The Refined Instrumental Variable Method: Unified Estimation of Discrete and Continuous-Time Transfer Function Models

Peter C. YOUNG

Centre for Research on Environmental Systems and Statistics, Lancaster University, Lancaster, LA1 5BL, U.K.
and Centre for Resource and Environmental Studies, Australian National University, Canberra

p.young@lancaster.ac.uk
http://www.es.lancs.ac.uk/cres/

Résumé— *This paper describes the unified Refined Instrumental Variable approach to the time domain identification and estimation of both discrete-time (RIV) and continuous-time (RIVC) transfer function models. It demonstrates how this approach yields parameter estimates with optimal statistical properties for the Box-Jenkins and hybrid Box-Jenkins model forms on which the associated RIV and RIVC estimation algorithms are based. The performance of the algorithms, which can be implemented in en-bloc or recursive form, is evaluated by Monte Carlo simulation analysis and their practical utility is illustrated by a number of practical environmental examples.*

Mots-clés— *Identification, estimation, discrete-time, continuous-time, transfer function, optimal instrumental variable.*

I. INTRODUCTION

The *Instrumental Variable* (IV) approach to the identification and estimation[1] of transfer function models has a rich history in the control and systems literature, with the earliest algorithms of this type dating back to the 1960s [22] [21] [10] [18] [23] [26] [24]. This paper outlines the main aspects of the statistically optimal *Refined Instrumental Variable* approach to the identification and estimation of both discrete-time (RIV) and continuous-time (RIVC) *Transfer Function* (TF) models [27][43][5][44][28]. It also demonstrates how this approach yields parameter estimates with optimal statistical properties for both the Box-Jenkins [2] and hybrid Box-Jenkins [41] model forms on which, respectively, the associated RIV and RIVC estimation algorithms are based.

As far as the author is aware, the RIV/RIVC algorithms together constitute the only unified, time domain, family of algorithms that provide statistically optimal solutions to the estimation of both discrete-time and 'hybrid' continuous-time TF models of the Box-Jenkins type. In this regard, they have advantages over alternative algorithms, such as the well known *Prediction Error Minimization* (PEM) approach [8] used in the Matlab System Identification (SID) Toolbox, where the general time domain algorithms for *direct* TF model estimation are only available for discrete-time models. Following from their exploitation of IV methodology, the RIV and RIVC algorithms are robust to violation of the assumptions about the statistical properties of the noise. In particular, the discrete-time RIV algorithm has an inherent advantage, in this respect, over the alternative PEM approach, which requires these assumptions to be satisfied (except when used in its 'output error' form, where the noise model is not specified). Moreover, the iterative (relaxation) IV algorithm, which forms the basis of both the RIV and RIVC algorithms, seems less sensitive than the PEM algorithm to the specification of the initial conditions required for the implementation of the algorithms (e.g., in the RIV case, initial parameter estimates obtained for the equivalent ARX model using linear least squares estimation). Moreover, they appear to handle 'stiff' dynamic systems, with widely spaced eigenvalues, more reliably than PEM. The instrumental variable formulation also means that the RIV and RIVC algorithms are able to exploit the properties of the *Instrumental Product Matrix* (IPM) for model order identification [17] [45] (see Appendix 1, which outlines the identification statistics used in the simulation and practical examples considered later in this paper).

The paper also points out how the discrete-time Box-Jenkins model form exploited by the RIV algorithm has certain advantages over the ARMAX alternative, although the RIV algorithm can be used to estimate the ARMAX model, if this is required. In particular, a theorem is presented that shows how the *Maximum Likelihood* (ML) estimates of the TF system model parameters are asymptotically independent of the *Auto-Regressive Moving Average* (ARMA) noise model parameters. The paper shows how this property proves very useful for both the theoretical justification and the practical implementation of the RIV and RIVC algorithms.

Finally, the practical utility of this RIV/RIVC approach to data-based modelling is demonstrated by the simulation and real examples presented in the paper, as well as its successful practical application to many different real data examples over the past 30 years. The paper also discusses briefly a nonlinear extension of linear transfer function models, based on a *State-Dependent Parameter (SDP)* form, and demonstrates its efficacy by means of a practical example concerned with the analysis and modelling of climate data.

II. THE REFINED INSTRUMENTAL VARIABLE METHOD

The RIV algorithm for discrete-time models is applied to the following Box-Jenkins TF model form, where z^{-1} is the

[1] In this paper, the statistical meanings of 'identification' and 'estimation' are utilized : namely, 'identification' refers to the identification of a uniquely identifiable model order and structure ; while 'estimation' refers to the estimation of the parameters that characterize this identified model form.

backward shift operator, i.e. $z^{-r}y_k = y_{k-r}$:

$$y_k = \frac{B(z^{-1})}{A(z^{-1})}u_{k-\delta} + \frac{D(z^{-1})}{C(z^{-1})}e_k \qquad e_k = \mathcal{N}(0,\sigma^2) \quad (1)$$

This can also be represented in the following decomposed form, which is particularly relevant in the present instrumental variable context :

$$y_k = x_k + \xi_k \quad (2)$$

Here, the noise-free output of the system, x_k, is generated by the equation,

$$x_k = \frac{B(z^{-1})}{A(z^{-1})}u_{k-\delta} \quad (3)$$

which is the *TF system* part of the model ; and the coloured noise ξ_k is generated by the equation,

$$\xi_k = \frac{D(z^{-1})}{C(z^{-1})}e_k \quad (4)$$

which is the associated *ARMA noise* part of the model. In the equations, (1) to (4), the polynomials in z^{-1} are defined as follows :

$$\begin{array}{l} A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_n z^{-n} \\ B(z^{-1}) = b_0 + b_1 z^{-1} + \cdots + b_m z^{-m} \\ C(z^{-1}) = 1 + c_1 z^{-1} + \cdots + c_p z^{-p} \\ D(z^{-1}) = 1 + d_1 z^{-1} + \cdots + d_q z^{-q} \end{array}$$

and δ denotes a pure time delay of δ sampling intervals, each of Δt time units.

The estimation problem posed by the model (1) is to estimate the parameters in the TF system polynomials $\{A(z^{-1})\ B(z^{-1})\}$ and the ARMA noise model polynomials $\{C(z^{-1})\ D(z^{-1})\}$ based on the input-output data $Z^N = \{u_k; y_k\}_{k=1}^{N}$. The RIV solution to this problem is based on forming estimation equations for the system and noise model parameters in the form of special, pseudo-linear regression relationships [14] [9]. These estimation equations then provide the basis for an iterative (relaxation) algorithm that estimates the system and noise model parameters in separate but linked iterative sub-algorithms.

The RIV algorithm was originally devised [27] from the manipulation of the ML expressions related to the BJ model (1). The main aim of the following sub-sections is to provide the basic motivation behind this iterative RIV algorithm in a simpler and more transparent form, before outlining the algorithm in the subsequent Section IV of the paper.

A. The System TF Model

Following the usual Prediction Error Minimization (PEM) approach (which is ML estimation in the present situation because of the Gaussian assumptions on e_k), a suitable error function ε_k, is defined as follows :

$$\varepsilon_k = \frac{C(z^{-1})}{D(z^{-1})}\left[y_k - \frac{B(z^{-1})}{A(z^{-1})}u_{k-\delta}\right].$$

which can be written as,

$$\varepsilon_k = \frac{C(z^{-1})}{D(z^{-1})A(z^{-1})}\left[A(z^{-1})y_k - B(z^{-1})u_{k-\delta}\right]. \quad (5)$$

Minimization of a least squares criterion function in ε_k, measured at the sampling instants, provides the basis for stochastic estimation. However, since the polynomial operators commute in this linear case, the prefilter :

$$f_1(z^{-1}) \triangleq \frac{C(z^{-1})}{D(z^{-1})A(z^{-1})} \quad (6)$$

can be taken inside the square brackets to yield :

$$\varepsilon_k = A(z^{-1})y_k^{f_1} - B(z^{-1})u_{k-\delta}^{f_1} \quad (7)$$

or,

$$\begin{aligned} \varepsilon_k = {} & y_k^{f_1} + a_1 y_{k-1}^{f_1} + \ldots + a_n y_{k-n}^{f_1} \\ & - b_0 u_{k-\delta}^{f_1} - \ldots - b_m u_{k-\delta-m}^{f_1} \end{aligned} \quad (8)$$

where the superscripts f_1 denote that the associated variables have been prefiltered by $f_1(z^{-1})$. As a result, it is now possible to formulate the following *estimation equation* in the *pseudo-linear regression* form :

$$\text{System Estimation Equation :} \quad y_k^{f_1} = \boldsymbol{\phi}_k^T \boldsymbol{\theta}_{ab} + e_k \quad (9)$$

where,

$$\begin{aligned} \boldsymbol{\phi}_k^T &= [-y_{k-1}^{f_1} \cdots - y_{k-n}^{f_1}\ u_{k-\delta}^{f_1} \cdots\ u_{k-\delta-m}^{f_1}] \quad (10) \\ \boldsymbol{\theta}_{ab} &= [a_1 \cdots a_n\ b_0 \cdots\ b_m]^T \quad (11) \end{aligned}$$

This estimation equation is simply a way of presenting the BJ model (1) in a pseudo-linear form, so that it can provide a suitable basis for the estimation of the TF system parameter vector $\boldsymbol{\theta}_{ab}$. Thus, provided we assume that $A(z^{-1})$, $C(z^{-1})$ and $D(z^{-1})$ are known *a priori*, it forms a basis for the definition of a likelihood function and ML estimation.

There are two problems with this formulation. The most obvious one is, of course, that the $A(z^{-1})$, $C(z^{-1})$ and $D(z^{-1})$ polynomials are *not* known *a priori*. The less obvious one is that, in practical applications, the theoretical assumptions on which it is based may not be satisfied and, in particular, the noise may not have rational spectral density and cannot, therefore, be described very well by an ARMA model.

As we shall see later, both of these problems are solved by the RIV algorithm which employs an IV optimization procedure that iteratively adjusts the unknown polynomials in the BJ model (1) until they converge on an optimal solution.

B. The ARMA Noise Model

In the simplest situation where the noise ξ_k in (2) is purely white, so that $C(z^{-1}) = D(z^{-1}) = 1$, the prefilter $f_1(z^{-1})$ in (6) reduces to $1/A(z^{-1})$ and no noise model estimation is required. Estimation is particularly simple in this case and the iterative optimization approach has been termed the *Simplified Refined Instrumental Variable* (SRIV) algorithm [29]. In the more general situation where the noise can be represented by an ARMA model, the full RIV algorithm is naturally more complex and involves the separate but linked estimation of the ARMA noise model parameters. Any method of ARMA model estimation can be used and, in the original implementation of the RIV algorithm, the recursive *Approximate Maximum Likelihood* (AML) algorithm [25] was employed for this purpose (see e.g. [28]). Also, the ARMA model can be approximated by a high order AR model : indeed, this is the method used in the current implementation of the RIV algorithm in the CAPTAIN Toolbox for Matlab[2]. Another alternative, which does not provide the facility for recursive estimation, would be a standard gradient optimization algorithm limited to ARMA model estimation, as used in the PEM and ARMAX algorithms of the Matlab SID Toolbox.

Here, however, we will consider briefly another approach suggested recently [39] as an improvement on an earlier, related algorithm [30] : namely, the IVARMA algorithm outlined in Appendix 2. This has the advantage that it is motivated in a similar manner to that used above for TF system model parameter estimation. In particular, a suitable error function ε_k^n in this case is defined as follows :

$$\varepsilon_k^n = \frac{C(z^{-1})}{D(z^{-1})}\xi_k - e_k \tag{12}$$

Now, by introducing the prefilter $f_2(z^{-1})$ defined as,

$$f_2(z^{-1}) \triangleq \frac{1}{D(z^{-1})} \tag{13}$$

the error function can be written in the form,

$$\varepsilon_k^n = -D(z^{-1})e_k^{f_2} + C(z^{-1})\xi_k^{f_2} \tag{14}$$

Then, an associated estimation equation can be written in the pseudo-linear regression form :

$$\text{Noise Estimation Equation :}\quad e_k^{f_2} = \boldsymbol{\psi}_k^T\boldsymbol{\theta}_{dc} + \epsilon_k \tag{15}$$

where,

$$\boldsymbol{\psi}_k^T = [-e_{k-1}^{f_2} \cdots - e_{k-q}^{f_2}\ \xi_k^{f_2}\ \xi_{k-1}^{f_2} \cdots\ \xi_{k-p}^{f_2}] \tag{16}$$

$$\boldsymbol{\theta}_{dc} = [d_1 \cdots d_q\ 1\ c_1 \cdots\ c_p]^T \tag{17}$$

and ϵ_k is the approximation error arising from the need to replace the unobserved variables in $\boldsymbol{\psi}_k$ by their estimates (see below and Appendix 2).

As in the TF system model case, there are clear problems with the utilization of the estimation equation (15) to estimate the noise model parameters : (i) we do not have access to e_k or ξ_k ; and (ii) the $D(z^{-1})$ polynomial is not known *a priori*. But again, there are solutions to this problem.

First, if it is assumed that $A(z^{-1})$ and $B(z^{-1})$ are available, then the noise-free output x_k can be obtained from (3) and ξ_k can be obtained by reference to equation (2), i.e.,

$$\xi_k = y_k - x_k \tag{18}$$

Accessing e_k is more difficult but it is well known that a high order AR model of ξ_k yields residuals that provide a good estimate of e_k (see Appendix 2 and [4] [30]).

The above reasoning is the basis of the IVARMA algorithm. First, an estimate $\hat{\xi}_k$ of ξ_k is obtained from the latest, iteratively updated estimates of the $A(z^{-1})$ and $B(z^{-1})$ polynomials (see later (28)). Then, high order AR estimation is used to obtain a high order AR model for $\hat{\xi}_k$ and to generate an estimate $\hat{e}_k$ of e_k from the high order AR model residuals. In order to form the data vector $\boldsymbol{\psi}_k$ in (16), these estimates have to be prefiltered by $f_2(z^{-1})$, which is updated in the iterative IVARMA algorithm (see the later equations (31) and Appendix 2). These prefiltered variables are then used to construct the estimate of the vector $\hat{\boldsymbol{\psi}}_k$ given by,

$$\hat{\boldsymbol{\psi}}_k = [-\hat{e}_{k-1}^{f_2} \cdots - \hat{e}_{k-q}^{f_2}\ \hat{\xi}_k^{f_2}\ \cdots\ \hat{\xi}_{k-p}^{f_2}]^T \tag{19}$$

C. Instrumental Variable Estimation

A practical estimation methodology should be robust to violation of the theoretical assumptions on which the associated estimation procedure is based. In the case of TF model estimation, a useful technique for engendering such robustness is the exploitation of optimal IV estimation. Considering an estimation model of the following general form :

$$\text{General Estimation Equation :}\quad v_k = \boldsymbol{\varphi}_k^T\boldsymbol{\theta} + e_k \tag{20}$$

this involves the formulation of the following IV optimization problem,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \left\| \left[\frac{1}{N}\sum_{k=1}^{N}\hat{\boldsymbol{\varphi}}_k\boldsymbol{\varphi}_k^T\right]\boldsymbol{\theta} - \left[\frac{1}{N}\sum_{k=1}^{N}\hat{\boldsymbol{\varphi}}_k v_k\right]\right\|^2 \tag{21}$$

which results in the solution of the IV estimation (normal) equations :

$$\hat{\boldsymbol{\theta}} = \left[\sum_{k=1}^{N}\hat{\boldsymbol{\varphi}}_k\boldsymbol{\varphi}_k^T\right]^{-1}\sum_{k=1}^{N}\hat{\boldsymbol{\varphi}}_k v_k \tag{22}$$

where $\hat{\boldsymbol{\theta}}$ is a general IV estimate of the model parameter vector based on the input-output data $Z^N = \{u_k; y_k\}_{k=1}^N$; $\boldsymbol{\varphi}_k$ is the data vector ; and $\hat{\boldsymbol{\varphi}}_k$ is the associated IV vector.

In the case of the TF system model, $\hat{\boldsymbol{\varphi}}_k \triangleq \hat{\boldsymbol{\phi}}_k$, where,

$$\hat{\boldsymbol{\phi}}_k = [-\hat{x}_{k-1}^{f_1} \cdots - \hat{x}_{k-n}^{f_1}\ u_{k-\delta}^{f_1} \cdots\ u_{k-\delta-m}^{f_1}]^T \tag{23}$$

The initial n elements of this vector are generated from the basic instrumental variable $\hat{x}_k$, which is central to the definition of the IV vector $\hat{\boldsymbol{\phi}}_k$ at each iteration of the algorithm. This is an estimate of the noise-free output of the system generated by the following *auxiliary model* :

$$\hat{x}_k = \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})}u_{k-\delta}. \tag{24}$$

where $\hat{A}(z^{-1})$ and $\hat{B}(z^{-1})$ are estimates of the system model polynomials $A(z^{-1})$ and $B(z^{-1})$ based on the estimated parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ obtained at the previous iteration of the algorithm. In order to induce optimality, this basic IV, like the input variable u_k, has to be prefiltered by the $f_1(z^{-1})$ prefilter, i.e.,

$$\hat{x}_k^{f_1} = \frac{\hat{C}(z^{-1})}{\hat{D}(z^{-1})\hat{A}(z^{-1})}\hat{x}_k. \tag{25}$$

with the polynomials based on the estimated parameter vectors $\hat{\boldsymbol{\theta}}_{ab}$ and $\hat{\boldsymbol{\theta}}_{dc}$ obtained at the previous iteration of the algorithm.

[2] This is a Toolbox developed by the author and his colleagues over many years : see later in the Conclusions Section IX.

Note that, in the ideal but entirely hypothetical situation where the noise-free output x_k is available for measurement, the IV vector in (23) would be replaced by,

$$\mathring{\boldsymbol{\phi}}_k = [-x_{k-1}^{f_1} \cdots - x_{k-n}^{f_1}\ u_{k-\delta}^{f_1} \cdots\ u_{k-\delta-m}^{f_1}]^T \qquad (26)$$

This vector is referred to later in Section III.

In the case of the TF noise model, $\hat{\boldsymbol{\varphi}}_k \triangleq \hat{\boldsymbol{\psi}}_k$, where $\hat{\boldsymbol{\psi}}_k$ is the estimate of $\boldsymbol{\psi}_k$ in (16), as given by (19). It is now the basic noise instrumental variable, which will be denoted by $\hat{\hat{e}}_k$, that is central to the definition of the IV vector $\hat{\boldsymbol{\psi}}_k$ at each iteration of the algorithm and this is generated by the following noise auxiliary model :

$$\hat{\hat{e}}_k = \frac{\hat{C}(z^{-1})}{\hat{D}(z^{-1})}\hat{\xi}_k. \qquad (27)$$

where $\hat{C}(z^{-1})$ and $\hat{D}(z^{-1})$ are estimates of the noise model polynomials $C(z^{-1})$ and $D(z^{-1})$ based on the estimated parameter vector $\hat{\boldsymbol{\theta}}_{dc}$ obtained at the previous iteration of the algorithm ; while $\hat{\xi}_k$ is an estimate of the noise ξ_k obtained from the equation (cf. (18)),

$$\hat{\xi}_k = y_k - \hat{x}_k \qquad (28)$$

Once again, in order to induce optimality, this basic IV has to be prefiltered, this time by the $f_2(z^{-1})$ prefilter, i.e.,

$$\hat{\hat{e}}_k^{f_2} = \frac{1}{\hat{D}(z^{-1})}\hat{\hat{e}}_k. \qquad (29)$$

It is important to note the difference between $\hat{\hat{e}}_k$, which is the source of the noise model instrumental variables and is generated by the noise auxiliary model in (27), and the estimate $\hat{e}_k$ of the white noise, which is obtained via high order AR estimation (see previous discussion).

D. The IV Estimation Equations

Bearing the above motivational discussion in mind, the underlying computational aspects of the full RIV algorithm are summarized below. Here $\hat{\boldsymbol{\theta}}_{ab}$ is the estimate of the parameter vector $\boldsymbol{\theta}_{ab}$ for the system part of the model (1) and $\hat{\mathbf{P}}_{ab}$ is the associated covariance matrix. The vector $\hat{\boldsymbol{\theta}}_{dc}$ is the estimate of the parameter vector $\boldsymbol{\theta}_{dc}$ for the ARMA noise part of the model and $\hat{\mathbf{P}}_{dc}$ is its associated covariance matrix. And $\hat{\sigma}^2$ is the estimate of the variance of the white noise input e_k, as obtained from the residuals of the noise model estimation, in the usual manner. Note that because the *en bloc* estimation equations given in this summary are in a standard IV form, it is straightforward to convert them into their recursive IV equivalent (see e.g. [28]).

System Model Parameter Estimation

$$\begin{cases} \hat{\boldsymbol{\theta}}_{ab} = \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_k \boldsymbol{\phi}_k^T\right]^{-1} \sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_k y_k^{f_1} \\ \hat{\mathbf{P}}_{ab} = \hat{\sigma}^2 \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_k \hat{\boldsymbol{\phi}}_k\right]^{-1} \\ \boldsymbol{\phi}_k = [-y_{k-1}^{f_1} \cdots - y_{k-n}^{f_1}\ u_{k-\delta}^{f_1} \cdots\ u_{k-\delta-m}^{f_1}]^T \\ \hat{\boldsymbol{\theta}}_{ab} = [\hat{a}_1 \cdots \hat{a}_n\ \hat{b}_0 \cdots\ \hat{b}_m]^T \\ \hat{x}_k = \frac{\hat{B}(z^{-1})}{\hat{A}(z^{-1})}u_{k-\delta}; \quad \hat{x}_k^{f_1} = \frac{\hat{C}(z^{-1})}{\hat{D}(z^{-1})\hat{A}(z^{-1})}\hat{x}_k \\ IV vector : \\ \hat{\boldsymbol{\phi}}_k = [-\hat{x}_{k-1}^{f_1} \cdots - \hat{x}_{k-n}^{f_1}\ u_{k-\delta}^{f_1} \cdots\ u_{k-\delta-m}^{f_1}]^T \end{cases} \qquad (30)$$

Noise Model Parameter Estimation (IVARMA)

$$\begin{cases} \hat{\boldsymbol{\theta}}_{dc} = \left[\sum_{k=1}^{N} \hat{\boldsymbol{\psi}}_k \boldsymbol{\psi}_k^T\right]^{-1} \sum_{k=1}^{N} \hat{\boldsymbol{\psi}}_k \hat{e}_k^{f_2} \\ \hat{\mathbf{P}}_{dc} = \hat{\sigma}^2 \left[\sum_{k=1}^{N} \hat{\boldsymbol{\psi}}_k \hat{\boldsymbol{\psi}}_k\right]^{-1} \\ \boldsymbol{\psi}_k = [-\hat{e}_{k-1}^{f_2} \cdots - \hat{e}_{k-q}^{f_2}\ \hat{\xi}_k^{f_2}\ \hat{\xi}_{k-1}^{f_2} \cdots\ \hat{\xi}_{k-p}^{f_2}]^T \\ \hat{\boldsymbol{\theta}}_{dc} = [\hat{d}_1 \cdots\ \hat{d}_q\ 1\ \hat{c}_1 \cdots\ \hat{c}_p]^T \\ \hat{\xi}_k = y_k - \hat{x}_k; \quad \hat{\xi}_k^{f_2} = \frac{1}{\hat{D}(z^{-1})}\hat{\xi}_k \\ \hat{e}_k \text{(high order AR residuals)}; \quad \hat{e}_k^{f_2} = \frac{1}{\hat{D}(z^{-1})}\hat{e}_k \\ \hat{\hat{e}}_k = \frac{\hat{C}(z^{-1})}{\hat{D}(z^{-1})}\hat{\xi}_k; \quad \hat{\hat{e}}_k^{f_2} = \frac{1}{\hat{D}(z^{-1})}\hat{\hat{e}}_k \\ IV vector : \\ \hat{\boldsymbol{\psi}}_k = [-\hat{\hat{e}}_{k-1}^{f_2} \cdots - \hat{\hat{e}}_{k-q}^{f_2}\ \hat{\xi}_k^{f_2}\ \cdots\ \hat{\xi}_{k-p}^{f_2}]^T \end{cases} \qquad (31)$$

The iterative form of the algorithm that utilizes the estimation equations in (30) and (31), or their recursive equivalents, is described later in Section IV. First, however, it is necessary to consider the theoretical justification for the decomposition of the estimation into the above separate, but inter-linked, system and noise model estimation modules.

III. Theoretical Justification of the RIV Method

The original maximum likelihood development of the RIV method [27] [43][28], as well as the related motivation outlined in the previous section, are based on the decomposition of the estimation problem into two separate but inter-linked subproblems : first the estimation of the system transfer function model parameters under the assumption that the noise model parameters are known ; and second, the estimation of the

ARMA noise model parameters under the assumption that TF system model parameters are known. This approach is then carried over to the formulation of both the discrete-time RIV and continuous-time RIVC algorithms (see later Section VI). The justification for such a simplifying approach is given by the following modified version of the theorem due originally to Pierce [12] and formulated in the present control theoretic form by Young and Jakeman [43][44][28].

Theorem ([27] [43] after [12])

If, in the Box-Jenkins TF model (1) :

(i) the e_k are independent and identically distributed with zero mean, variance σ^2 and skewness and kurtosis κ_1 and κ_2 ;
(ii) the parameter values are admissible (i.e. the model is stable and identifiable), and
(iii) the u_k are persistently exciting ;

then the ML parameter estimates, obtained from a data set of N samples, possess a limiting normal distribution, such that the following results hold :

1. the asymptotic covariance matrix of the estimation errors associated with the estimate of the system parameters $\{a_i; b_j\}$ is of the form :

$$\mathbf{P}_{ab} = \frac{\sigma^2}{N}\left[p\lim \frac{1}{N}\sum_{k=1}^{N} \mathring{\boldsymbol{\phi}}_k \mathring{\boldsymbol{\phi}}_k^T\right]^{-1}$$

2. the estimates of the noise model parameters $\{c_i; d_j\}$ are *asymptotically independent* of the $\{a_i; b_j\}$ estimates and have an error covariance matrix of the form :

$$\mathbf{P}_{dc} = \frac{\sigma^2}{N}\left[E\left\{\sum_{k=1}^{N} \boldsymbol{\psi}_k \boldsymbol{\psi}_k^T\right\}\right]^{-1}$$

3. the estimate $\hat{\sigma}^2$ has asymptotic variance $(2\sigma^4/N)(1 + 0.5\kappa_2)$ and, if $\kappa_1 = 0$, is independent of the above estimates.

Here, $\mathring{\boldsymbol{\phi}}_k$ and $\boldsymbol{\psi}_k$ are, respectively, the underlying 'ideal' IV vectors for the system and noise estimation models used as the basis for the RIV algorithm outlined previously (see equations (26) and (16), respectively). It is clear, therefore, that, upon convergence of the RIV algorithm, the elements of the vectors $\hat{\boldsymbol{\phi}}_k$ and $\hat{\boldsymbol{\psi}}_k$ will converge in probability (see e.g. [28]) to the equivalent elements of these ideal vectors. Consequently, $\hat{\boldsymbol{\phi}}_k$ and $\hat{\boldsymbol{\psi}}_k$ can be used to compute estimates $\hat{\mathbf{P}}_{ab}$ and $\hat{\mathbf{P}}_{dc}$ of the parametric covariance matrices $\mathbf{P}_{ab}$ and $\mathbf{P}_{dc}$ defined in the Theorem, as shown in equations (30) and (31).

Proof See Pierce [12], using the formulation of the Theorem in [43][44][28].

Comment The selection of the TF model form is often decided by the predilection of the analyst. However, the above theorem suggests that the ML estimation of the BJ model has one particularly attractive statistical advantage : namely, the asymptotic independence of the system and noise model parameter estimates (i.e. the covariance matrix is block diagonal), thus justifying the system-noise model decomposition that is an essential element of the iterative RIV/RIVC algorithms. In addition, Jakeman and Young [6][7] have investigated this aspect of the RIV algorithm further and showed that, while a BJ model estimation algorithm, such as RIV, is able to estimate an ARMAX model, without any approximation, from data generated by a stochastic ARMAX system (the ARMAX model is simply a constrained BJ model), the reverse is not true. If the polynomials are not to share common factors, then the ARMAX form of the BJ model is :

$$A(z^{-1})y_k = B(z^{-1})u_{k-\delta} + F(z^{-1})e_k$$
$$F(z^{-1}) = \frac{D(z^{-1})A(z^{-1})}{C(z^{-1})}$$

so that the noise polynomial is not normally of finite dimension and estimation of a finite size polynomial will imply an approximation.

IV. Summaries of the Iterative RIV Algorithms

In this section, the simplified SRIV and full RIV algorithms are outlined to clarify the nature of the iterative updating.

The SRIV Algorithm (additive white noise)

Step 1. Use the standard *en-bloc* or recursive Least Squares algorithms (*i.e.* ARX model estimation) to generate an initial (iteration 1) estimate of the TF system model parameter vector $\hat{\boldsymbol{\theta}}_{ab}$.

Step 2. Iterative IV estimation with prefilters.
for $j = 2 : convergence$

(i) Generate the IV series $\hat{x}_{j,k}$ from the auxilary model :

$$\hat{x}_{j,k} = \frac{\hat{B}_{j-1}(z^{-1})}{\hat{A}_{j-1}(z^{-1})}u_{k-\delta}$$

with the estimated polynomials based on the estimated parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ obtained at the previous iteration of the algorithm.

(ii) Prefilter y_k, u_k and $\hat{x}_{j,k}$ with $f_1(z^{-1}) = 1/\hat{A}_{j-1}(z^{-1})$.

(iii) Based on these prefiltered data, compute the estimate of the TF system model parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ using the *en bloc* algorithm in (30) or its recursive equivalent.

end

Step 3. Compute the estimated parametric error covariance matrix associated with the parameter estimates from the following expression (see (30) and Section III) :

$$\hat{\mathbf{P}}_{ab} = \hat{\sigma}^2\left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_k \hat{\boldsymbol{\phi}}_k\right]^{-1}$$

It should be noted that this SRIV algorithm is very simple in comparison to the RIV algorithm outlined below and so it is much more computationally efficient. This allows for very rapid model order/structure identification using the associated RIVID algorithm in the CAPTAIN Toolbox (see later example).

The fact that the SRIV algorithm yields consistent estimates of the system model parameters, even in the coloured noise situation, means that these estimates can provide the information required for the initiation of the full RIV algorithm. In particular :

1. The SRIV estimate of the $A(z^{-1})$ polynomial provides an initial estimate of the associated prefilter polynomial.
2. the initial identification (the polynomial orders p and q) and estimation of the ARMA(p, q) noise model can be based on the estimate $\hat{\xi}_k$ from the equation :

$$\hat{\xi}_k = y_k - \hat{x}_k \tag{32}$$

where $\hat{x}_k$ is the estimate of noise-free output of the system from equation (24) (and also the source of the IVs). Any algorithm for the ML estimation of ARMA models can be used, including the IVARMA algorithm described in the previous Section II and Appendix 2. Of course, if the noise model is identified as an AR(p) process, then simple least squares estimation of the AR model can be employed. Indeed, as mentioned previously, the algorithm could be simplified by always estimating an AR noise model, under the assumption that this can approximate any stationary ARMA model (see Appendix 2). The RIV algorithm in the CAPTAIN Toolbox has used this approach very successfully for many years (although it will shortly be modified to allow for a full ARMA noise model, if this is required by the user).
3. Of course, because of the pseudo-linear regression approach used in RIV, the implementation of recursive RIV and SRIV algorithms is obvious (see e.g. [28]) and is an option in the CAPTAIN version of these algorithms.

With the above observations in mind, the main steps in the RIV algorithm are as follows :

The Full RIV Algorithm (additive ARMA noise)

Step 1. Apply the SRIV algorithm in the standard manner and compute an initial (iteration 1) estimate of the TF system model parameter vector $\hat{\boldsymbol{\theta}}_{ab}$.

Step 2. Iterative IV estimation with prefilters.
for $j = 2 : convergence$

(i) Generate the IV series $\hat{x}_{j,k}$ from the auxilary model :

$$\hat{x}_{j,k} = \frac{\hat{B}_{j-1}(z^{-1})}{\hat{A}_{j-1}(z^{-1})} u_{k-\delta}$$

with the polynomials based on the estimated parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ obtained at the previous iteration of the algorithm.

(ii) Obtain an estimate of the noise model parameter vector $\hat{\boldsymbol{\theta}}_{dc}$ based on the estimated noise sequence $\hat{\xi}_k$ from equation (27), using a selected ARMA estimation algorithm (in the case of the IVARMA algorithm the *en-bloc* solution in (31) or its recursive equivalent).

(iii) Prefilter the input u_k, output y_k and instrumental variable $\hat{x}_{j,k}$ signals by the filter

$$f_1(z^{-1}) = \frac{\hat{C}_j(z^{-1})}{\hat{D}_j(z^{-1})\hat{A}_{j-1}(z^{-1})}$$

with the polynomials based on the estimated parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ obtained at the previous iteration of the algorithm and $\hat{\boldsymbol{\theta}}_{dc}$ obtained in (ii).

(iv) Based on these prefiltered data, compute the estimate of the TF system model parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ using the *en bloc* IV algorithm in (30) or its recursive equivalent.

end

Step 3. Compute the estimated parametric error covariance matrices associated with the parameter estimates from the following expressions (see (30), (31) and Section III) :

$$\hat{\mathbf{P}}_{ab} = \hat{\sigma}^2 \left[\sum_{k=1}^{N} \hat{\boldsymbol{\phi}}_k \hat{\boldsymbol{\phi}}_k \right]^{-1} \qquad \hat{\mathbf{P}}_{dc} = \hat{\sigma}^2 \left[\sum_{k=1}^{N} \hat{\boldsymbol{\psi}}_k \hat{\boldsymbol{\psi}}_k \right]^{-1}$$

Normally, only 4 iterations are required for convergence. In general, however, it is safer to either specify more than 4 iterations or, as indicated in the algorithm above, use an automatic convergence rule based on the change in the parameter estimates.

A. Convergenceof the Iterative Algorithms

The SRIV and RIV algorithms are rapidly convergent, normally converging to a stationary solution in only a few iterations (typically 3-4). Moreover, they have been available for many years in the CAPTAIN Toolbox, where they have been very robust in simulation and practical applications, converging in all cases where the model is identifiable from the data. The convergence of these algorithms has not been considered theoretically but the SRIV algorithm is quite similar to the iterative least squares algorithm of Steiglitz and McBride [15], the convergence of which has been established in the case of white additive noise [16]. Moreover, the inherent optimal instrumental variable nature of the SRIV algorithm removes the limitations of the Steiglitz and McBride-type algorithm in the coloured noise situation [16]. The convergence of the more complex RIV algorithm has not been established theoretically but, again, it has been used successfully for many years in the form where the noise is modelled as an AR process (later referred to as RIV-AR), without any convergence problems in identifiable situations.

V. Discrete-Time Examples

A. Discrete-Time Simulation Example 1

Here, the RIV algorithm is applied to $N = 1700$ samples of simulated data generated by the following Box-Jenkins TF model :

$$y_k = \frac{0.016 + 0.026z^{-1} - 0.0375z^{-2}}{1 - 1.6252z^{-1} + 0.642z^{-2}} u_k + \frac{1 + 0.5z^{-1}}{1 - 0.85z^{-1}} e_k$$
$$u_k = \mathcal{N}(0, 8.8) \qquad e_k = \mathcal{N}(0, 0.0009)$$

This is a 'stiff' dynamic system, with widely spaced eigenvalues, and it has a reasonable noise level (noise/signal ratio of 0.33 by standard deviation). The TF can be decomposed into a parallel connection of three TFs : one a simple gain, 0.015964; and two with first order TFs having time constants of 2.6 and 18.7 samples. This is a common model form in the environmental sciences but it appears to pose some problems for the PEM algorithm.

Table I compares the results of single run and MCS analysis, based on 100 realizations, for the RIV, SRIV, PEM and IV4 algorithms (the latter two from the latest Matlab 7.2.0.283 (R2006a) SID Toolbox). It also presents the results obtained with the RIV algorithm when the ARMA model is approximated by an AIC (see Appendix 2 and [1]) identified AR(5) model (RIV-AR). As would be expected because of their common basis in maximum likelihood estimation, both of the RIV and PEM algorithms perform similarly when convergence occurs, with the single run predicted standard errors on the parameter estimates matching reasonably the standard deviations computed from the MCS analysis. However, in this example, the PEM algorithm fails to converge in 9 of the 100 realizations (these realizations were removed in computing the statistics shown in the Table), while the RIV algorithm does not fail at all. A plot of the results is shown in Figure 1, with RIV in the left hand panel and PEM in the right hand panel.

The SRIV algorithm also performs well : in fact, in terms of the single run and MCS estimated mean parameter estimates, it performs better than PEM in the sense that it has no failures. However, we see that its estimated standard errors are too optimistic, as might be expected. Also, note that an estimate of an ARMA model for the noise can be obtained by applying the IVARMA algorithm separately to the residual noise estimate obtained from the SRIV estimation results. For illustration, this is shown in the Table only for the single run case. Of course, it had no influence of the TF system model parameter estimates and it was not computed at all in the MCS simulation results presented below this. Note that this SRIV algorithm is computationally a quite efficient algorithm (a little faster than PEM in this case and only marginally slower than IV4) and so, as pointed out previously, it provides the best algorithm for initial model structure/order identification analysis.

The results in Table I are typical of the performance comparison in the case of examples such as that considered here. The poorer performance of the PEM algorithm appears to be due to the 'stiff' nature of the TF model in this example and a consequent failure to converge from the initial conditions specified for the parameter estimates in the PEM gradient optimization algorithm. It is clear from the results that the RIV algorithm does not suffer from this problem and is always providing statistically consistent and efficient estimates. It is, in other words, another approach to optimal estimation of discrete-time TF models of the Box-Jenkins type (which includes the ARMAX model as a special constrained case, as pointed out previously).

Finally, we see that the RIV-AR results are comparable with those of RIV (for convenience, the AR(5) noise model parameter estimates are not shown). They demonstrate how this 'approximate' implementation of the RIV algorithm is appealing because it is computationally much more efficient than RIV and yet performs similarly in most cases. For this reason, as pointed out previously, it is the current implementation used in the CAPTAIN Toolbox.

The IV4 algorithm, which also uses an AR noise model, has been likened to the RIV algorithm but it is, in fact, quite different (see Conclusions Section IX) and, in general, does not perform nearly as well as RIV (although it is computationally much more efficient : about the same as

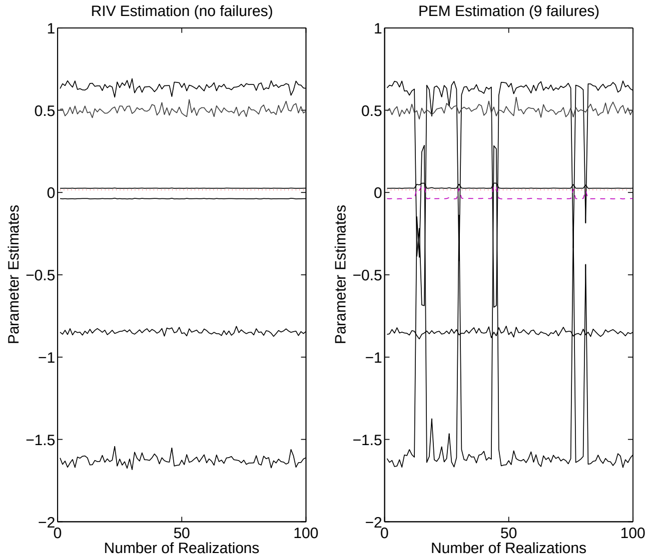

Fig. 1. MCS comparison between RIV and PEM on a difficult example

SRIV). In this case, it produces reasonable results but they are noticeably poorer than those of RIV and SRIV. However, like them, it has no failures amongst the MCS realizations. This is not always the case, however, as we see in the next simulation example.

B. *Discrete-Time Simulation Example 2*

This example is concerned with a simulation model based on a $[2\ 2\ 4]$ TF model identified and estimated from the real effective rainfall-flow data shown in Figure 2. It is a re-appraisal of an example used in a previous comparative study [49] but now using the latest available versions of the RIV, PEM and IV4 algorithms[3]. The lower panel in Figure 2 shows the hourly flow y_k measured in a river over most of a year (7500 hours or 312.5 days) ; while the upper panel shows the associated 'effective rainfall' u_k (see next Section V-C). The simulation data are generated by passing this effective rainfall input through the model, with its parameters set to those estimated from the real data. The output is then contaminated by white noise with variance $\sigma^2 = 5$, giving a noise/signal ratio (by standard deviation) of 0.62.

Table II compares the results of the RIV estimation (here effectively SRIV because the additive noise is white) with those obtained using the PEM and IV4 algorithms. Again, as would be expected, both of the RIV and PEM algorithms perform similarly *when convergence occurs*, with the single run predicted standard errors on the parameter estimates matching the standard deviations computed from the MCS analysis. However, the PEM algorithm has a quite high failure rate of 19.4% : it fails to converge satisfactorily in 24 of the 124 realizations (these realizations were removed in computing the statistics shown in the Table) while, as in the previous example, the RIV algorithm does not fail at all.

[3]the previous results obtained with the PEM algorithm were considerably worse than those shown here but it would appear that some improvements have been introduced into the algorithm since the original analysis was carried out.

Method	Value	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{c}_1$	$\hat{d}_1$	Failures
	True Values	-1.6252	0.642	0.016	0.026	-0.0375	-0.85	0.5	
RIV (SR)	$\hat{\boldsymbol{\theta}}$	-1.6112	0.6331	0.0162	0.0264	-0.0375	-0.845	0.494	
	SE	0.0449	0.0334	0.0003	0.0007	0.0018	0.01	0.02	
RIV (MCS)	$\hat{\boldsymbol{\theta}}$	-1.6255	0.642	0.0160	0.0260	-0.0375	-0.847	0.501	
	SD	0.0254	0.0200	0.0002	0.0004	0.0010	0.01	0.02	
SRIV (SR)	$\hat{\boldsymbol{\theta}}$	1.6312	0.649	0.0165	0.0256	-0.0379	-0.848	0.485	
	SE	0.0199	0.0159	0.0007	0.0014	0.0009	0.010	0.02	
SRIV (MCS)	$\hat{\boldsymbol{\theta}}$	-1.6164	0.635	0.0159	0.0262	-0.0372	-	-	
	SD	0.0535	0.0428	0.0005	0.0013	0.0014	-	-	
PEM (SR)	$\hat{\boldsymbol{\theta}}$	-1.5939	0.6204	0.0161	0.0266	-0.0367	-0.849	0.502	
	SE	0.0650	0.0480	0.0003	0.0010	0.0026	0.01	0.02	
PEM (MCS : 91/100)	$\hat{\boldsymbol{\theta}}$	-1.6166	0.6354	0.0160	0.0261	-0.0371	-0.849	0.502	9
	SD	0.0400	0.030	0.0003	0.0007	0.00164	0.01	0.02	
RIV-AR (SR)	$\hat{\boldsymbol{\theta}}$	-1.611	0.6328	0.0162	0.0263	-0.0375	-	-	
	SE	0.0440	0.0328	0.0003	0.0007	0.0017	-	-	
RIV-AR (MCS)	$\hat{\boldsymbol{\theta}}$	-1.619	0.6371	0.016	0.0261	-0.0373	-	-	
	SD	0.0356	0.0270	0.0002	0.0006	0.0014	-	-	
IV4 (SR)	$\hat{\boldsymbol{\theta}}$	-1.568	0.6019	0.0162	0.0270	-0.0357	-	-	
	SE	0.0991	0.0716	0.0003	0.0016	0.0040	-	-	
IV4 (MCS)	$\hat{\boldsymbol{\theta}}$	-1.6090	0.6298	0.0160	0.0262	-0.0369	-	-	
	SD	0.0513	0.0384	0.0002	0.0008	0.0021	-	-	

TABLE I

MONTE CARLO SIMULATION RESULTS FOR SIMULATION EXAMPLE 1 : SE DENOTES THE STANDARD ERROR ON THE ESTIMATES ; SD THE STANDARD DEVIATION OF THE MCS REALIZATIONS ; SR THE SINGLE RUN RESULTS ; AND MCS THE RESULTS FROM MCS ANALYSIS BASED ON 100 RANDOM REALIZATIONS.

The performance of IV4 is much worse : it fails to converge satisfactorily in 114 of the 124 realizations, an unacceptable failure rate of 91.9%.

This model is an even 'stiffer' dynamic system than the previous example, with time constants of 6.5 and 605 hours[4], and PEM's problems seem, once again, to be connected with this property. When the algorithm fails to converge on the correct system, it most often converges on a false optimum with one root of the denominator polynomial $A(z^{-1})$ negative and very close to the unit circle ; while the other is positive and just greater than 0.9 (a typical example is $\{-0.99943, 0.91657\}$). And, in all cases such as this, the explanation of the data is poor : e.g. the coefficient of determination R_T^2 (see Appendix 1), based on the simulated noise-free output $\hat{x}_k$, is only *circa* 0.85, compared with R_T^2 very close to unity when correct convergence occurs (as in all the RIV estimated models).

C. Discrete-Time Practical Example

The Leaf River watershed is a humid watershed with an area of 1944 km^2 located north of Collins, Mississippi. It has been the subject of recent research in which the Bayesian-inspired *Ensemble Kalman Filter* has been used as an approach to flow forecasting [11]. This is a very computationally expensive methodology, however, requiring the generation of many Monte Carlo realizations at each recursive update of the algorithm. Moreover, it is not particularly well justified in practical terms because, as we shall see, the only significant nonlinearity occurs at the input to an otherwise linear process (i.e. it is identified as a Hammerstein process). The objective of the modelling in this case is, therefore, to identify a *Data-Based Mechanistic* (DBM) model (see [32] [38] and the references cited therein) that has a satisfactory physical interpretation and can provide the basis for much simpler and *computationally inexpensive* real-time flow forecasting.

The relationship between daily measures of rainfall r_k and flow y_k is nonlinear, so *State-Dependent Parameter* (SDP) estimation (see later and [33] [35] [48] [34]) is used to identify the model structure. This identifies a Hammerstein-type model with an input nonlinearity, in which the SDP is dependent on the flow, acting as a surrogate measure of the soil moisture (catchment storage) effects. The function of this nonlinearity can be explained by considering the nature of the rainfall-flow process. When the flow is small, the soil is corresponding dry and so rainfall falling on it tends to be absorbed. As a result, it produces less flow in the river channel than it does if the flow happens to be high and the soil is saturated with water. The output of this nonlinearity u_k, which is the input to the linear TF part of the model, is called the 'effective rainfall' because it is that part of the rainfall that is not trapped by the soil and is effective in causing flow variations.

Although an unconstrained TF model explains the data well ($R_T^2 = 0.86$: i.e. 86% of the output variance explained by

[4]Note that these time constants are sensitive to the estimated model parameter values and were computed from estimates with more decimal places than those shown in Table II.

Method	Value	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_0$	$\hat{b}_1$	Failures
	True Values	-1.8563	0.8565	0.0545	-0.0542	
RIV (SR)	$\hat{\boldsymbol{\theta}}$	-1.8575	0.8578	0.0543	-0.0541	
	SE	0.0028	0.0027	0.0009	0.0009	
RIV (MCS)	$\hat{\boldsymbol{\theta}}$	-1.8560	0.8563	0.0545	-0.0543	
	SD	0.0027	0.0026	0.0008	0.0008	
PEM (SR)	$\hat{\boldsymbol{\theta}}$	-1.8561	0.8563	0.0546	-0.0543	
	SE	0.0028	0.0028	0.0008	0.0008	
PEM (MCS)	$\hat{\boldsymbol{\theta}}$	-1.8585	0.8587	0.0541	-0.0539	24
	SD	0.0027	0.0027	0.0008	0.0009	
IV4 (MCS)	$\hat{\boldsymbol{\theta}}$	0.0954	-0.8806	0.0585	0.0481	114
	SD	13.0	11.6	0.0224	0.694	

TABLE II

MONTE CARLO SIMULATION RESULTS FOR SIMULATION EXAMPLE 2 : SE DENOTES THE STANDARD ERROR ON THE ESTIMATES ; SD THE STANDARD DEVIATION OF THE MCS REALIZATIONS ; SR THE SINGLE RUN RESULTS ; AND MCS THE RESULTS FROM MCS ANALYSIS BASED ON 100 RANDOM REALIZATIONS (OUT OF 124 TOTAL REALIZATIONS).

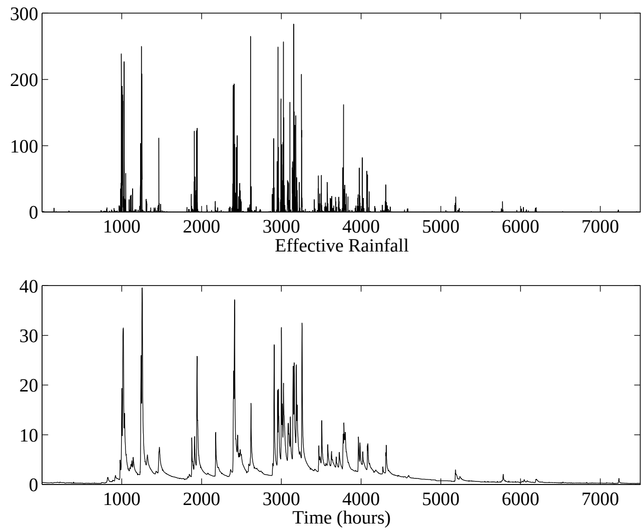

Fig. 2. Hourly effective rainfall (upper panel) and river flow (lower panel) data from a typical catchment. The effective rainfall is used as the input signal for the second simulation example.

the simulated model output $\hat{x}_k$), the TF poles are complex ($0.9533; 0.5500 \pm 0.3180j$) and there is no obvious physical explanation for such characteristics. Consequently, in order to satisfy the tenets of DBM modelling, which require that the model should be capable of interpretation in physically meaningful terms, a constrained form of SRIV estimation is utilized. This involves a novel optimization routine in which the constrained real pole parameters α_1 and α_2 in the model equation (33) below, as well as the parameter γ in the associated nonlinear input model equation (34), are continually optimized by nonlinear least squares and inserted as *a priori* known parameters in a constrained version of the SRIV algorithm, that then provides the estimates of the TF numerator parameters $b_i, i = 0, 1, \cdots, 3$. The optimization cost function is based on the error $y_k - \hat{x}_k$ between the measured output y_k and the output $\hat{x}_k$ of the complete

TF	Slow	Quick 1	Quick 2	Inst.
Static Gain	0.201	0.238	0.433	0.068
Partition %	21.4	25.3	46.1	7.2
Resid. Time (d)	24.8	1.23	2.46	0.0

TABLE III

TF DECOMPOSITION

constrained model.

$$y_k = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{(1 + \alpha_1 z^{-1})(1 + \alpha_2 z^{-1})^2} u_k + \xi_k \qquad (33)$$

$$u_k = (1 - e^{\gamma y_k}) r_k \qquad (34)$$

The estimation results obtained in this manner are as follows, where the figures in parentheses are the estimated standard errors :

$$\begin{aligned} \hat{\alpha}_1 &= -0.961(0.008); \hat{\alpha}_2 = -0.443(0.01) \\ \hat{b}_0 &= 0.068(0.004); \hat{b}_1 = 0.0151(0.009) \\ \hat{b}_2 &= 0.0125(0.009); \hat{b}_3 = -0.0841(0.004) \\ \hat{\gamma} &= 0.0124(0.0005) \end{aligned}$$

The nonlinear input function in (34) is a simple exponentially rising function of y_k that was initially identified in this form using state-dependent parameter estimation (see later and Section VIII). This model explains the data as well as the unconstrained version, and the noise ξ_k is identified as either an AR(5) or ARMA$(3, 2)$ process. The estimation and validation results are shown in Figure 3, where we see that the model based on the estimation data set continues to explain the flow behaviour in the validation data measured some nine years later, with a $R_T^2 = 0.89$ (i.e. marginally better than that obtained on the estimation data set) based only on the rainfall and without any re-estimation of the model parameters.

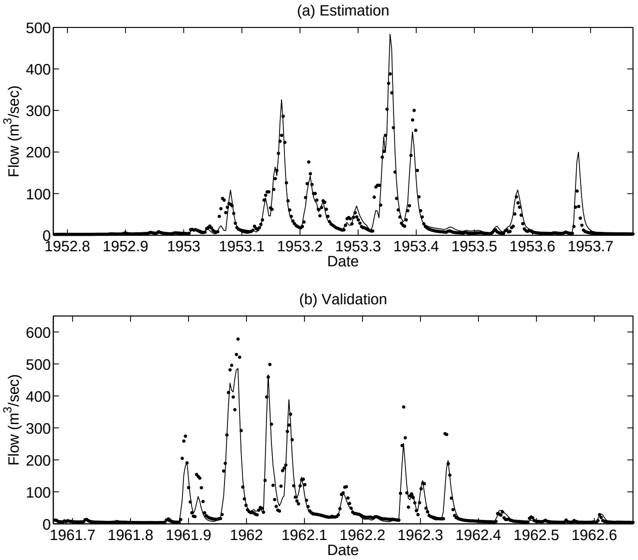

Fig. 3. Estimation and validation results for the final parameterized DBM model

The physical nature of the TF model (33) can be investigated by using partial fraction expansion to decompose it into a variety of forms. However, the most physically meaningful decomposition obtained in this manner is the parallel pathway form shown in Figure 4 (see next page), where the first order TF blocks are defined by the TF decomposition in Table III. This makes sense in hydrological terms, with the quick residence time (time constant) blocks of 1.23 ($\times$2) and 2.46 days, respectively, explaining the surface process dynamics and the slow residence time block (24.8 days) representing the groundwater effects.

In this example, the main reason for modelling is to facilitate adaptive, real-time flow forecasting based on a modified version of the *Kalman Filter* (KF). For these forecasting purposes, the TF model is transformed into a state-space form with the state variables in the state vector $\hat{\boldsymbol{x}}_k$ defined as the outputs of the parallel pathways defined in Table III and shown in Figure 4. The associated KF forecasting engine is based on the approach used previously by the author and his colleagues (e.g. [36] [13]) and is implemented in the following prediction-correction form :

Prediction :

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k-1} &= \mathbf{F}\hat{\boldsymbol{x}}_{k-1} + \mathbf{G}(1-e^{\gamma y_k})r_k \Leftarrow \text{ nonlinear input} \\
\mathbf{P}_{k|k-1} &= \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \sigma_k^2\mathbf{Q}_r \Leftarrow \text{NVR matrix} \\
\hat{y}_{k|k-1} &= \mathbf{h}^T\hat{\boldsymbol{x}}_{k|k-1} \\
\sigma_k^2 &= \sigma_k^2(y_k^2) \Leftarrow \text{ heteroscedastic noise variance}
\end{aligned}$$

Correction :

$$\begin{aligned}
\hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_{k|k-1} + \mathbf{\Pi}_k \cdot \left\{y_k - \hat{y}_{k|k-1}\right\} \\
\mathbf{\Pi}_k &= \mathbf{P}_{k|k-1}\mathbf{h}[\sigma_k^2 + \mathbf{h}^T\mathbf{P}_{k|k-1}\mathbf{h}]^{-1} \\
\mathbf{P}_k &= \mathbf{P}_{k|k-1} - \mathbf{\Pi}_k\mathbf{h}^T\mathbf{P}_{k|k-1}
\end{aligned}$$

Figure 5 shows the one-day-ahead forecasting results obtained when the above forecasting algorithm is applied to a part of the validation data.

A novel feature of this KF implementation is its exploitation of SDP elements to model the effective rainfall nonlinearity and the state-dependent heteroscedasticity in the measurement noise (shown by arrows in the above modified KF algorithm). Note how these methodological innovations provide not only a good flow forecast (normal Coefficient of Determination based on the forecasting errors of $R^2 = 0.90$, with residual variance of 613 $cumecs^2$, compared with 1207 $cumecs^2$ for a naive forecast set equal to the previous day's measured flow) but also forecast standard errors that capture the changing uncertainty associated with the flow forecasts. In the latter regard, note how the standard error bounds on the forecasts are functions of the flow, with larger standard errors at the larger flows.

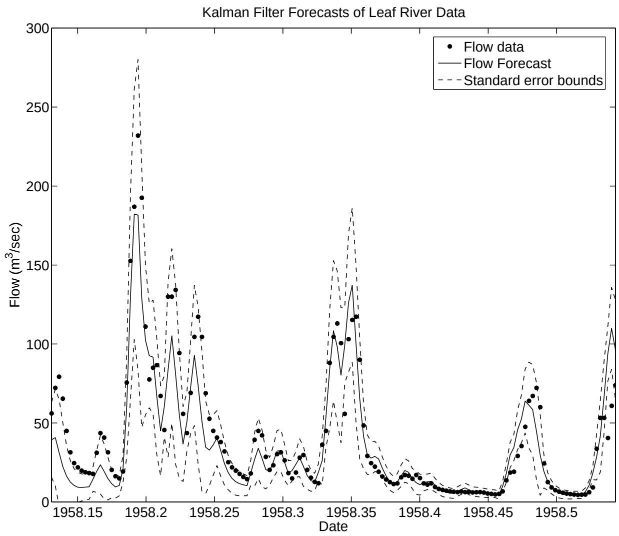

Fig. 5. Nonlinear Kalman Filter forecasting results for the validation data set

VI. The Continuous-Time RIVC Algorithm

The continuous-time RIVC algorithm is concerned with the estimation of the parameters in the following multi-order differential equation model based on discrete-time, sampled data measurements of the input and output variables :

$$\begin{aligned}
&\frac{d^n x(t)}{dt^n} + a_1\frac{d^{n-1}x(t)}{dt^{n-1}} + \cdots + a_n x(t) \\
&\qquad = b_0\frac{d^m u(t-\tau)}{dt^m} + \cdots + b_m u(t-\tau)
\end{aligned}$$

or,

$$\begin{aligned}
&x^{(n)}(t) + a_1 x^{(n-1)}(t) + \ldots + a_n x^{(0)}(t) \\
&\qquad = b_0 u^{(m)}(t-\tau) + \ldots + b_m u^{(0)}(t-\tau)
\end{aligned}$$

In transfer function (TF) terms, this takes the form :

$$x(t) = \frac{B(s)}{A(s)}u(t-\tau), \tag{35}$$

with

$$\begin{aligned}
B(s) &= b_0 s^m + b_1 s^{m-1} + \cdots + b_m, \\
A(s) &= s^n + a_1 s^{n-1} + \cdots + a_n,
\end{aligned}$$

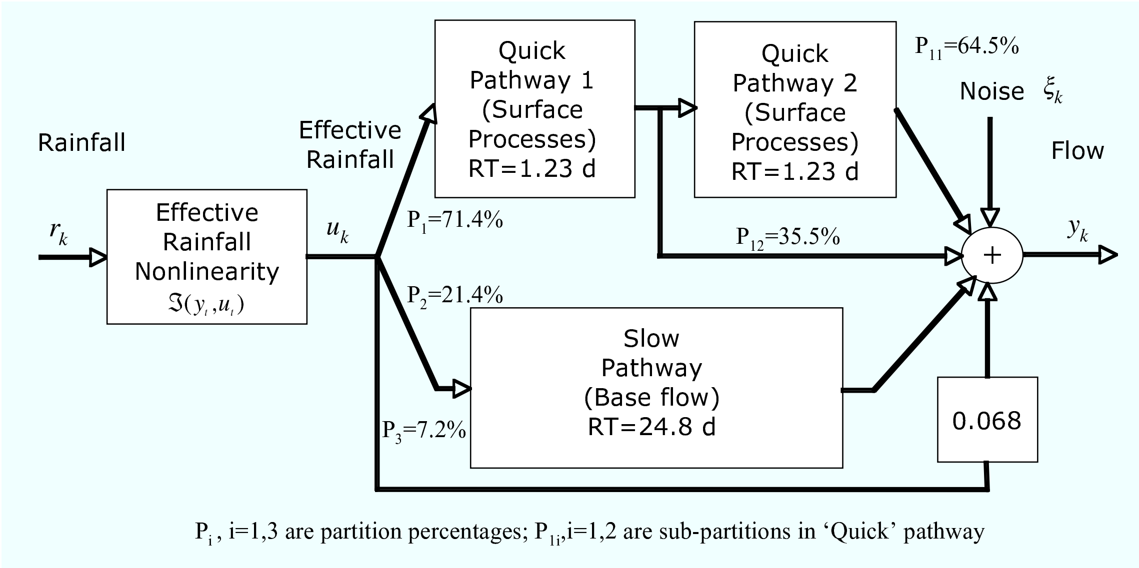

Fig. 4. Physically meaningful decomposition of the DBM model into 3 parallel pathways with partitions $P_i, i = 1, 2, 3$. RT denotes the Residence Time (time constant) in days (d).

where s is the differential operator, *i.e.* $s^p x(t) = \frac{d^p x(t)}{dt^p}$. It is assumed that the input signal $\{u(t), t_1 < t < t_N\}$ is applied to the system and that this input and the output $x(t)$ are sampled at discrete times $t_1, \cdots, t_N$, not necessarily uniformly spaced. Note that for simplicity of presentation in this paper, the nomenclature for the TF model parameters used here is the same as that used in the discrete-time case. However, it should be stressed that these parameters are associated with the continuous-time TF model and, of course, both their meaning and values are quite different.

In the case of uniformly sampled data at a sampling interval Δt, the measured output $y(t_k)$, where $t_k = k\Delta t$, is assumed to be corrupted by an additive measurement noise $\xi(t_k)$,

$$y(t_k) = x(t_k) + \xi(t_k) \qquad (36)$$

where $x(t_k)$ is the hypothetical noise-free, deterministic output of the system and, as in the discrete-time case, $\xi(t_k)$ is modelled as a discrete-time ARMA process, i.e.,

$$\xi(t_k) = \frac{D(z^{-1})}{C(z^{-1})} e(t_k) \qquad e(t_k) = \mathcal{N}(0, \sigma^2) \qquad (37)$$

The estimation problem posed in this manner is to estimate the parameters of the differential equation model from N sampled measurements of the input and output $Z^N = \{u(t_k); y(t_k)\}_{k=1}^{N}$. In the case of non-uniform sampling, the sampling interval Δt will itself be a function of the sampling integer k, i.e. $\Delta t = \Delta t(k)$.

In this continuous-time situation, the ARMA noise estimation model remains in the form (15), since the noise model is not changed. However, the TF system estimation model at the k^{th} sampling instant is written in the following pseudo-linear regression form :

$$y_f^{(n)}(t_k) = \boldsymbol{\phi}_f^T(t_k)\boldsymbol{\theta}_{ab} + e(t_k) \qquad (38)$$

$$\boldsymbol{\phi}_f^T(t_k) = [-y_f^{(n-1)}(t_k) \cdots - y_f^{(0)}(t_k) u_f^{(m)}(t_k) \cdots u_f^{(0)}(t_k)] \qquad (39)$$

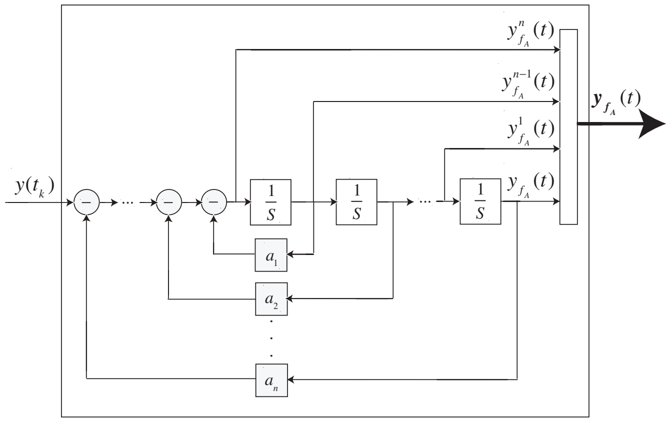

Fig. 6. Generation of filtered derivatives for the output $y(t)$ by the prefilter $1/A(s)$. This is the inside of the block marked $\mathbf{A}$ in Figure 7 below (based on Figure 2(b) in [44]).

$$\boldsymbol{\theta}_{ab} = [a_1 \ \ldots \ a_n \ b_0 \ \ldots \ b_m]^T . \qquad (40)$$

where now the subscript f denotes *hybrid prefiltering* which is completely analogous in function to that used in the discrete-time modelling situation, but involves a combination of continuous and discrete-time filters. First, the prefiltered derivatives are obtained as the inputs to the integrators in the continuous-time implementation of the initial prefilter $1/A(s)$, as shown in Figure 6[5]. These are then sampled at the sampling interval Δt, prior to discrete-time prefiltering using the inverse noise filter $C(z^{-1})/D(z^{-1})$, as shown in Figure 7. Of course, as in the discrete-time situation, these prefilters are based on estimates of the model polynomials and both they, and the polynomials of the continuous-time auxiliary model, are updated iteratively based on the iterative estimates of parameter vectors $\boldsymbol{\theta}_{ab}$ and $\boldsymbol{\theta}_{dc}$.

The SRIVC algorithm was first suggested and implemented in

[5] Note that the inputs to these continuous-time prefilters are interpolated, normally by a first order hold operation. Optimal interpolation is possible but these simpler options seem to be sufficient.

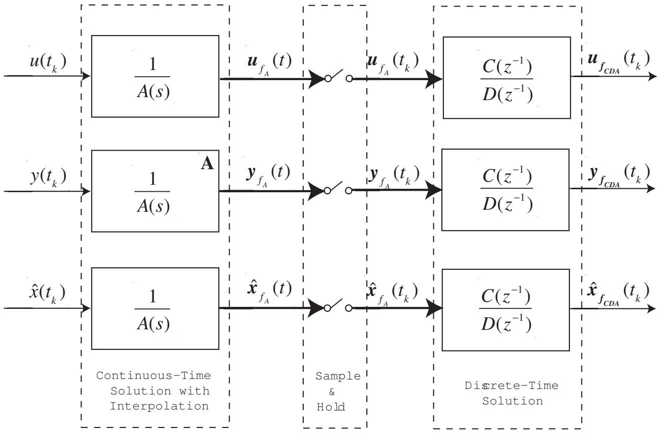

Fig. 7. The hybrid analog-digital prefiltering operations used in RIVC estimation : the bold arrows and bold-face letters denote vector quantities with elements defined as the appropriate prefiltered derivatives, with the subscripts denoting the associated prefiltering operations. The inside of block $\mathbf{A}$ in this Figure is shown in Figure 6 (based on Figure 2(a) in [44]).

1980 [44], while the full RIVC algorithm has been implemented recently [41] [42]. Since the algorithms are so similar to their discrete-time counterparts, they will not be discussed in detail here but simply summarized below. The interested reader should consult the cited references to find a more detailed description.

The SRIV Algorithm (additive white noise)

Step 1. Specify $\hat{A}_0(s)$ based on either converting the SRIV estimated discrete-time model to continuous-time ; or by using a simple, single parameter 'state variable filter' (SVF : see comment below). Prefilter the input and output variables by $1/\hat{A}_0(s)$ in order to generate the required prefiltered derivative signals (Figure 6) and then estimate the TF system model parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ by the *en-bloc* or recursive IV algorithm to yield initial estimates $\hat{A}_1(s)$ and $\hat{B}_1(s)$.

Step 2. Iterative IV estimation with prefilters.
for $j = 2 : convergence$

(i) Generate the IV variable $\hat{x}(t)$ from the auxilary model :

$$\hat{x}(t) = \frac{\hat{B}_{j-1}(s)}{\hat{A}_{j-1}(s)} u(t - \tau)$$

with the polynomials based on the estimated parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ obtained at the previous iteration of the algorithm.

(ii) Prefilter the input $u(t)$, output $y(t)$ and instrumental variable $\hat{x}(t)$ signals by the continuous-time filter $1/\hat{A}_{j-1}(s)$ in order to generate the filtered derivatives of these variables that are available at the inputs to the integrators in the prefilter (Figure 6).

(iii) Sample the prefiltered derivatives from (ii) at the discrete-time sampling interval Δt and compute the estimate of the TF system model parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ based on the estimation model (38), using the *en bloc* IV algorithm or its recursive equivalent.

end

Step 3. Compute the estimated parametric error covariance matrix $\hat{\mathbf{P}}_{ab}$ associated with the parameter estimates from the equivalent expression to that in (30).

Comment The SVF approach simply requires the selection of the single breakpoint parameter λ (breakpoint frequency in radians/time unit) of the prefilter,

$$f_{svf}(s) = \frac{1}{E(s)} = \frac{1}{(s+\lambda)^n} \tag{41}$$

which is chosen so that it is equal to, or larger than, the bandwidth of the system to be identified. This filter form was suggested a long while ago (e.g. [19], [20]) but has proven popular ever since.

The RIVC Algorithm (optimal for additive ARMA noise)

Step 1. Apply the SRIVC algorithm based on an initial $\hat{A}_0(s)$, chosen as discussed previously, and generate an initial estimate of the TF model parameter vector $\hat{\boldsymbol{\theta}}_{ab}$.

Step 2. Iterative estimation.
for $j = 2 : convergence$

(i) Generate the IV variable $\hat{x}(t)$ from the auxilary model :

$$\hat{x}(t) = \frac{\hat{B}_{j-1}(s)}{\hat{A}_{j-1}(s)} u(t - \tau)$$

with the polynomials based on the estimated parameter vector $\hat{\boldsymbol{\theta}}_{ab}$ obtained at the previous iteration of the algorithm.

(ii) Generate the discrete-time series $u(t_k)$, $y(t_k)$ and $\hat{x}(t_k)$ by sampling $u(t)$, $y(t)$ and $\hat{x}(t)$ at the discrete-time sampling interval Δt. Then, as in the discrete-time case, obtain an estimate of the noise model parameter vector $\boldsymbol{\theta}_{dc}$ based on the estimated noise sequence $\hat{\xi}(t_k) = y(t_k) - \hat{x}(t_k)$, using a discrete-time AR or ARMA model estimation algorithm (e.g. IVARMA).

(iii) Prefilter the input $u(t)$, output $y(t)$ and instrumental variable $\hat{x}(t)$ signals by the continuous-time filter $1/\hat{A}_{j-1}(s)$ in order to generate the filtered derivatives of these variables, as in the SRIVC algorithm.

(iv) Sample the continuous-time prefiltered input and output derivative signals from (iii) and filter them by the discrete-time inverse noise filter $\hat{C}_j(z^{-1})/\hat{D}_j(z^{-1})$.

(v) Based on these sampled, prefiltered derivatives and the estimation model (38), estimate $\boldsymbol{\theta}_{ab}$ using the *en bloc* IV algorithm or its recursive equivalent.

end

Step 3. Compute the estimated parametric error covariance matrices $\hat{\mathbf{P}}_{ab}$ and $\hat{\mathbf{P}}_{dc}$ associated with the parameter estimates from the equivalent expressions to those in (30) and (31).

VII. Continuous-Time Examples

A. Continuous-Time Simulation Example

The RIVC algorithm has been evaluated comprehensively elsewhere (e.g. [41] [42]) and the reader should consult

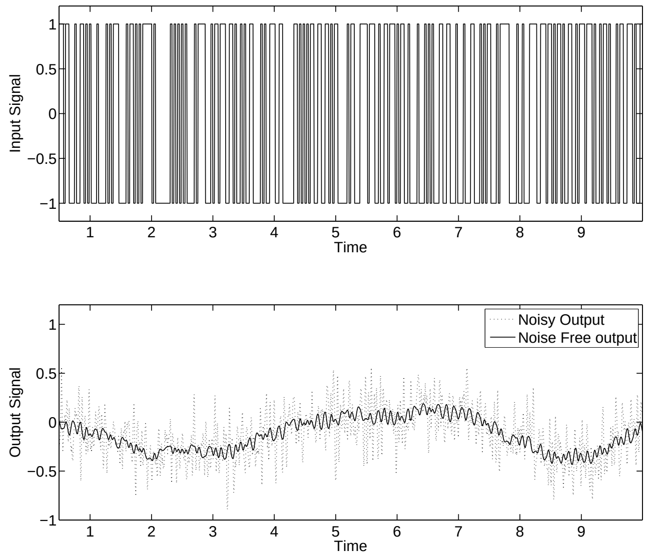

Fig. 8. A section of the modified IFAC SYSID'06 Benchmark data

these publication for details of the MCS analysis. As in the discrete-time case discussed in previous sections, these MCS results confirm the utility and statistical efficiency of the RIVC algorithm. Here, we will illustrate the performance of the algorithm by applying it to a Benchmark example that was prepared for the recent IFAC SYSID'06 Symposium in Newcastle, NSW, Australia.

Unfortunately, the associated Benchmark Session at SYSID was cancelled because referees felt that insufficient submitted papers were acceptable (only one of the papers submitted to the proposed benchmark session got even close to the correct model, demonstrating the difficulty of the benchmark exercise). The present example is a modified version of the original benchmark example, 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 :

$$x(t) = \frac{-120s^2 - 1560s + 3600}{s^4 + 30.2s^3 + 3607s^2 + 750s + 3600} u(t - 0.035)$$

$$y(t_k) = x(t_k) + \frac{1 + 0.5z^{-1}}{1 - 1.4z^{-1} + 0.7z^{-2}} e(t_k)$$
$$e(t_k) = \mathcal{N}(0, 0.0025)$$

The input signal is a 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). A section of the input-output data used in the example is plotted in Figure 8 : the noise-signal-ratio (by standard deviation) is 0.8.

As pointed out previously in the introductory Section I, a significant advantage of optimal IV estimation is that the properties of the IPM [17] facilitate the identification of model order (here via the YIC function [46] [31], based on the inverse of the IPM : see Appendix 1). The model order identification results in this case are summarised in Table IV, which clearly identifies the correct [4 3 7] order model. The subsequent single run SRIVC and RIVC estimation results are shown in Table V, where we see that both algorithms provide good estimates of the parameters. The main difference is that the SRIVC[6] provides rather optimistic estimates of the standard error (SE) on the parameters ; while RIVC provides more realistic estimates of this uncertainty. Figure 9 shows that the Bode plots of the RIVC estimated model (full line) are hardly distinguishable from those of the true model (dash-dot line). The SRIVC estimated model produced a very similar Bode plot. Indirect estimation using the discrete-time RIV, PEM and IV4 algorithms, followed by conversion to continuous-time using the Matlab D2CM function, failed at this fast sampling rate because the algorithms did not converge on acceptable discrete-time models.

Model	YIC	R_T^2
4 3 7	-4.3443	0.601817
4 3 6	-4.3324	0.601295
4 3 8	-4.3104	0.601671
4 2 7	-3.8679	0.588101
4 2 8	-3.7863	0.587521
3 3 8	-3.4524	0.594352
4 4 6	-2.3979	0.601675
4 5 6	-2.1836	0.602147
5 3 6	-1.6287	0.602004
5 7 6	-1.3330	0.602221
4 4 7	0.2252	0.601837

TABLE IV
TF MODEL IDENTIFICATION

It is worth noting that the RIVC algorithm has a much longer computation time than the SRIVC algorithm. As a result, it is advantageous to use the SRIVC algorithm for initial model order identification and only employ the full RIVC algorithm in those situations where the theoretical assumptions are satisfied and it is essential to have the most efficient parameter estimates and better estimates of the uncertainty on the parameters. For day-to-day usage, the SRIVC algorithm provides a quick and reliable approach to continuous-time model identification and estimation and has been used for many years as the algorithm of choice for this in the CAPTAIN Toolbox.

B. *Continuous-Time Practical Example*

The *Global Circulation Models* (GCMs) used to study the possibility of global warming are amongst the largest computer simulation models ever constructed. In this example, we use the RIVC algorithm to identify and estimate a continuous-time model between radiative forcing $u(t)$ and the global mean temperature perturbation $y(t)$ obtained from a standard forcing experiment performed on the U.K. HadCM3 A-OGCM GCM model, as shown in Figure 10.

[6] The ARMA(2, 1) noise model presented in this case was estimated separately by the IVARMA algorithm based on the estimated model residuals.

Method	Value	$\hat{a}_1$	$\hat{a}_2$	$\hat{a}_3$	$\hat{a}_4$	$\hat{b}_0$	$\hat{b}_1$	$\hat{b}_2$	$\hat{c}_1$	$\hat{c}_2$	$\hat{d}_1$	$\hat{\sigma}^2$
	True	30.2	3607	750	3600	-120	-1560	3600	-1.4	0.7	0.5	0.0025
RIVC	$\hat{\boldsymbol{\theta}}$	31.2	3622	799	3645	-131.9	-1608	3650	-1.399	0.698	0.483	0.0025
	SE	5.9	224	59	229	19	132	241	0.01	0.01	0.013	
SRIVC	$\hat{\boldsymbol{\theta}}$	34.4	3710	820	3732	-145.5	-1644	3729	-1.399	0.698	0.482	0.0025
	SE	3.82	141	39	145	12.7	89	155	0.01	0.01	0.013	

TABLE V
BENCHMARK EXAMPLE ESTIMATION RESULTS (NOMENCLATURE AS FOR EARLIER TABLES)

Model	Details	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_0$	$\hat{b}_1$	R_T^2	MCS response
RIVC	$\hat{\boldsymbol{\theta}}$	0.15206	0.00037039	0.33328	0.0014026	0.99547	Good
	SE	0.015185	4.8019e-05	0.031479	0.00017601		
RIVC	mean (MCS)	0.15147	0.00036899	0.33199	0.0013972	0.99547	Good
	SD (MCS)	0.015941	4.9796e-05	0.033128	0.0001829		
RIV (d2cm)	mean (MCS)	0.14304	0.00035324	0.31011	0.0022757	0.99546	Poor
	SD (MCS)	0.027994	0.00015783	0.02799	0.035868		
PEM (d2cm)	mean (MCS)	0.14877	0.00036609	0.32469	0.0013846	0.99546	Unacceptable
	SD (MCS)	0.095537	0.00030185	0.19801	0.0011053		

TABLE VI
GCM MODEL REDUCTION ESTIMATION RESULTS (NOMENCLATURE AS FOR EARLIER TABLES)

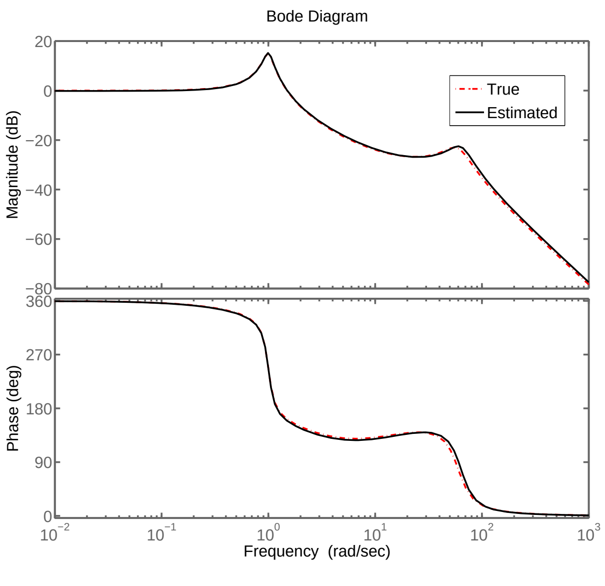

Fig. 9. Comparison of true and RIVC estimated Bode diagrams

Identification using the YIC and R_T^2 statistics suggest a second order [2 2 0] model with an associated AR(3) noise model. The RIVC estimated model then takes the following form :

$$x(t) = \frac{0.3333s + 0.001403}{s^2 + 0.1521s + 0.00037}u(t)$$

$$y(t_k) = x(t_k) + \frac{1}{1 - 0.492z^{-1} + 0.135z^{-2} - 0.158z^{-3}}e(t_k)$$

$$e(t_k) = \mathcal{N}(0, 0.017) \qquad \Delta t = 1\ year$$

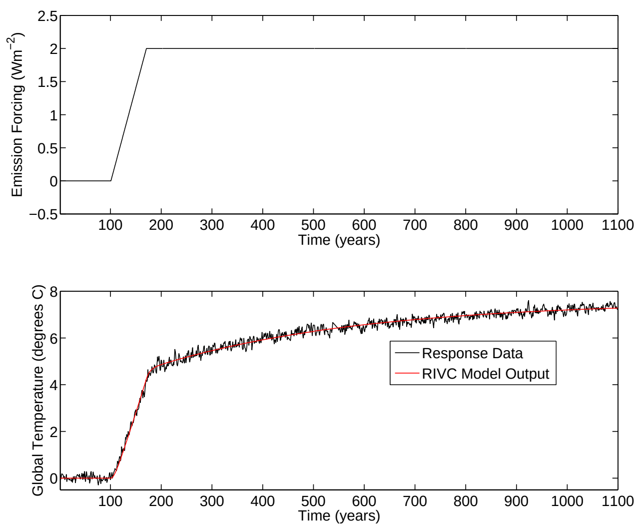

Fig. 10. Hadley GCM data and RIVC estimated model output

It is interesting to note that this model can be decomposed straightforwardly into a feedback connection of two first order processes. This is easily interpreted in energy balance terms as a feedback process with quick (6.8 years, forward path) and very slow (238 years, feedback path) modes, an interpretation that is immediately useful in scientific terms and can be used to compare the nature of the reduced order model with that of the very high order GCM from which it has been derived.

Table VI compares the RIVC estimation results with those obtained by 1000 realization MCS analysis using a simulation model based on the RIVC parameter and covariance matrix estimates. We see that the single run and MCS derived results are very similar, again demonstrating the efficacy of the RIVC algorithm. The Table also compares the RIVC estimates with the estimation results obtained by indirect estimation using the RIV and PEM algorithms, with conversion using the D2CM algorithm in Matlab. At first sight, these indirect results look reasonable. However, the final column in the Table reports the nature of the results obtained in the MCS analysis : here we see that the indirect estimates produce poor (RIV) and unacceptable (PEM) results, in the sense that the ensemble of model output responses had a very high variance, much greater than that obtained using the direct RIVC estimation results. This is a consequence of the sensitivity, in this example, of the discrete-continuous conversion D2CM to the estimated uncertainty defined by the RIV and PEM estimated parametric covariance matrices.

VIII. Nonstationary and Nonlinear Model Estimation

As pointed out previously, the RIV and RIVC algorithms are easily formulated in recursive terms. As a result, it is straightforward to use them for the estimation of time variable parameters in nonstationary dynamic systems. The discrete-time DTF algorithm in the CAPTAIN Toolbox, for instance, allows for such *Time variable Parameter* (TVP) estimation in the case where the temporal variation of the parameters can be modelled by random walk or integrated random walk stochastic processes, with the associated hyper-parameters (noise variance ratios) optimized by maximum likelihood using prediction error decomposition [33] [37]. It is also possible to extend the methods to handle the estimation transfer function models with a *State-Dependent Parameter* (SDP) form, so allowing for the identification and estimation of a wide class of nonlinear stochastic systems.

SDP estimation can be in discrete or continuous time : the simplest SDP continuous-time model is a nonlinear equivalent of the linear TF model and takes the following form :

$$y(t) = \frac{b_0(z_{n+1,t})s^m + \cdots + b_m(z_{n+m+1,t})}{s^n + a_1(z_{1,t})s^{n-1} + \cdots + a_n(z_{n,t})} u(t-\tau) + e(t)$$

where $\mathbf{z}_t = [z_{1,t} \cdots z_{n+m+1,t}]^T$ is a vector of measured variables (states) on which the parameters may be dependent. In estimation equation terms, this model can be written most conveniently as :

$$s^n y(t) = \boldsymbol{\phi}_t^T \mathbf{p}_t + e_t$$

where,

$$\begin{aligned} \boldsymbol{\phi}_t^T &= \left[-s^{n-1}y(t) \ldots -y(t)\; s^m u(t-\tau) \ldots u(t-\tau)\right] \\ \mathbf{p}_t &= [p_1(z_{1,t}) \ldots p_{n+m+1}(z_{n+m+1,t})]^T \end{aligned}$$

while,

$$\begin{aligned} &p_1(z_{1,t}) = a_1(z_{1,t}); p_2(z_{2,t}) = a_2(z_{2,t}); \\ &\qquad \cdots;\; p_{n+m+1}(z_{n+m+1,t}) = b_m(z_{n+m+1,t}) \end{aligned}$$

SDP estimation algorithm is based on this form of the SDP estimation model and is normally carried out in two stages : (i) initial non-parametric identification of the nonlinear model structure and form using a special form of recursive fixed interval smoothing estimation that results in a graphical estimate of the SDP plotted against the state on which it is dependent (see the example in the next section) ; (ii) parameterization of the identified state-dependencies (nonlinearitities) based on the non-parametric form of the SDP identified in (i), followed by statistically optimal estimation of the model in this parameterized form. Further details of this SDP approach to nonlinear modelling are available in the cited references.

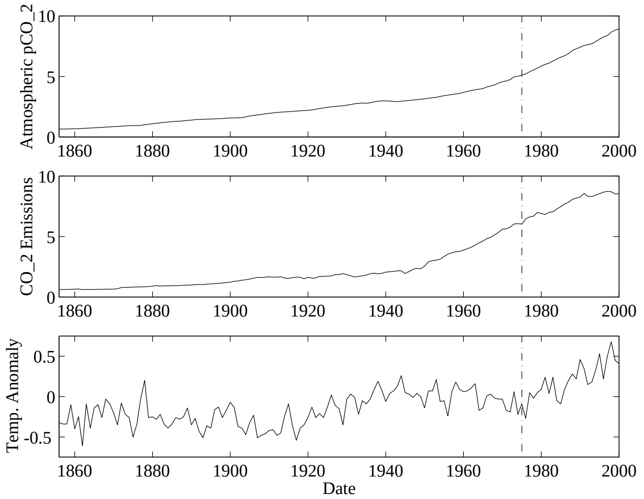

Fig. 11. Annual carbon dioxide and temperature anomaly data 1856-2000.

A. Continuous-Time SDP Practical Example

This example is taken from a recent study of the global carbon cycle, the full details of which are given in other publications [47] [40] [38]. The modelling aspects of this study are concerned with investigating the dynamic relationship, over the period 1856 to 2000, between the time series data shown in Figure 11. These are : in the top panel, globally averaged annual measures of CO_2 emissions (arising from both the use of carbon fuels and land-use changes) ; in the middle panel, perturbations in globally averaged atmospheric carbon dioxide partial pressure, pCO_2 ; and, in the lower panel, the Northern Hemisphere temperature anomaly (i.e. the changes in Northern Hemisphere averaged temperature about its long term average).

As shown in Figure 12, initial non-parametric SDP estimation suggests the possible presence of a temperature-dependent nonlinearity in the dynamic relationship between CO_2 emissions and atmospheric pCO_2. In this figure, the non-parametric SDP estimates (dash-dot lines) are compared with the estimates of two possible parameterized functions (full lines) : a linear relationship in temperature in the left hand panel and a sigmoidal relationship in temperature in the right hand panel. The finally estimated model using the latter parameterization takes the form :

$$\begin{cases} x(t) = \frac{b_0}{s+a_1(T(t))}\{u(t-\tau)+c\} \\ y(t) = x(t) + \xi(t) \end{cases}$$

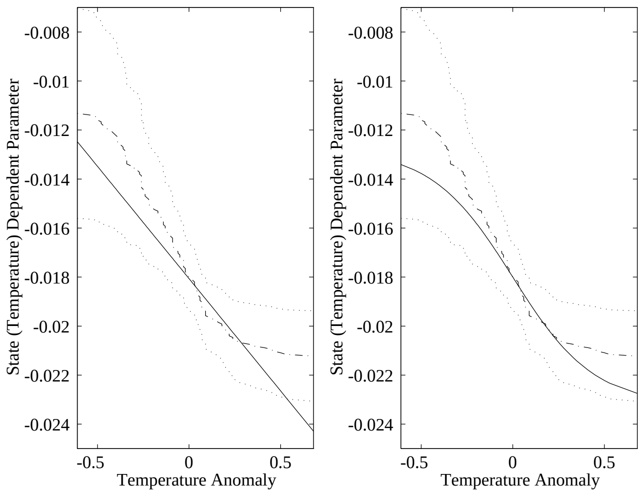

Fig. 12. Initial non-parametric and final parametric estimates of the SDP nonlinearity in the estimated climate data model.

where the SDP $a_1(T(t)) = \{\alpha + \frac{1}{1+e^{-\beta T(t)}}(\gamma - \alpha)\}$; or,

$$\begin{cases} \frac{dx(t)}{dt} = \{\alpha + \frac{1}{1+e^{-\beta T(t)}}(\gamma - \alpha)\}x(t) + b_0 u(t-\tau) + c \\ y(t) = x(t) + \xi(t). \end{cases} \tag{42}$$

and the associated parameter estimates based on the whole data set (up to year 2000), are as follows[7]

$$\hat{\alpha} = -0.0232(0.0062); \hat{\beta} = -4.503(7.321);$$
$$\hat{\gamma} = -0.0128(0.0054); \hat{b}_0 = 0.0402(0.0017);$$
$$\hat{c} = 0.0066(0.0015); \hat{\tau} = 5.0;$$
$$\sigma_\xi^2 = 0.00240; R_T^2 = 0.99949$$

Figure 13 shows the results on MCS-based predictive validation of the model (42). Here, the model was estimated on the basis of the data up to 1975 and then this estimated model, together with the associated estimate of the parametric covariance matrix, were used to generate an MCS ensemble of model outputs from the start of the data to the year 2000. The 95 percentile boundaries of this ensemble are shown in the Figure and we see that the uncertainty in the prediction is very low, showing how well the model is identified and estimated. In particular, the model predicts the atmospheric pCO_2 very well indeed over the validation period (the last 25 years of the twentieth Century), the data for which were not used in the model estimation.

Finally, it is worth mentioning that the following SRIVC estimated linear first order model is also able explain the data of Figure 11 very well, almost as well as the nonlinear model (42), but it is characterized by more parameters. This model takes the form :

$$\begin{cases} x(t) = \frac{b_{10}+b_{11}s}{s+a_1}u(t-\tau) + \frac{b_{20}+b_{21}s}{s+a_1}c \\ y(t) = x(t) + \xi(t) \end{cases}$$

[7] These results were obtained with a fixed $\tau = 5$: however, this was based on prior estimation with τ allowed to take on non integral values.

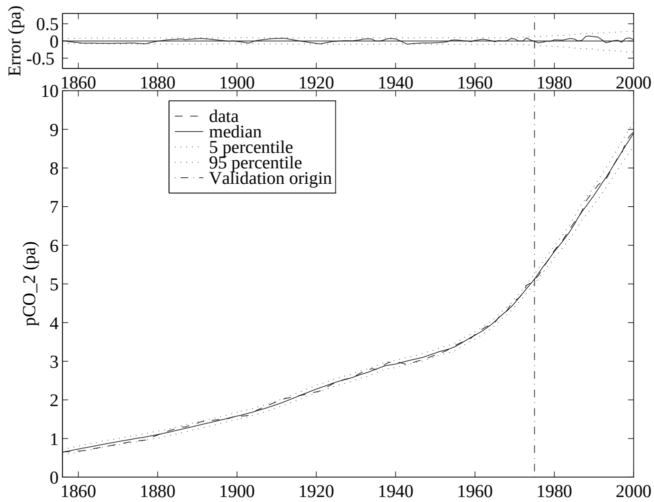

Fig. 13. Monte Carlo simulation and predictive validation results for the estimated climate data model.

where c in a constant input of 1.0 $\forall t$. The associated parameter estimates are as follows :

$$\hat{a}_1 = -0.02226(0.0010); \hat{b}_{10} = 0.0748(0.0152);$$
$$\hat{b}_{11} = 0.0407(0.0011); \hat{b}_{21} = 0.6199(0.0190);$$
$$\hat{b}_{22} = 0.01775(0.0009); \hat{\tau} = 7.0;$$
$$\sigma_\xi^2 = 0.00257; R_T^2 = 0.99945;$$

Whilst this represents quite a good model, it is rejected in favour of the nonlinear model for three reasons : first, it requires more parameters with no associated improvement in the explanation of the data ; second, unlike the nonlinear model, its residual noise series is significantly correlated with the temperature anomaly series ; and third, it does not perform as well in predictive validation.

IX. Conclusions

As far as the author is aware, the RIV/SRIV and RIVC/SRIVC algorithms constitute the only unified, time domain, family of algorithms that provide statistically optimal solutions to the estimation of both discrete-time and 'hybrid' continuous-time TF Models of the Box-Jenkins type. Moreover, because of their IV heritage, the simpler SRIV/SRIVC algorithms can be applied in circumstances where the noise process does not conform to the rational spectral density assumptions. Consequently, they have proven to very robust, providing excellent, computationally efficient, 'day-to-day' approaches to TF model identification and estimation that have been applied successfully to a wide range of practical applications, in areas ranging from ecology, through engineering, to economics. Also, the RIV/RIVC algorithms can be extended easily for use in a feedback situation provided an 'external' input is being applied (e.g. [3]).

The main advantage of the RIV/RIVC estimation algorithms, in relation to their simpler SRIV/SRIVC versions, is that they allow for improved statistical efficiency, with better estimation of the standard error bounds on the parameter estimates. The main disadvantages are : (i) they depend on concurrent noise model estimation and so have a longer computation time ; and (ii), in the case of RIVC based on rapidly sampled data, estimation will be less robust when the roots of the noise model denominator polynomial $C(z^{-1})$ approach the unit circle.

At high sampling frequencies, direct SRIVC/RIVC identification is much superior to the alternative of indirect discrete-time estimation (with associated continuous-time model conversion). Also, physical interpretation of the TF models is facilitated by their differential equation form and the uniqueness of the continuous-time model parameters, which are not a function of the sampling interval Δt.

The asymptotic independence of the system and noise model parameter estimates is a most attractive feature of the Box-Jenkins family of models that form the basis for RIV/RIVC estimation. This, together with the iterative, relaxation form of the parameter estimation, seem to engender some advantages for these algorithms in relation to alternative gradient optimization algorithms, such as PEM and standard ML.

In case there is any confusion, it must be emphasized that the discrete-time RIV algorithm described in this paper is quite different to the IV4 algorithm available in the Matlab SID Toolbox. Although they have some elements in common, the IV4 is a four *step* algorithm which, unlike RIV, does not involve any iteration. IV4 is also formulated on the basis of an ARX model form, under the assumption of an AR(n+m) noise model[8], rather than the BJ model form used by RIV, and so it does not exploit the advantages of the BJ model mentioned above. Most importantly in practical terms, the RIV algorithm seems more reliable than IV4, which can exhibit rather poor performance. This may have contributed deleteriously to the reputation of optimal IV methods and it is hoped that the present paper will encourage readers to utilize the RIV/SRIV algorithms and so restore this reputation.

Finally, it should be noted that the whole family of RIV/RIVC algorithms, as well as the related time variable parameter (DARX, DTF) and state-dependent parameter (SDP) algorithms, are all available in the CAPTAIN Toolbox for Matlab that can be downloaded from http ://www.es.lancs.ac.uk/cres/captain/. The continuous-time RIVC algorithm, together with almost all other continuous-time TF estimation algorithms, are also available in the CONTSID Toolbox that can be downloaded from http ://www.cran.uhp-nancy.fr/contsid/.

X. Acknowledgements

The author is very grateful to Professor Hugues Garnier for his help in preparing this paper : his careful reading of initial drafts and his insightful comments on the content have been extremely useful. Of course, the author remains responsible for any errors or omissions in this final version.

Appendix 1 : Model Identification Statistics

Two main model identification statistics used in this paper are the *Coefficient of Determination* R_T^2 ; and the YIC model structure identification statistic. The well-known *Akaike Identification Criterion* (AIC : see [1]) is also utilized but only for AR noise model identification.

[8] although it does not return details of this noise model

The R_T^2 is defined as :

$$R_T^2 = 1 - \frac{\sigma_{\hat{\xi}}^2}{\sigma_y^2};\ \ \sigma_{\hat{\xi}}^2 = \frac{1}{N}\sum_{k=1}^{N}\hat{\xi}_k^2;\ \ \sigma_y^2 = \frac{1}{N}\sum_{k=1}^{N} y_k^2 \quad (43)$$

where $\hat{\xi}_t$ is the error $y_k - \hat{x}_k$ between the measured output y_k and simulated model output $\hat{x}_k$ from equation (28). This is often a more discerning measure of model adequacy than the *Coefficient of Determination* R^2, which is normally defined in terms of the one-step-ahead prediction errors.

The YIC [31] is a heuristically defined identification criterion derived from the earlier EVN criterion [46] and exploits the special properties of the Instrumental Product Matrix (IPM). It is defined as follows :

$$YIC = \log_e\left\{\frac{\hat{\sigma}^2}{\sigma_y^2}\right\} + \log_e NVN; \quad (44)$$

$$NVN = \frac{1}{np}\sum_{i=1}^{np}\frac{\hat{\sigma}^2 p_{ii}}{\hat{\theta}_i^2} \quad (45)$$

where $\hat{\sigma}^2$ is the variance of the model residuals ; np is the total number of model parameters (for the full BJ model, $np = n + m + p + q + 1$) ; p_{ii} are the np diagonal elements of the IPM inverse generated by the RIV algorithm ; and $\hat{\theta}_i, i = 1, 2 \cdots np$ are the RIV parameter estimates.

The first term in (44) is simply a relative measure of how well the model explains the data : the smaller the model residuals the more negative this logarithmic term becomes. The second logarthmic term provides a normalized measure that reflects the conditioning of the IPM, which is inverted when the IV estimation equations are solved. If the model is over-parameterized, then it can be shown (e.g. [17]) that the IPM will tend to singularity and, because of its ill-conditioning, the elements of its inverse (on which the estimate of the parametric covariance matrix depends) will increase in value, often by several orders of magnitude. When this happens, the second term in the YIC tends to dominate the criterion function, indicating over-parameterization. Another justification of the YIC can be obtained from statistical considerations [31].

Appendix 2 : The IVARMA Algorithm

The *AutoRegressive Moving Average* (ARMA) model for a stochastic time series ξ_k with rational spectral density has the well-known form[9] :

$$\xi_k = \frac{D(z^{-1})}{C(z^{-1})}e_k \quad (46)$$

The transfer function defining this model can be written as,

$$\frac{D(z^{-1})}{C(z^{-1})} = 1/\frac{C(z^{-1})}{D(z^{-1})} = \frac{1}{G(z^{-1})} \quad (47)$$

where, in general, $G(z^{-1})$ is an infinite dimensional polynomial in z^{-1}. Consequently, it is obvious that, for sufficiently

[9] This Appendix is an expanded and slightly corrected version of the analysis in [39], which had two lines omitted in error

high h, this ARMA model can be approximated by a high order AR(h) model of the form,

$$\xi_k = \frac{1}{G(z^{-1})} e_k \tag{48}$$

where,

$$\begin{gathered} G(z^{-1}) = 1 + g_1 z^{-1} + ... + g_h z^{-h} \\ h \gg p \end{gathered} \tag{49}$$

and that, in this case, an estimate $\hat{e}_k$ of e_k can be obtained from,

$$\hat{e}_k = \xi_k + g_1 \xi_{k-1} + ... + g_h \xi_{k-h} \tag{50}$$

where the degree of approximation will depend primarily upon the nature of the MA polynomial and the order h.

If it is assumed that the order h of $G(z^{-1})$ in (48) has been selected such that $\hat{e}_k$ is a good approximation to e_k, then we can consider that it provides an observation of e_k, so that,

$$\hat{e}_k = e_k + \epsilon_k \tag{51}$$

where ϵ_k represents the approximation error and will almost always have a very small variance in comparison with the variance of e_k. As a result, a linear regression equation model for $\hat{e}_k$ can be written in the form :

$$\hat{e}_k = \tilde{\boldsymbol{\psi}}_k^T \boldsymbol{\theta}_{dc} + \epsilon_k \tag{52}$$

where[10],

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_k^T &= [-\hat{e}_{k-1}, ..., -\hat{e}_{k-q}\ \xi_k\ \xi_{k-1}, ..., \xi_{k-p}] \\ \boldsymbol{\theta}_{dc} &= [d_1, ..., d_q\ 1\ c_1, ..., c_p]^T \end{aligned} \tag{53}$$

As pointed out in the main text, a suitable error function ε_k^n in this case is defined as follows :

$$\varepsilon_k^n = \frac{C(z^{-1})}{D(z^{-1})} \xi_k - \hat{e}_k \tag{54}$$

Now, by introducing the prefilter $f_2(z^{-1})$ defined as,

$$f_2(z^{-1}) \triangleq \frac{1}{D(z^{-1})} \tag{55}$$

the error function can be written in the form,

$$\varepsilon_k^n = -D(z^{-1}) e_k^{f_2} + C(z^{-1}) \xi_k^{f_2} \tag{56}$$

An associated estimation model can then be written in the pseudo-linear regression form :

$$\text{Noise Estimation Equation :} \quad \hat{e}_k^{f_2} = \boldsymbol{\psi}_k^T \boldsymbol{\theta}_{dc} + \epsilon_k \tag{57}$$

where,

$$\boldsymbol{\psi}_k^T \quad = \quad [-\hat{e}_{k-1}^{f_2} \cdots - \hat{e}_{k-q}^{f_2}\ \xi_k^{f_2}\ \xi_{k-1}^{f_2} \cdots\ \xi_{k-p}^{f_2}] \tag{58}$$

Note that the unity element is retained in the definition of $\boldsymbol{\theta}_{dc}$ so that the format is consistent with its use in TF algorithms from the CAPTAIN Toolbox, which allow for the imposition of such a constraint on the estimation.

Equation (57) provides a basis for the estimation of the ARMA parameters in the parameter vector $\boldsymbol{\theta}_{dc}$. If the error series ϵ_k could be assumed to have zero mean, serially uncorrelated characteristics, then constrained linear least squares (LS) estimation could be used to solve this estimation problem, with the constraint applying to the unity element in $\boldsymbol{\theta}_{dc}$. However, it is more robust in these circumstances to utilize a constrained implementation of the SRIV-type algorithm discussed in the main text and available in the CAPTAIN Toolbox. This can be applied directly to the basic vector form of the noise model (52) because the prefiltering operation required for the estimation model (57) is inherent in the SRIV algorithm (for this noise model the prefilter is, of course, $1/D(z^{-1})$ rather than $1/A(z^{-1})$ for the TF system estimation model (9) for which the SRIV algorithm is primarily intended).

This SRIV-based approach will be more robust to the statistical nature of ϵ_k. If the algorithm converges and ϵ_k is a zero mean, serially uncorrelated, Gaussian normal sequence of random variables, then the resulting SRIV parameter estimates will be consistent and statistically efficient ; if not, then the estimates will be consistent and asymptotically unbiased [29]. In any case, as pointed out previously, ϵ_k will almost always have a very small variance in comparison with the variance of e_k, so it will not have much effect on the estimation.

Finally, it is obvious from the definition of the pseudo-linear regression form of the estimation model (57) (and confirmed by the Pierce Theorem in Section III) that an estimate of the covariance matrix $\mathbf{P}_{dc}$ associated with the SRIV parameter estimates can be obtained as (see equations (31)) :

$$\hat{\mathbf{P}}_{dc} = \hat{\sigma}^2 \left[\sum_{k=1}^{N} \hat{\boldsymbol{\psi}}_k \hat{\boldsymbol{\psi}}_k \right]^{-1} \tag{59}$$

where $\hat{\sigma}^2$ is the estimate of the variance of the white noise input e_k, as obtained from the noise model estimation residuals in the usual manner.

Following from the above discussion, the main steps in the iterative IVARMA algorithm are as follows :

The IVARMA Algorithm

Step 1. If a simple AR model is requested, estimate it in the normal LS manner ; otherwise estimate a high order AR model with the order based on the user specification or some automatic method of order specification. The method is not sensitive to the order selection and a default order of 50 is usually sufficient.

Step 2. Generate the residual error series $\hat{e}_k$ associated with the high order AR model and estimate the parameter vector $\boldsymbol{\theta}_{dc}$ of the model (52) with the unity element constrained to this value using the SRIV option of the RIV algorithm in CAPTAIN. For this purpose, the input is specified as ξ_k and the output as $\hat{e}_k$. The SRIV algorithm yields the estimate $\hat{\boldsymbol{\theta}}_{dc}$ and the covariance matrix $\mathbf{P}_{dc}$ is computed from equation (59).

Note that the iterative nature of the solution in the algorithm, as described above, is contained within the SRIV algorithm.

[10] Note that when IVARMA is used in the RIVC algorithm, the ξ_k noise signal here is replaced by the estimated noise signal $\hat{\xi}_k$ obtained from equation (18).

A. A Simulation Example

This example is concerned with the analysis of the time series ξ_k generated by the ARMA model :

$$\xi_k = \frac{1 - 0.9z^{-1}}{1 - 0.1z^{-1} - 0.6375z^{-2}} e_k \quad e_k = N(0,1) \qquad (60)$$

where the denominator polynomial factorizes to the form,

$$C(z^{-1}) = (1 - 0.85z^{-1})(1 + 0.75z^{-1})$$

Table VII compares the results of single run and MCS analysis for the IVARMA and PEM algorithms (using its ARMA estimation option). The first thing to note is the last column of the Table, which reports some failures to converge on an acceptable model : the PEM algorithm fails in 7 of the 100 realizations, while the RIV algorithm fails twice (these realizations were removed in computing the statistics shown in the Table). However, other than in these two realizations, the IVARMA performance is good, with the single run predicted standard errors on the parameter estimates matching well the standard deviations computed from the MCS analysis. The PEM performance is not so good, with relatively poor agreement between the single run standard errors and the MCS estimated standard deviations ; and these standard deviations are between between 20% and 50% greater than those achieved by IVARMA. This is also illustrated in Figure 14, which shows the Bode gain diagrams plotted from the MCS ensemble of models, with the IVARMA results in the left panel and the PEM ones in the right panel. Here, the true model is shown as dots, the mean of the realizations as a full line and the standard deviation (SD) bounds as dashed lines. We see that the main difference between the results is that IVARMA is more successful than PEM at estimating the lower frequency components of the noise signal (i.e. those associated with the eigenvalue at 0.85). It is clear from these results that, in this particular example, IVARMA is performing rather better than the PEM, even if the realizations that failed to converge properly are omitted. In other simulation examples (e.g. [39]), however, the results obtained from the two algorithms are very similar. So, as in the case of the RIV algorithm considered in the main body of this paper, it would appear that IVARMA can perform better than PEM in certain situations. Once again, this seems to be related to both the nature of the system (here the ARMA noise model) and the initiation of the gradient optimization used for PEM model estimation. However, it should be noted that the PEM algorithm is computationally more efficient than IVARMA : in this example, for instance, it takes 0.2 seconds, while IVARMA takes 0.85 seconds.

Method	$\hat{a}_1$	$\hat{a}_2$	$\hat{b}_0$	Failures
True Values	-0.1	-0.6375	-0.9	
IVARMA (SR)	-0.0911	-0.6479	-0.9629	
SE	0.032	0.031	0.017	
IVARMA (MCS)	-0.0987	-0.6344	-0.9489	2
SE	0.030	0.036	0.017	
PEM (SR)	-0.1014	-0.7051	-0.9853	
SE	0.027	0.027	0.009	
PEM (MCS)	-0.1046	-0.6371	-0.9523	7
SE	0.038	0.043	0.025	

TABLE VII
MONTE CARLO SIMULATION RESULTS FOR APPENDIX EXAMPLE : SR DENOTES SINGLE RUN RESULTS AND MCS DENOTES THE RESULTS FROM MCS ANALYSIS BASED ON 100 RANDOM REALIZATIONS.

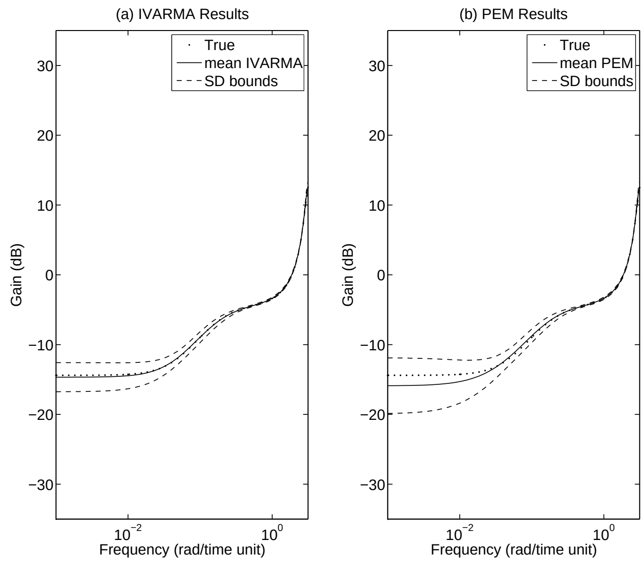

Fig. 14. Comparison of the MCS ensembles of the Bode gain plots for the ARMA$(2,1)$ models estimated by the IVARMA (left panel) and PEM (right panel) algorithms.

RÉFÉRENCES

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

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

[3] M. Gilson, H. Garnier, P. C. Young, and Paul M.J. Van den Hof. A refined IV method for closed-loop system identification. In *Proceedings 14th IFAC Symposium on System Identification SYSID06, Newcastle, NSW, Australia*, pages 903–908, Newcastle, NSW, 2006.

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

[5] 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.

[6] 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.

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

[8] L. Ljung. *System Identification. Theory for the User*. Second edition 1999. Prentice Hall, 1987.

[9] L. Ljung and T. Soderstrom. *Theory and Practice of Recursive Estimation*. MIT Press : Cambridge, Mass., 1983.

[10] D. Q. Mayne. A method for estimating discrete-time transfer functions. In *Advances in Computer Control, Second UKAC Control Convention*, University of Bristol, 1967.

[11] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser. Dual state-parameter estimation of hydrological models using ensemble Kalman filter. *Advances in Water Resources*, 28 :135–147, 2005.

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

[13] R. J. Romanowicz, P. C. Young, and K. J. Beven. Data assimilation and adaptive forecasting of water levels in the River Severn catchment. *Water Resources Research*, 42(W06407) :doi :10.1029/2005WR004373, 2006.

[14] V. Solo. *Time Series Recursions and Stochastic Approximation*. PhD thesis, Australian National University, Canberra, Australia, 1978.

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

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

[17] P. E. Wellstead. An instrumental product moment test for model order estimation. *Automatica*, 14 :89–91, 1978.

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

[19] P. C. Young. In flight dynamic checkout - a discussion. *IEEE Transactions on Aerospace*, AS2(3) :1106–1111, 1964.

[20] P. C. Young. The determination of the parameters of a dynamic process. *Radio and Electronic Eng. (Jnl. Of IERE)*, 29 :345–361, 1965.

[21] P. C. Young. On a weighted steepest descent method of process parameter estimation. Report, Department of Engineering, Cambridge University, 1965.

[22] P. C. Young. Process parameter estimation and self adaptive control. In P. H. Hammond, editor, *Theory of Self Adaptive Control Systems*, pages 118–140. Plenum Press : New York, 1965.

[23] P. C. Young. Regression analysis and process parameter estimation : a cautionary message. *Simulation*, 10 :125–128, 1968.

[24] P. C. Young. *The Differential Equation-Error Method of Real-Time Process Identification*. PhD thesis, Department of Engineering, Cambridge University, 1969.

[25] P. C. Young. The use of linear regression and related procedures for the identification of dynamic processes. In *Proc. 7th IEEE Symposium on Adaptive Processes*, New York, 1969.

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

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

[28] P. C. Young. *Recursive Estimation and Time-Series Analysis*. Springer-Verlag, Berlin, 1984.

[29] P. C. Young. The instrumental variable method : a practical approach to identification and system parameter estimation. In H. A. Barker and P. C. Young, editors, *Identification and System Parameter Estimation : Vols. 1 & 2*, pages 1–15. Pergamon Press : Oxford, 1985.

[30] P. C. Young. Recursive identification, estimation and control. In E. J. Hannan, P. R. Krishnaiah, and M. M. Rao, editors, *Handbook of Statistics : Time Series in the Time Domain*, volume 5, pages 213–255. North Holland : Amsterdam, 1985.

[31] P. C. Young. Recursive estimation, forecasting and adaptive control. In C. T. Leondes, editor, *Control and Dynamic Systems*, pages 119–166. Academic Press : San Diego, 1989.

[32] P. C. Young. Data-based mechanistic modeling of environmental, ecological, economic and engineering systems. *Journal of Environmental Modelling and Software*, 13 :105–122, 1998.

[33] P. C. Young. Stochastic, dynamic modelling and signal processing : time variable and state dependent parameter estimation. In W. J. Fitzgerald, A. Walden, R. Smith, and P. C. Young, editors, *Nonlinear and Nonstationary Signal Processing*, pages 74–114. Cambridge University Press : Cambridge, 2000.

[34] P. C. Young. Data-based mechanistic modelling and validation of rainfall-flow processes. In M. G. Anderson and P. D. Bates, editors, *Model Validation : Perspectives in Hydrological Science*, pages 117–161. John Wiley : Chichester, 2001.

[35] P. C. Young. The identification and estimation of nonlinear stochastic systems. In A. I. Mees, editor, *Nonlinear Dynamics and Statistics*, pages 127–166. Birkhauser : Boston, 2001.

[36] P. C. Young. Advances in real-time flood forecasting. *Philosophical Trans. Royal Society, Physical and Engineering Sciences*, 360(9) :1433–1450, 2002.

[37] P. C. Young. Identification of time varying systems. In H. Unbehauen, editor, *Encyclopedia of Life Support Systems (EOLSS)*, volume 6.43 : Control Systems, Robotics and Automation. Developed under the Auspices of the UNESCO, EOLSS Publishers : Oxford, 2002.

[38] P. C. Young. The data-based mechanistic approach to the modelling, forecasting and control of environmental systems. *Annual Reviews in Control*, in press, 2006.

[39] P. C. Young. An instrumental variable approach to ARMA model identification and estimation. In *Proceedings 14th IFAC Symposium on System Identification SYSID06, Newcastle, NSW, Australia*, pages 410–415, 2006.

[40] P. C. Young and H. Garnier. Identification and estimation of continuous-time, data-based mechanistic models for environmental systems. *Environmental Modelling & Software*, 21 :1055–1072, 2006.

[41] P. C. Young, H. Garnier, and M. Gilson. An optimal instrumental variable approach for identifying hybrid Box-Jenkins models. In *Proceedings 14th IFAC Symposium on System Identification SYSID06, Newcastle, NSW, Australia*, pages 225–230, 2006.

[42] P. C. Young, H. Garnier, and M. Gilson. The refined instrumental variable method for identifying continuous-time hybrid box-jenkins models. In H. Garnier and L. Wang, editors, *Continuous-Time Model Identification From Sampled Data*. Springer-Verlag : Berlin (in preparation), 2007.

[43] 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.

[44] 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.

[45] P. C. Young, A. J. Jakeman, and R. McMurtrie. An instrumental variable method for model order identification. *Automatica*, 16 :281–294, 1980.

[46] P. C. Young, A. J. Jakeman, and R. McMurtries. An instrumental variable method for model order identification. *Automatica*, 16 :281–296, 1980.

[47] P. C. Young and A. J. Jarvis. Data-based mechanistic modelling, the global carbon cycle and global warming. Technical Report TR/177, Centre for Research on Environmental Systems and Statistics, Lancaster University, 2002.

[48] 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.

[49] Peter C Young. Transfer function identification and estimation in nonstandard situations. Centre for Research on Environmental Systems and Statistics Report No. TR/205 (38 pages), Lancaster University, March 2003.