hessian is now symmetric. It is then clear from the ML formulation of the algorithm and the Pierce theorem [29] that the update is indeed the product of the Hessian and gradient of the ML cost function. Indeed, this is the reason why the standard implementation of the RIV algorithms includes a final iteration using this symmetric gain version of the solution. Originally, the symmetric gain form of the RIV algorithm was recommended [52], however, it is not now used until the final iteration because it has been found, by practical application of the RIV algorithm over many years [18,48], that

it is more robust to use the standard asymmetric RIV solution (28) (i.e. the *en-bloc* equivalent of (49)).

If the GN update (51) was being applied to a pure regression rather than a pseudo-linear regression model, then it would constitute a Newton algorithm and so converge in one step. In the present context, however, it constitutes an efficient gradient descent algorithm and, depending on the provision of a suitable initial estimate, as discussed above in sub-Section 5.1, it will converge to a local minimum or stationary point and can be compared to the 'damped' GN algorithm normally used for PEM optimization [24], except that there is no 'step-size' parameter. Indeed, as Van den Hof and Douma point out in relation to the SRIVC algorithm [41], unlike this classic GN algorithm, updates such as (49) and (51) do not rely on an approximate Taylor series expansion for their formulation and may, therefore, have improved convergence properties. In other words, at each iteration of the RIV algorithm the pseudo-linear regression model is automatically linearized by the introduction of the prefilters and, therefore, provides a very suitable basis for GN-type updating. This is certainly borne out by practical experience with RIV algorithms over many years, as well as simulation results (e.g. [47]) which show that RIV can have advantages over PEM, particularly when applied to data from 'stiff' systems that have a wide range of eigenvalues. Of course, the PEM approach is much more general than RIV estimation and has advantages in other areas, such as the estimation of multiple input models with a different denominator in each constituent transfer function, where the RIV 'backfitting' approach [15,9] is less efficient.

5.3 RIV optimality in Instrumental Variable Terms

Söderström and Stoica [37] develop a general framework for instrumental variable estimation that they term *Extended Instrumental Variable* (EIV) estimation. This includes the use of prefilters that depend on the form of the transfer function model being considered and, in the case of the BJ model considered here, these prefilters are the same as those considered in the previous Sections of this paper (see also [12,11]). In particular, using the nomenclature of the present paper, this EIV formulation considers IV estimation in terms of the following optimal solution:

$$\hat{\boldsymbol{\rho}} = \arg\min_{\boldsymbol{\rho}} \|\mathbf{E}(k)\|^2 \tag{52}$$

where,

$$\mathbf{E}(k) = \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_{f_\rho}(k)\boldsymbol{\phi}_{f_\rho}^T(k)\right]\boldsymbol{\rho} - \sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_{f_\rho}(k)y_{f_\rho}(k) \tag{53}$$

and optimization of this criterion with respect to $\boldsymbol{\rho}$ then yields the RIV solution (28), i.e.,

$$\hat{\boldsymbol{\rho}} = \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_{f_\rho}(k)\boldsymbol{\phi}_{f_\rho}^T(k)\right]^{-1}\left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_{f_\rho}(k)y_{f_\rho}(k)\right],$$

Moreover, Söderström [34], in considering "How accurate can instrumental variables become", concludes that, upon convergence of the iterations, the prefilters used in this RIV solution are optimal from EIV and PEM standpoints. In other words, not only are the RIV estimates optimal in maximum likelihood and prediction error terms, but also the RIV algorithm is an optimal instrumental variable estimator.

6 Simulation Examples

Recently, a number of publications by the author and his colleagues have provided simulation and real examples of RIV-type estimation and have compared the results with those obtained by other methods (see e.g. [47,48]). Here, therefore, we will consider only two simulation examples that reasonably exemplify the performance of the unified RIV algorithm. In these examples, generic names will be used for the various algorithms but routines from the CAPTAIN Toolbox [2] for Matlab™ will be used for RIV model identification and parameter estimation. These are the rivbjid and rivbj routines that implement RIVD model structure identification and parameter estimation, respectively; and the analogous rivcbjid and rivcbj routines in the case of RIVC. A similar implementation of RIVC is available in the CONTSID Toolbox [3]. For comparison, the pem and tfest routines from the Matlab SID Toolbox are used for PEM estimation of discrete and continuous-time models, respectively. The results for δ operator model estimation are obtained using a recently updated version of an algorithm developed some years ago [50] but not available in the CAPTAIN Toolbox.

Although model structure identification will not be discussed in these examples, such identification has been used in all cases using the rivbjid and rivcbjid routines in CAPTAIN and based on well known order identification criteria such as AIC [1], BIC [2,33] and YIC (see [48] and the prior references therein), as well as reference to the recursive estimation results (see e.g. section 7.3 of chapter 7 in [48]). However, the degree to which a model explains the measured output data will be specified by

[2] The CAPTAIN Toolbox can be downloaded free via `http://captaintoolbox.co.uk/Captain_Toolbox.html` or `http://www.es.lancs.ac.uk/cres/systems.html`

[3] The CONTSID Toolbox can be downloaded free from `www.cran.uhp-nancy.fr/contsid/`

the following coefficient of determination:

$$R_T^2(x) = 1 - \{\sigma_{e_x}^2/\sigma_x^2\} \tag{54}$$

where σ_x^2 is the sample variance of the noise-free output $x(k)$; and $\sigma_{e_x}^2$ is the sample variance of the error $e_x(k) = x(k) - \hat{x}(k)$ between the noise-free output and the simulated noise-free output of the model. In other words, $R_T^2(x)$ measures how much of the noise-free output variance is explained by the model.

6.1 Simulation Example 1: Discrete-Time Example

This example is concerned with the following, fourth order, backward shift operator TF model [55]:

$$\begin{aligned} y(k) &= \frac{z^{-1} + 0.5z^{-2} - 2.0z^{-3} + z^{-4}}{1 - 1.5z^{-1} + 0.7z^{-2} + 0.3z^{-3} - 0.2z^{-4}}u(k) \\ &\quad + \frac{1 - 0.6z^{-1} + 0.4z^{-2}}{1 - 0.95z^{-1} + 0.9506z^{-2}}e(k) \\ e(k) &= \mathcal{N}(0, \sigma^2) \end{aligned}$$

where σ^2 is selected so that the noise/signal ratio, in relation to the noise-free output, is 0.2 by variance or 0.447 by standard deviation (equivalent to a signal/noise ratio (SNR) of 8.7 dB, using the standard SNR definition). The system and noise models are highly oscillatory with natural frequencies that are quite close together at 0.676 and 1.060 radians per time unit; and the associated damping factors are 0.102 and 0.238. The input signal is a 1000 sample, pseudo-random binary signal between 0 and 1 with a switching period of 10 samples and an adequate 'run-in' of 20 zero samples to avoid the well known problems that can occur because of the sharp step changes in the input. This is quite a difficult example with considerable data distortion entering via the reasonable level of additive, highly coloured and oscillatory noise.

The model is identified with the correct [4 4 1 2 2] structure and the RIVD estimation is evaluated by *Monte Carlo Simulation* (MCS) experiments involving 200 random realizations. For each realization, the RIVD algorithm is initiated using standard ARX model estimation (see Section 5.1). The results are given in Table 1, which compares the performance of the RIVD algorithm with the PEM algorithm. The results for the noise model are omitted because of space limitations in the table. Also shown are the results obtained when the GN-type updating (see Section 5.2), referred to here as 'trimming', is applied subsequent to convergence of the standard RIVD algorithm. It is clear that these two RIVD results are almost identical, as expected. The PEM results are also similar, although there were nine failures where the estimated parameters were far from the true values. These were removed in calculating the statistics for Table 1.

The estimates obtained in a typical single realization are as follows, where the figures in parentheses are the estimated standard errors, which we see agree reasonably with the standard deviations obtained in the MCS analysis.

$$\begin{array}{ll} \hat{a}_1 = -1.498(0.037) & \hat{a}_2 = 0.697(0.067) \\ \hat{a}_3 = 0.298(0.065) & \hat{a}_4 = -0.194(0.033) \\ \hat{b}_0 = 1.047(0.032) & \hat{b}_1 = 0.442(0.079) \\ \hat{b}_2 = -2.006(0.084) & \hat{b}_3 = 1.022(0.036) \\ \hat{c}_1 = -0.942(0.012) & \hat{c}_2 = 0.948(0.012) \\ \hat{d}_1 = -0.587(0.033) & \hat{d}_2 = 0.367(0.033) \end{array} \tag{55}$$

The associated $R_T^2(x) = 0.9998$; i.e. 99.98% of the noise-free output variance is explained by the model. The estimates obtained for the worst realization in these terms are:

$$\begin{array}{ll} \hat{a}_1 = -1.490(0.043) & \hat{a}_2 = 0.694(0.070) \\ \hat{a}_3 = 0.295(0.068) & \hat{a}_4 = -0.192(0.038) \\ \hat{b}_0 = 0.981(0.035) & \hat{b}_1 = 0.544(0.083) \\ \hat{b}_2 = -1.982(0.088) & \hat{b}_3 = 0.967(0.039) \\ \hat{c}_1 = -0.955(0.012) & \hat{c}_2 = -0.941(0.012) \\ \hat{d}_1 = -0.524(0.034) & \hat{d}_2 = 0.357(0.034) \end{array} \tag{56}$$

where $R_T^2(x) = 0.995$. Of course, this is at the extremity of the long tailed $R_T^2(x)$ distribution, which has a median value of $R_T^2(x) = 0.9990$. Even in this case, however, the associated step and frequency responses are very similar to the true ones.

As pointed out previously, one advantage of the RIV approach to estimation is that it is straightforward to generate the recursive estimates, if these are required. Figure 2 shows the recursive estimates for each of the parameters $\hat{a}_1(k)$ and $\hat{b}_2(k)$, as obtained from the final iteration (see Section 5) of four random realizations (represented, respectively, by full, dashed, dash-dot and dotted black lines), using a longer data set of 3000 samples. The grey regions in Figure 2 indicate the changing estimated 95% confidence bounds for the estimates shown by the black line and reveal that the estimate of a_1 is much better defined than b_2. The recursive estimates provide the user with an excellent visualization of the uncertainty in the parameters, considerable enhancing the standard error estimates that are returned at the end of the data set after *en bloc* estimation. Taking the uncertainty bounds in Figure 2 into account, at least 1000 samples are required to obtain reasonable estimates of the b_2 parameter. In other words, although the model is identifiable, the recursive estimates show that estimation, particularly of the transfer function numerator parameters, is made difficult because of the highly oscillatory noise effects; and they suggest that, if the numerical values of the parameters are important for the application being

Parameter		$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{b}_3$	Failures
True Values		-1.5	0.7	0.3	-0.2	1.0	0.5	-2	1	
RIVD	mean	-1.504	0.704	0.299	-0.201	1.002	0.488	-1.992	0.999	0
	SD	0.034	0.062	0.063	0.035	0.032	0.079	0.084	0.035	
PEM	Mean	-1.503	0.704	0.298	-0.200	1.001	0.489	-1.992	0.998	9
	SD	0.034	0.062	0.064	0.035	0.032	0.080	0.084	0.035	
RIVD with GN 'trimming'	mean	-1.504	0.704	0.299	-0.201	1.002	0.488	-1.992	0.999	0
	SD	0.034	0.062	0.064	0.035	0.032	0.079	0.084	0.035	

Table 1

MCS estimation results obtained by the RIVD, PEM and RIVD with additional GN 'trimming' (based on 200 realizations; 'mean' and 'SD' denote the mean values and standard deviation of these MCS realizations)

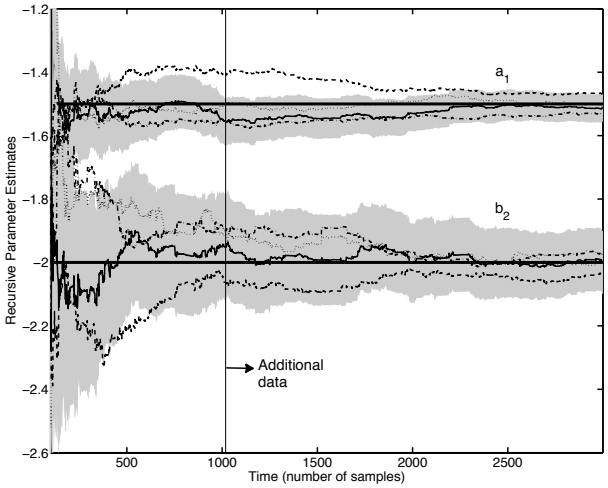

Fig. 2. Four realizations of the RIVD recursive estimates of the a_1 and b_2 parameters, shown as full, dashed, dash-dot and dotted black lines, with the grey regions showing the 95% confidence bounds for the full line estimates.

considered, then it would be desirable to base single realization estimation (often the practical situation) on at least 2000 samples in order to ensure well converged estimates. In all of the cases shown in Figure 2, however, $R^2_T(x) > 0.999$ after 1000 samples, so the models would be appropriate for applications such as forecasting. Note that recursive estimates are even more useful in cases where the model is poorly identifiable, for example because of over-parameterization: here, the recursive estimates do not show any clear convergence, whatever the sample size, but tend to wander about because of the ill-defined minimum in the cost function hypersurface. Example 7.4 of [48] shows typical results in this regard.

Finally, although this is a discrete-time model, it is interesting to note that the RIVC algorithm is able to produce a reasonable explanation of the data: e.g. in a typical single run, the best identified model has a structure pentad [4 4 0 2 2] and the time delay is accommodated by a small non-minimum phase effect. This yields a model with a coefficient of determination, based on the noise free output, of $R^2_T(x) = 0.994$.

6.2 Simulation Example 2: Continuous-Time Model

This is a modified version of a difficult benchmark example prepared for the IFAC SYSID'06 Symposium in Newcastle, NSW, Australia, in which the simple white additive noise of the original is replaced by ARMA noise, making it still more difficult. The example is the following fourth order system with widely separated modal frequencies and an ARMA(2,1) noise model:

$$\begin{aligned} x(k) &= \frac{-120s^2 - 1560s + 3600}{s^4 + 30.2s^3 + 3607s^2 + 750s + 3600}u(k) \\ &\quad + \frac{1 + 0.5z^{-1}}{1 - 1.4z^{-1} + 0.7z^{-2}}e(k); \\ e(k) &= \mathcal{N}(0, 0.0025) \end{aligned} \tag{57}$$

The input signal is a maximal length pseudo-random binary sequence (±1.0); and the complete data set consists of 6138 input-output samples with a sampling interval of $\Delta t = 0.005$ time units (i.e. total time 30.69 time units). The noise level at the output is high: in relation to the noise-free output, the noise-signal-ratio is 0.64 by variance or 0.8 by standard deviation (equivalent to a SNR=1.94 dB).

The [4 3 0 2 1] model structure is correctly identified and RIVC estimation is evaluated by MCS analysis in a similar manner to the previous discrete-time example. In all realizations, the RIVC algorithm was initiated with the SVF breakpoint frequency $\lambda = 0.1$, although the same results were obtained for λ between 0.00001 and 0.5. The results are shown in Table 2. Also shown are the results obtained using the tfest routine, recently introduced into the Matlab^{TM} SID Toolbox, which is initiated with a SID implementation of SRIVC. Here, the mean estimates are very similar to the RIVC algorithm estimates but the standard deviations are marginally larger and there was one failure (the estimates are far removed from their true

Parameter		$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{c}_1$	$\hat{c}_2$	$\hat{d}_1$	Failures
True Values		30.2	3607	750	3600	-120	-1560	3600	-1.4	0.70	0.50	
RIVC	mean	30.2	3646	763	3641	-121	-1591	3646	-1.399	0.70	0.50	0
	SD	6.55	281	68.6	282	18.5	145	288	0.010	0.010	0.013	
TFEST	Mean	30.3	3674	770	3668	-122.2	-1609	3671	-	-	-	1
	SD	6.95	299	72.8	300	19.6	153	309				

Table 2

MCS estimation results obtained by the RIVC and TFEST algorithms for the model (57) (based on 100 realizations; 'mean' and 'SD' denote the mean values and standard deviation of these MCS realizations)

values). Noise model estimates are not given by tfest [4] .

The RIVC estimated model from the closest realization to the median of the $R_T^2(x)$ distribution from the MCS analysis has the following parameter estimates, with the estimated standard errors shown in parentheses:

$$\begin{aligned} \hat{a}_1 &= 28.46(6.04) & \hat{a}_2 &= 3501(237) \\ \hat{a}_3 &= 724(57.7) & \hat{a}_4 &= 3525(241) \\ \hat{b}_0 &= -116(17.8) & \hat{b}_1 &= -1618(138) \\ \hat{b}_2 &= 3364(241) & \hat{c}_1 &= -1.401(0.010) \\ \hat{c}_2 &= 0.704(0.010) & \hat{d}_1 &= 0.512(0.012) \\ \hat{\sigma}^2 &= 0.0025 \end{aligned} \tag{58}$$

where $R_T^2(x) = 0.999$.

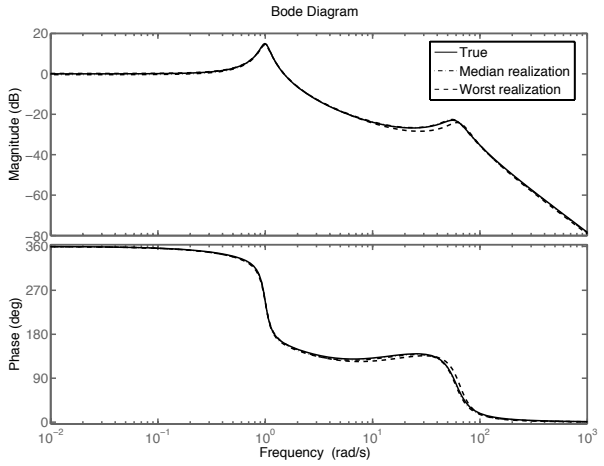

Fig. 3. Comparison of true and RIVC estimated Bode diagrams

The estimated standard errors are of the same order as the standard deviations in Table 2. Figure 3 shows that the Bode plots of the RIVC estimated model (dash-dot) are hardly distinguishable from those of the true model (full line). The initial SRIVC estimated model produced very similar Bode plots. The estimates obtained for the worst realization in terms of $R_T^2(x)$ are:

$$\begin{aligned} \hat{a}_1 &= 29.1(7.58) & \hat{a}_2 &= 4246(332) \\ \hat{a}_3 &= 974(86.3) & \hat{a}_4 &= 4238(336) \\ \hat{b}_0 &= -109.1(20.3) & \hat{b}_1 &= -1969(188) \\ \hat{b}_2 &= 4328(353) & \hat{c}_1 &= -1.393(0.010) \\ \hat{c}_2 &= 0.693(0.010) & \hat{d}_1 &= 0.518(0.012); \\ \hat{\sigma}^2 &= 0.0025 \end{aligned} \tag{59}$$

where $R_T^2(x) = 0.996$. Once again, the same estimates were obtained for all initiations with λ ranging between 0.00001 and 0.5. The Bode plots in this case are shown as dashed lines in Figure 3.

If the Matlab c2d routine is applied to the continuous-time model (57) it yields complex roots in the denominator polynomial at $0.9995 \pm 0.00497j$. These roots are very close to the unit circle because the sampling rate is very fast in this example and this leads to both the RIVD and PEM algorithms failing to converge to satisfactory discrete-time models. In control system terms, it would be advantageous at this fast sampling rate to use the associated δ operator model [28,50] obtained using either conversion from the continuous-time to the delta operator model, or direct estimation of this model using RIVδ. The latter yields the model:

$$y(k) = \frac{-115.5\delta^2 - 1546\delta + 3526}{\delta^4 + 47.6\delta^3 + 3608\delta^2 + 792\delta + 3611} u(k) + \xi(k) \tag{60}$$

which explains the noise-free output well with $R_T^2 = 0.998$, marginally lower than the continuous-time model (58).

7 Conclusions

This paper has presented the *Refined Instrumental Variable* (RIV) approach to the estimation of a unified transfer function model of the Box-Jenkins type that subsumes transfer functions in the backward shift, δ

[4] tfest is associated with the idtf routine, whose instructions state that 'unlike idss and idpoly, idtf uses a trivial noise model and does not parameterize the noise'.

and derivative operators as special cases. It has shown how this RIV algorithm is formulated and developed within the context of maximum likelihood optimization and that the optimization strategy is based on iterative solution of pseudo linear regression equations. It has also shown that this strategy can be considered in an entirely equivalent *Gauss-Newton* (GN) updating form that overtly minimizes the sum-of-squares of the stochastic model prediction errors. We see, therefore, that in maximum likelihood estimation of Box-Jenkins transfer function models, the RIV-type algorithm is an effective alternative to prediction error minimization using gradient-based numerical optimization. As pointed out in Section 5, it can be used also, in a simply constrained form, for the identification and estimation of ARMAX and other transfer function model forms [17,18]. A RIV algorithm for full, multivariable TF model estimation has been developed and evaluated [16] but it is complex to implement and use, so the amalgamation of separately estimated multiple input models is recommended, even though this is not strictly optimal in statistical terms.

The GN formulation of the RIV optimization in equation (49) has its attractions, particularly from a theoretical standpoint where it demonstrates that the algorithm will converge to a local minimum of the ML cost function, requiring only an easily specified initial estimate of the transfer function denominator polynomial. In more practical terms, however, its full implementation (see e.g [26]) removes the algorithm's ability to easily generate recursive estimates and can have other disadvantages [49]. GN updates could be used to 'trim' the RIV estimates subsequent to convergence of the standard RIV iterations, but simulation and practical examples suggest that this rarely makes any noticeable difference to the estimates, although it may be worthwhile introducing it as a user option for the standard RIV algorithms.

The option to generate recursive estimates of the model parameters is an important advantage of standard RIV algorithmic family. Indeed, as far as the author is aware, this RIV approach is the only, fully unified method of transfer function model identification and estimation that exploits a virtually identical algorithm for both discrete and continuous-time transfer function models while, at the same time, allowing for inherent recursive estimation in which the recursive estimates are identical to those obtained by stage-wise *en-bloc* estimation. It is also distinguished by its optimality in both maximum likelihood and instrumental variable terms, with the instrumental nature of the algorithmic implementation providing robustness to violation of the statistical assumptions associated with the noise model, as well as an assurance of consistent parameter estimates even if the additive noise does not have rational spectral density.

The paper has not described various additional RIV algorithms that have been developed and are available in the CAPTAIN and CONTSID Toolboxes. These include multiple input models with either a common denominator in all constituent transfer functions or different denominators [15,9], where the former is a trivial extension of the single input case described in the present paper; models within a feedback system [47,11]; *Linear Parameter Varying* (LPV) models [21]; *State Dependent Parameter* (SDP) nonlinear models [54]; and frequency-domain formulations [13]. Except for the latter approach, these are described in [48], and the prior references therein. This latter reference also discusses real-time recursive estimation of time variable and state-dependent parameters which are, of course, most appropriate extensions to constant parameter RIV estimation because of its inherent recursive estimation facility.

8 Acknowledgements

I am grateful to my good friend Hugues Garnier for suggesting that an article such as this would be useful to clarify the nature of refined instrumental variable estimation to a wider audience and for helping to check both the original and revised versions. Also thanks to the anonymous referees, Hugues and Marion Gilson-Bagrel for useful suggestions that have improved the quality of this paper. Of course, I remain responsible for any errors or omissions.

References

[1] H. Akaike. A new look at statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[2] H. Akaike. A Bayesian extension of the minimum AIC procedure of autoregressive model fitting. *Biometrika*, 66:237–242, 1979.

[3] K. J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16:551–574, 1980.

[4] K. J. Åström and T. Bohlin. Numerical identification of linear dynamic systems from normal operating records. In P. H. Hammond, editor, *Theory of Self Adaptive Control Systems*, pages 96–111, New York, 1966. Plenum Press.

[5] G. E. P. Box and G. M. Jenkins. *Time Series Analysis Forecasting and Control*. Holden-Day: San Francisco, 1970.

[6] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 3rd edition, 1994.

[7] J. Durbin. Estimation of parameters in time-series regression models. *Jnl. Roy. Stat. Soc. Series B*, 22:139–153, 1960.

[8] S. Garatti, M.C. Campi, and S. Bittanti. The asymptotic model quality assessment for instrumental variable identification revisited. *Systems & Control Letters*, 45:494–500, 2006.

[9] H. Garnier, M. Gilson, P. C. Young, and E. Huselstein. An optimal IV technique for identifying continuous-time transfer function model of multiple input systems. *Control Engineering Practice*, 15:471–486, 2007.

[10] H. Garnier and P. C. Young. The advantages of directly identifying continuous-time transfer function models in practical applications. *Int. J. Control*, 87(7):1319–1338, 2014.

[11] M. Gilson, H. Garnier, P.C. Young, and P. Van den Hof. Optimal instrumental variable methods for closed-loop identification. *IET Control Theory & Applications*, 5(10):1147–1154, July 2011.

[12] M. Gilson and P. Van den Hof. Instrumental variable methods for closed-loop system identification. *Automatica*, 41(2):241–249, February 2005.

[13] M. Gilson, J. Welsh, and H. Garnier. Frequency-domain instrumental variable based method for wide band system identification. In *American Control Conference (ACC'2013)*, Washington, DC, USA, 2013.

[14] E. J. Hannan and J. Rissanen. Recursive estimation of mixed autoregressive moving average order. *Biometrika*, 69:81–94, 1982.

[15] A. J. Jakeman, L. P. Steele, and P. C. Young. Instrumental variable algorithms for multiple input systems described by multiple transfer functions. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-10:593–602, 1980.

[16] A. J. Jakeman and P. C. Young. Refined instrumental variable methods of time-series analysis: Part II, multivariable systems. *International Journal of Control*, 29:621–644, 1979.

[17] A. J. Jakeman and P. C. Young. On the decoupling of system and noise model parameter estimation in time-series analysis. *International Journal of Control*, 34:423–431, 1981.

[18] A. J. Jakeman and P. C. Young. Advanced methods of recursive time-series analysis. *International Journal of Control*, 37:1291–1310, 1983.

[19] M.G. Kendall and A. Stuart. *Advanced Theory of Statistics*. Griffin, London, 1961.

[20] E. K. Larsson, M. Mossberg, and T. Soderstrom. An overview of important practical aspects of continuous-time ARMA system identification. *Circuits, Systems and Signal Processing*, 25(1):17–46, 2006.

[21] V. Laurain, R. Toth, M. Gilson, and H. Garnier. Refined instrumental variable methods for identification of LPV Box–Jenkins models. *Automatica*, 46(6):959–967, 2010.

[22] V. Laurain, R. Toth, M. Gilson, and H. Garnier. Direct identification of continuous-time linear parameter-varying input/output models. *IET Control Theory & Applications*, 5(7):878–888, May 2011.

[23] X. Liu, J. Wang, and W. X. Zheng. Convergence analysis of refined instrumental variable method for continuous-time system identification. *IET Control Theory & Applications*, 5(7):868–877, 2011.

[24] L. Ljung. *System identification. Theory for the user*. Prentice Hall, Upper Saddle River, N.J., 2nd edition, 1999.

[25] L. Ljung. Experiments with identification of continuous-time models. In *15th IFAC Symposium on System Identification (SYSID'2009)*, Saint-Malo (France), July 2009.

[26] I. Maruta and T. Sugie. Projection-based identification algorithm for grey-box continuous-time models. *Systems & Control Letters*, 62(11):1090 – 1097, 2013.

[27] D.Q. Mayne. A method for estimating discrete-time transfer functions. In *Advances in computer control, Second UKAC control convention*, Bristol (UK), 1967.

[28] R. H. Middleton and G. C. Goodwin. *Digital Control and Estimation - A Unified Approach*. Prentice Hall, 1990.

[29] D. A. Pierce. Least squares estimation in dynamic disturbance time-series models. *Biometrika*, 5:73–78, 1972.

[30] R. Pintelon and J. Schoukens. *System identification: a frequency domain appraoch*. IEEE Press, 2001.

[31] M. B. Priestley. *Spectral Analysis and Time Series*. Academic Press, London, 1981.

[32] O. Reiersol. Confluence analysis by means of lag moments and other methods of confluence analysis. *Econometrica*, 9:1–24, 1941.

[33] G. Schwarz. Estimating the dimension of a model. *Annal. Statist.*, 6:461–464, 1978.

[34] T. Söderström. How accurate can instrumental variable models become? In *System Identification, Environmental Modelling, and Control System Design*, volume L. Wang and H. Garnier (Eds.), pages 3–26, Springer-Verlag, 2012. London.

[35] T. Söderström and P. Stoica. *Instrumental variable methods for system identification*. Springer Verlag, New York, 1983.

[36] T. Söderström and P. Stoica. On some system identification techniques for adaptive filtering. *IEEE Trans on Circuits and Systems*, 35:457–461, 1988.

[37] T. Söderström and P. Stoica. *System Identification*. Series in Systems and Control Engineering. Prentice Hall, New York, 1989.

[38] K. Steiglitz and L. E. McBride. A technique for the identification of linear systems. *IEEE Transactions on Automatic Control*, 10:461–464, October 1965.

[39] P. Stoica and T. Söderström. The Steiglitz-McBride identification algorithms revisited. Convergence analysis and accuracy aspects. *IEEE Transactions on Automatic Control*, AC-26:712–717, 1981.

[40] R. Toth, V. Laurain, M. Gilson, and H. Garnier. Instrumental variable scheme for closed-loop LPV model identification. *Automatica*, 48(9):2314–2320, 2012.

[41] P. Van den Hof and S.G. Douma. An IV-based iterative linear regression algorithm with optimal output error properties. Technical Report 09-018, Delft Center for Systems and Control, Delft University of Technology, 2008.

[42] J. Wang, W.X. Zheng, and T. Chen. Identification of linear dynamic systems operating in a networked environment. *Automatica*, 45(12):2763–2772, 2009.

[43] K.Y. Wong, E. Polak, and T. Chen. Identification of linear discrete time systems using the instrumental variable approach. *IEEE Transactions on Automatic Control*, 12:707–718, 1967.

[44] P. C. Young. An instrumental variable method for real-time identification of a noisy process. *Automatica*, 6:271–287, 1970.

[45] P. C. Young. Some observations on instrumental variable methods of time-series analysis. *International Journal of Control*, 23:593–612, 1976.

[46] P. C. Young. *Recursive Estimation and Time-Series Analysis: An Introduction*. Springer-Verlag, Berlin, 1984.

[47] P. C. Young. The refined instrumental variable method: unified estimation of discrete and continuous-time transfer function models. *Journal Européen des Systèmes Automatisés*, 42:149–179, 2008.

[48] P. C. Young. *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. Springer-Verlag, Berlin. (revised and much enlarged version of 1984 book), 2011.

[49] P. C. Young. Comment on 'projection-based identification algorithm for grey-box continuous-time models'. *Systems and Control Letters*, 69:62–64, 2014.

[50] P. C. Young, A. Chotai, and W. Tych. Identification, estimation and control of continuous-time systems described by delta operator models. In N.K. Sinha and G.P. Rao, editors, *Identification of Continuous-Time Systems*, pages 363–418. Kluwer: Dordrecht, 1991.

[51] P. C. Young, H. Garnier, and M. Gilson. Refined instrumental variable identification of continuous-time hybrid Box-Jenkins models. In H. Garnier and L. Wang, editors, *Identification of Continuous-Time Models from Sampled Data*, pages 91–131. Springer-Verlag: London, 2008.

[52] P. C. Young and A. J. Jakeman. Refined instrumental variable methods of time-series analysis: Part I, SISO systems. *International Journal of Control*, 29:1–30, 1979.

[53] P. C. Young and A. J. Jakeman. Refined instrumental variable methods of time-series analysis: Part III, extensions. *International Journal of Control*, 31:741–764, 1980.

[54] P. C. Young, P. McKenna, and J. Bruun. Identification of nonlinear stochastic systems by state dependent parameter estimation. *International Journal of Control*, 74:1837–1857, 2001.

[55] Y. Zhu. A Box-Jenkins method that is asymptotically globally convergent for open loop data. In *Proceedings of the IFAC World Congress*, pages 9047–9051, Milano (Italy), 2011.