On the Initialization of Refined Instrumental Variable Estimation Algorithms

Peter C Young

Lancaster Environment Centre, Lancaster University, UK;
Fenner School of Environment & Society, Australian National University, Canberra.
email: p.young@lancaster.ac.uk

1 Introduction

A recent paper (Schoukens et al., 2011) has pointed out how, in some circumstances, the initialization of identification algorithms for the *Box-Jenkins* (BJ) class of transfer models can be critical for good performance. One example they consider is a 6th order system that has spectral peaks at three frequencies, two of which are close together. Here, their results reveal that parametric methods of identification, such as the BJ routine in the Matlab$^{\text{TM}}$SID Toolbox, may suffer from convergence to local, sub-optimal minima in the cost-function-parameter hyperspace. The authors suggest an approach based on the use of nonparametric estimation that solves this problem satisfactorily, although with a small reduction in statistical efficiency. The present paper suggests a simple alternative parametric approach for *Refined Instrumental Variable* (RIV) identification algorithms that yield maximum likelihood estimates of the parameters in a unified class of discrete and continuous-time Box-Jenkins (BJ) models (Young, 2015).

2 The Unified Box-Jenkins Model Class

This paper is concerned with the estimation of the parameters that characterise a *Single-Input, Single-Output* (SISO) or *Multiple-Input, Single-Output* (MISO), linear, time-invariant and stable transfer function model from uniformly sampled input-output data $\{u(k),\ y(k)\},\ k = 1,2,\ \ldots,N$, where the argument k could denote the kth sample from an underlying continuous-time system if the system model is in the form of a continuous-time differential equation or its transfer function equivalent.

The stochastic SISO transfer function models first conceived and promoted by Box and Jenkins (Box and Jenkins, 1970; Box et al., 1994) for discrete-time systems, can be unified (see Young, 2015) and written, at any sampling instant k, in the following decomposed form[1]:

$$
\begin{aligned}
\text{System TF Model :}\quad & x(k) = \frac{B(\mu^{-1})}{A(\mu^{-1})}u(k-\tau) \\
\text{Noise TF Model :}\quad & \xi(k) = \frac{D(\mu^{-1})}{C(\mu^{-1})}e(k); \quad e(k) = \mathcal{N}(0,\sigma^2) \\
\text{Output Observation :}\quad & y(k) = x(k) + \xi(k)
\end{aligned} \tag{1}
$$

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

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

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

Although these definitions are required for the development of the unified results and apply directly to the polynomials of the backward shift operator model, where $\mu^{-1} = z^{-1}$, the polynomials used in the development of the alternative continuous-time and δ operator algorithms (see Young, 2015), are defined in terms of μ, i.e.,

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

which does not, of course, change the model. The structure of the above model is denoted by the pentad $[n\ m\ \tau\ p\ q]$ or $[n\ m\ \tau]$ if the additive noise $\xi(k)$ is assumed to be white.

For simplicity in this paper, the simulation example is restricted to the standard discrete-time BJ model in the z^{-1} operator with model identification and estimation carried out using the RIV algorithm, as implemented by the rivbj and rivbjid routines in the CAPTAIN Toolbox for Matlab, as well as a new version rivbjinit that includes initialization based on the procedure described in the next section 3. This allows the results to be compared directly with the discrete-time modelling results obtained by Schoukens et al. (2011). However, this initialisation procedure can be applied to any member of the RIV model class, including the continuous-time RIVC and RIVδ algorithms. For convenience in the remainder of the paper, both the discrete-time RIV algorithms and the associated CAPTAIN routines rivbj and rivbjid will be referred to by the common upper case acronym RIVBJ, with the initialized version denoted by RIVBJinit. Similar notation will be used for other algorithms and associated computer-based routines such as the BJ routine in the Matlab™ SID Toolbox.

[1]For convenience, only the SISO form of the model is shown here but the model and associated RIV analysis can be extended straighforwardly to the MISO form. Note also that the nomenclature used for transfer functions is that used for RIV estimation since 1976 Young (1976, 2011); in this unified context, models intended for *Prediction Error Minimization* (PEM) estimation in Matlab™ would use $C(\mu^{-1})/D(\mu^{-1})$ for the ARMA noise model; and/or $B(\mu^{-1})/F(\mu^{-1})$ for the system model.

3 Initialization Procedure

This new approach is very straightforward and involves the following three steps that are discussed in more detail below:

1 First, identify a high order *Auto-Regressive eXogenous Variables* (ARX) model from the input-output data set $\{u(k),\ y(k)\}$. This provides an initial model of the system that may be severely over-parameterized but normally explains the data well provided the model order is high enough to ensure that the model residuals (prediction errors) are serially uncorrelated.

2 Second, apply a model reduction technique that is based on the analysis of input-output data (i.e. not one of the standard model reduction techniques that require state-space descriptions) to the input-output data obtained by simulation of the high order ARX model: i.e. the input $u(k)$ and the noise-free simulated output, denoted by $\hat{x}(k)$. These data are used both to identify a suitable model order and structure for the reduced order model and to estimate the parameters that characterise this structure.

3 Finally, there are two possibilities:

(i) Re-estimate the model identified in stage **2**, based on the original noisy input-output data, using the full RIVBJ algorithm with the system model initialized with the parameter estimates obtained from stage **2**, rather than by the standard RIVBJ approach which uses least squares estimation of the ARX model as the first iterative step.

(ii) Re-estimate the model based on the original noisy input-output data using the full *recursive* RIVBJ algorithm with the initial, prior parameter estimates based on the parameter estimates obtained in stage **2**. The associated prior covariance matrix is chosen to allow the algorithm to find the estimated parameters that characterise the local minimum of the cost-function-parameter hyperspace that lies in the region around the prior estimates.

3.1 Comments

1. Since the ARX-estimation is linear-in-the-parameters, is does not suffer from local minima problems but yields asymptotically biased parameter estimates when applied to BJ models. However, it is well known that high order ARX models, where the identified order is based on an order that yields serially uncorrelated (white) residuals, can explain the data very well. Consequently, the model provides a very good input-output description of the underlying model, although the high order means that the estimation is statistically inefficient because of the large number of parameters and associated pole-zero cancellations. This approach, which has been called 'repeated least squares' by Åström and Eykhoff (1971) involves estimating ARX(n_{arx}, n_{arx}) models of increasing order n_{arx}, each time checking on the significance of the result in some manner. Åström and Eykhoff suggested evaluating the significance of the decrease in the sum of the squares using a Student 't' test (e.g. Kendall and Stuart, 1961). However, there are other statistical approaches, such as the Durbin-Watson or Ljung-Box statistics, and order identification criteria, such as the *Akaike Information Criterion* (AIC), that can be used to select a suitable n_{arx} order. In the later example, the

AIC is used for this purpose, although there is no claim that this is the best approach. Indeed, practical experience suggests that the selection of n_{arx} is not critical, provided the explanation of the data as defined, for instance, by measures such as the root mean square (rms) of the model error, or the coefficient of determination based on the variance of the modelling errors (R_T^2), has reached a 'plateau' where little improvement occurs with increasing order.

2. The simulated output of the high order ARX model identified at stage **2** normally provides very good input-output description of the underlying system model, so it is straightforward to use these data as a basis for the identification of a suitable low-order model structure, as well as the estimation of the parameters that characterise this identified model structure. This is effectively an exercise in high-order model reduction and one effective approach to such model reduction is the *Data-Based Mechanistic* (DBM) method of model reduction: see Young (1999); Young and Ratto (2011), where the model reduction is referred to as 'nominal model emulation'. As the simulated output of the high order ARX model provides an estimate $\hat{x}(k)$ of the noise-free output $x(k)$, the noise level is very low and the model reduction exercise employs the *Simplified RIV* (SRIV) identification algorithm (Young, 1984, 2011), which is an option in the RIVBJ routine.

3. The parameter estimates obtained at stage **2** for the system TF model in equation (1) should be quite good but, since they are based entirely on the simulated output series, they are not maximum likelihood estimates and do not provide information on the uncertainty in the parameter estimates. In order to obtain the final maximum likelihood estimates, therefore, it is necessary to apply full RIVBJ estimation to the original noisy data. Note, however, that the model obtained at stage **2** might be good enough for many practical applications that do not demand optimality (see the results in the later simulation example).

4. Of the two possibilities in stage **3**, option (i) is clearly the simplest and computationally the most efficient, so this is used for the example in section 4. Option (ii) is basically straightforward because it uses the existing RIVBJ routine. However, the associated prior covariance matrix P_0 must be selected small enough to constrain the recursive parameter estimates to lie in the region around the prior estimates but large enough to allow for free estimation in this region. One approach is to use a diagonal P_0 with elements set to the diagonal elements of the the covariance matrix associated with the parameters of the model estimated by the SRIV algorithm.

4 Simulation Example

This is based on the 6th order simulation model used by Schoukens et al. (2011) and takes the standard BJ structure, which can be written in the following decomposed form that reveals the presence of the noise-free output $x(k)$:

$$
\begin{aligned}
x(k) &= \frac{B(z^{-1})}{A(z^{-1})}u(k-\delta) && (i)\\
\xi(k) &= \frac{D(z^{-1})}{C(z^{-1})}e(k); \qquad e(k) = \mathcal{N}(0,1) && (ii)\\
y(k) &= x(k) + \xi(k) && (iii)
\end{aligned}
\tag{4}
$$

Here the input $u(k)$ is generated by the following ARMA model:

$$u(k) = \frac{H(z^{-1})}{G(z^{-1})}\varepsilon(k) \qquad \varepsilon(k) = \mathcal{N}(0,1) \tag{5}$$

where,

$$\begin{aligned} G(z^{-1}) &= 1 + 1.76z^{-1} + 1.1829z^{-1} + 0.27806z^{-1} \\ H(z^{-1}) &= 0.5276z^{-1} + 1.5829z^{-2} + 1.5829z^{-3} + 0.5276z^{-4} \end{aligned} \tag{6}$$

while the system model polynomials are as follows:

$$\begin{aligned} A(z^{-1}) &= 1 + 0.9468z^{-1} + 1.7955z^{-2} + 0.8304z^{-3} + 1.7347z^{-4} + 0.8808z^{-5} + \\ &\qquad 0.9558z^{-6} \\ B(z^{-1}) &= 0.0040z^{-1} + 0.0241z^{-2} + 0.0604z^{-3} + 0.0805z^{-4} + 0.0604z^{-5} + \\ &\qquad 0.0241z^{-6} + 0.0040z^{-7} \end{aligned} \tag{7}$$

Finally the noise model is a rather unconventional ARMA process with:

$$\begin{aligned} C(z^{-1}) &= 1 - 0.87727z^{-1} + 0.31106z^{-2} \\ D(z^{-1}) &= 0.10845 + 0.21689z^{-1} + 0.10845z^{-2} = 0.10845(1 + z^{-1})^2 \end{aligned} \tag{8}$$

where it will be noted that $D(z^{-1})$ has two roots at -1.

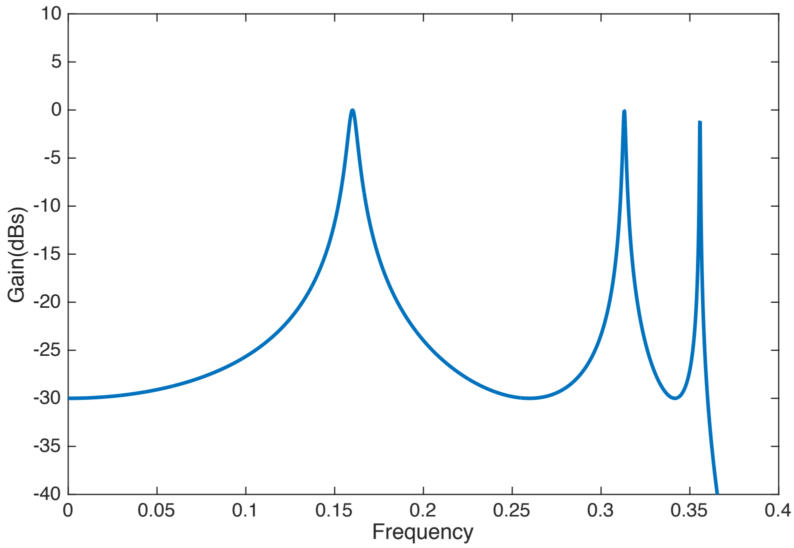

Figure 1: Gain frequency response of the simulation model .

As we see from the frequency response plot in Fig.1, the above model has quite complex dynamics with three sharp resonant peaks in the frequency spectrum, where the two higher frequency modes are located quite close to each other. As a result, the model represents a quite difficult estimation problem. This difficulty is exacerbated by the nature of the additive noise $\xi(k)$. This is coloured noise and at a reasonably high level, with a noise/signal ratio by standard deviation of 0.34. Moreover, the two numerator roots at -1 mean that the noise model is non-invertible (the impulse response of the inverse noise model is dominated by oscillations whose amplitude increases without bound).

The problems of parametric identification for the model (4) are illustrated by the results presented in Schoukens et al. (2011), where the parametric estimation results using the standard BJ routine in the Matlab SID Toolbox show numerous failures to find a model that captures the highest frequency peak in Fig.1. And it is clearly problematical also for RIV identification because this incorporates an iteratively updated prefilter that includes the inverse of the noise model (see Young, 2015). As might be expected, therefore, similarly mixed results are obtained from RIV estimation using the RIVBJ routine in the CAPTAIN Toolbox[2] when it uses the default method on initialization (least squares estimation of an ARX model of the same order as the user-specified model).

4.1 Initial model order identification

Before proceeding to consider the three-step procedure outlined in the previous section 3, it is instructive to go through the initial stages of RIV identification using the presently available routines in the CAPTAIN Toolbox. This normally starts using the SRIV option of the RIVBJ routine: i.e. identifying a range of models under the assumption that the residual noise is white. Now, while this is certainly not the case in this example, so that the estimation is statistically inefficient, we will see that the SRIV option yields consistent and rather good estimates of the parameters, as it does in most practical situations. The table below is part of the table (verbatim) produced by RIVBJ in this case:

```
__________________ BEST 7 models (YIC) __________________
na  nb td  nc nd  YIC (TF)    Rt2   BIC (ARMA)    S2         AIC
 6   7  0   0  0   -4.888   0.8824  -61978.47   2.812e-03   -22.508
 6   6  0   0  0   -4.806   0.8821  -61942.90   2.819e-03   -22.515
 7   8  0   0  0   -4.791   0.8814  -61884.52   2.837e-03   -21.398
 8   9  0   0  0   -4.647   0.8826  -61998.53   2.807e-03   -20.345
 7   6  0   0  0   -4.627   0.8824  -61958.30   2.812e-03   -22.359
 8   6  0   0  0   -2.463   0.8824  -61950.06   2.812e-03   -22.242
 5   6  0   0  0   -1.894   0.7364  -53460.91   6.304e-03   -18.145
```

The identification of the true [6 7 0] system model order is fairly clear in the above table with YIC (Young, 2011) and BIC (Akaike, 1979; Schwarz, 1978) selecting it as best, while the AIC (Akaike, 1974) marginally prefers the [6 6 0] model. These models explain the measured output well, as shown by the coefficient of determination based on the model output, Rt2 (R_T^2) and the mean squared value of the output errors, S2. Note that in this initial model structure identification analysis and all the subsequent RIVBJ analysis, a 'pre-pended' period of 50 zeros is added to the both the input and output data. This is because it is well known that one should make sure that any changes in the input signal do not occur too early relative to the order of the desired model and that this can be achieved by taking this precaution (which does not otherwise affect the estimation). Indeed, a warning to this effect appears in Matlab SID Toolbox routines when step or impulse inputs are utilized. It is particularly advisable in RIV-type estimation because it uses active prefiltering of the data and this can be badly affected by initial transients (see the discussion on this in Young, 2014).

RIV-type estimation is based on *Pseudo-Linear Regression* (PLR), which has the advantage that it can yield recursive estimates of the parameters that allow the user to evaluate whether the convergence is satisfactory (see the simulation examples in section

[2]CAPTAIN can be downloaded free via the web site http://captaintoolbox.co.uk/Captain_Toolbox.html

6 of Young, 2015). Proceeding with this initially identified [6 7 0] model, therefore, we can now use the RIVBJ routine to generate the recursive estimates of the parameters and examine how the parameters are converging on their final values during the final iteration of the algorithm. The recursive estimate of the a_1 parameter is plotted in Fig.1 and we see that convergence is relatively rapid and unambiguous, with the estimation virtually complete after 5000 samples. Similarly satisfactory convergence occurs for all the model parameters.

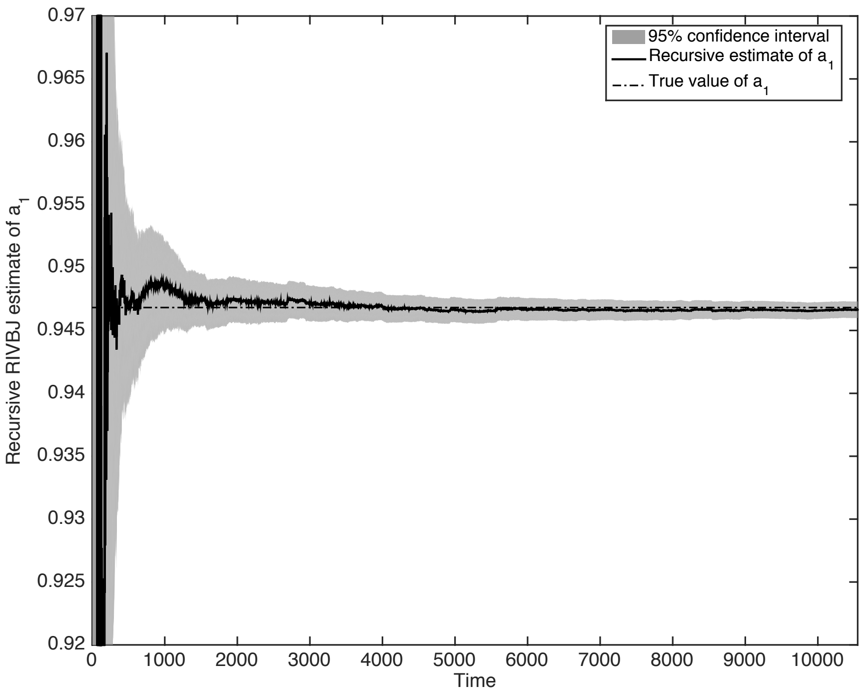

Figure 2: Recursive RIVBJ estimate of the a_1 system model parameter.

4.2 Final model estimation

Having identified a suitable model structure, we can now proceed to evaluate the three-step procedure outlined in the previous section 3 using *Monte-Carlo Simulation* (MCS) analysis based on 20,000 samples of the input-output data and 100 realizations, each time selecting different random realizations of the white noise $e(k)$. The repeated least squares identification of ARX models, using the data from the first random realisation and AIC order identification to define the best ARX model order, suggests an order $n_{arx} = 34$, i.e. a high order ARX(34,34) model. This is used for all realizations, with the simulated deterministic output of this model $\hat{x}(k)$, together with the input $u(k)$ providing the noise-free data for SRIV-based model reduction, again using the RIVBJ routine in the CAPTAIN, with the same [6 7 0] identified structure. The estimated parameter vector $\hat{\phi} = [\hat{a}_1\ \hat{a}_2\ \ldots\ \hat{a}_6\ \hat{b}_0\ \ldots\ \hat{b}_6]^T$ is then used to initialize final RIVBJ estimation, this time using the modified version RIVBJinit, with the full model structure [6 7 0 2 2] of the true model (4).

The initial MCS results show that the identification problems reported by Schoukens

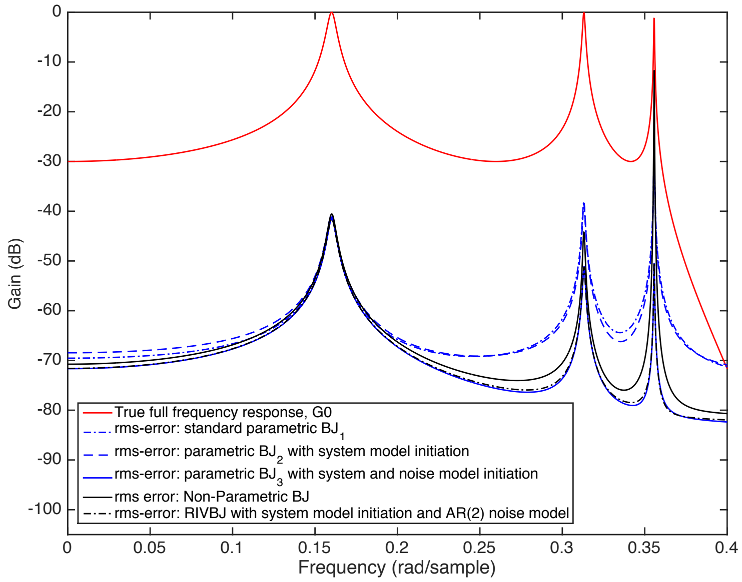

Figure 3: Comparison of the mean square errors in the frequency response of the models identified by the various algorithms.

et al. (2011) are not only caused by initiation problems, although this is the major reason for them. As we shall see below, another problem is the additive noise model where the numerator polynomial $D(z^{-1})$ causes problems with parametric estimation methods such as RIVBJ and the PEM-based BJ routine which explicitly and implicitly, respectively, use optimal prefilters $PF(z^{-1})$ of the form,

$$PF(z^{-1}) = \frac{C(z^{-1})}{D(z^{-1})A(z^{-1})} \tag{9}$$

In this case, these are unstable because the denominator polynomial is the convolution of $D(z^{-1})$, with its two roots at -1, and $A(z^{-1})$. Not surprisingly, therefore, the initial MCS analysis, using the model structure $[6\ 7\ 0\ 2\ 2]$ at the final RIVBJ identification stage, has 3 failures where the estimates are rather volatile, with one failing to identify the highest frequency peak in the spectrum. Moreover, the computational time for each random realisation is quite long, suggesting difficulties in convergence. When these poor results are removed from the RIVBJ ensemble, however, the MCS average estimation performance is very good.

Despite the small number of failures, the RIVBJ performance is much better than that of the standard BJ routine, where there are 79 failures. This suggests that the major improvement in RIVBJ performance is due to the new initialization procedure; and further MCS analysis confirms this by showing that the main reason for the failures is the non-invertible ARMA model used to generate the additive noise. In particular, it is found that, if noise model order specified for RIVBJ is changed from ARMA(2,2) to an autoregressive AR(34) model (following from the identified high order of the ARX model, where this is the effective noise model) then the 3 failures are removed.

This improvement in RIVBJ performance makes sense because the high order AR(34) specification can be considered as a quasi-nonparametric noise model estimation which is consistent with the Schoukens et al. (2011) use of a nonparametric noise model and explains why their approach is also effective in avoiding the problem introduced by the offending $D(z^{-1})$ polynomial in the noise model.

It is interesting, therefore, to generate new MCS results that can be compared with those presented in the Schoukens et al. (2011) paper, where they also consider the results obtained using the PEM-based BJ algorithm in Matlab$^{\text{TM}}$, with and without initialization. In the following analysis, the BJ1, BJ2 and BJ3 results are, respectively, those obtained when the BJ routine is used in its standard form; with the system model initialized with the estimates of the model parameters generated by the non-parametric algorithm; and with both the system and noise models initialized with the estimates form the non-parametric algorithm. As noted previously, the model obtained at stage **2** of the proposed procedure could be useful in practical terms, so this is also included in the MCS analysis and denoted by the acronym Mi.

All of the MCS results obtained in this manner are shown in Fig.3 and Tables 1 to 3. These are once again based on 100 MCS realizations but the number of input-output data samples used for each realization is reduced to 10,000 to see if this affects the algorithmic performance.

Parameter		a_1	a_2	a_3	a_4	a_5	a_6	Failures
	True	0.9468	1.7955	0.8304	1.7347	0.8808	0.9558	
	SE Opt	0.00027	0.00044	0.00063	0.00060	0.00040	0.00025	
RIVBJinit	Mean	0.9468	1.7955	0.8305	1.7349	0.8809	0.9559	1
	SD	0.00027	0.00045	0.00065	0.00060	0.00040	0.00024	
	SE	0.00025	0.00041	0.00061	0.00063	0.00043	0.00026	
FRD	Mean	0.9468	1.7955	0.8305	1.7348	0.8808	0.9558	2
	SD	0.00036	0.00049	0.00074	0.00077	0.00053	0.00046	
	SE	–	–	–	–	–	–	
BJ1	Mean	0.9468	1.7955	0.8305	1.7349	0.8809	0.9559	62
	SD	0.00031	0.00047	0.00062	0.00053	0.00038	0.00028	
	SE	0.00036	0.00059	0.00087	0.00087	0.00058	0.00035	
BJ2	Mean	0.9468	1.7955	0.8305	1.7349	0.8809	0.9559	2
	SD	0.00027	0.00045	0.00063	0.00060	0.00041	0.00027	
	SE	0.00027	0.00044	0.00064	0.00063	0.00042	0.00026	
BJ3	Mean	0.9468	1.7955	0.8305	1.7349	0.8809	0.9559	2
	SD	0.00027	0.00044	0.00063	0.00060	0.00040	0.00025	
	SE	0.00028	0.00045	0.00066	0.00068	0.00046	0.00028	
Mi	Mean	0.9474	1.7955	0.8301	1.7326	0.8792	0.9545	1
	SD	0.00043	0.00069	0.00098	0.00086	0.00057	0.00035	

Table 1: MCS estimation results for the $A(z^{-1})$ polynomial parameters obtained by the various methods: based on 100 realizations using 10,000 data samples. SD denotes standard deviation of MCS results and SE denotes the algorithmic estimated standard error for a single realization.

Comments

1. Fig.3 is similar to the spectral plots in Schoukens et al. (2011). However, note one difference: if any algorithm fails to identify the highest frequency peak in the spectrum, the results from the associated MCS realization are removed in order to compute the means and standard deviations for that algorithm. The number of such failures are reported on Fig.3, as well as in Tables 1 and 2, so the results presented here represent the best performance of each algorithm. Except for the large number of failures for BJ1, however, there is not much significance in these failure figures because they are so small and appear to be a consequence of the sample size reduction.

2. The tabulated results confirm those in Fig.1, with the RIVBJinit standard deviations close to those of BJ3 and the optimum standard errors; while the FRD standard deviations are somewhat higher confirming the results obtained by Schoukens

Parameter		b_0	b_1	b_2	b_3	b_4	b_5	b_6	Failures
	True	0.0040	0.0241	0.0604	0.0805	0.0604	0.0241	0.0040	
	SE Opt	0.00013	0.00030	0.00051	0.00053	0.00045	0.00025	0.00011	
RIVBJinit	Mean	0.0040	0.0241	0.0603	0.0805	0.0603	0.0241	0.0040	1
	SD	0.00013	0.00030	0.00051	0.00053	0.00047	0.00026	0.00012	
	SE	0.00012	0.00026	0.00046	0.00050	0.00047	0.00027	0.00013	
FRD	Mean	0.0040	0.0241	0.0603	0.0805	0.0603	0.0241	0.0040	2
	SD	0.00015	0.00033	0.00055	0.00057	0.00049	0.00028	0.00013	
	SE	–	–	–	–	–	–		
BJ1	Mean	0.0040	0.0241	0.0603	0.0804	0.0603	0.0241	0.0040	62
	SD	0.00027	0.00035	0.00059	0.00056	0.00063	0.00038	0.00023	
	SE	0.00021	0.00037	0.00064	0.00065	0.00065	0.00039	0.00022	
BJ2	Mean	0.0040	0.0241	0.0603	0.0804	0.0603	0.0241	0.0040	2
	SD	0.00029	0.00041	0.00070	0.00062	0.00067	0.00039	0.00024	
	SE	0.00027	0.00036	0.00064	0.00055	0.00066	0.00038	0.00029	
BJ3	Mean	0.0040	0.0241	0.0603	0.0804	0.0603	0.0241	0.0040	2
	SD	0.00013	0.00030	0.00051	0.00053	0.00045	0.00025	0.00011	
	SE	0.00012	0.00028	0.00048	0.00054	0.00049	0.00029	0.00013	
Mi	Mean	0.0040	0.0242	0.0605	0.0808	0.0606	0.0243	0.0040	1
	SD	0.00030	0.00040	0.00076	0.00075	0.00089	0.00060	0.00044	

Table 2: MCS estimation results for the $B(z^{-1})$ polynomial parameters obtained by the various methods: based on 100 realizations using 10,000 data samples. SD and SE defined in Table 1.

et al. (2011). In other words, RIVBJinit performs similarly to BJ3 but, like BJ2, does not require initialization of the noise model parameters. Note that the main difference between the performance of RIVBJinit and BJ2 are the larger standard deviations of the $B(z^{-1})$ polynomial parameter estimates for BJ2.

3. The results for Mi are interesting because they show that the estimates, as obtained quite simply and rapidly at stage **2** in the proposed initialization procedure, are statistically consistent and, in this example, the associated standard deviations are not that much greater than those of the more statistically efficient methods RIVBJinit and BJ3. It is clear, therefore, that these estimates could well have practical utility in applications such as control system design.

4.3 Results with an Invertible Noise Model

One way that we can investigate to what extent the non-invertible noise process is affecting the results reported in the previous section is to retain the same system model (4)(*i*) but replace this noise process by the following more standard one.

$$\xi(k) = \frac{1+0.2z^{-1}}{1-1.4z^{-1}+0.7z^{-2}}e(k); \qquad e(k) = \mathcal{N}(\sigma^2, 1) \tag{10}$$

in which σ^2 is set to yield the same signal/noise ratio of 0.34 used in the previous MCS analysis with the non-invertible noise process. In this case, the high order ARX model order was identified as ARX(27,27). Without initialization of the system model, both RIVBJ and BJ2 still have difficulties identifying the model, with the failures suggesting that the main problem is now the complex nature of the system model. However, with initialization neither algorithm encounters any problems. Tables 3 to 5 report the MCS results for both algorithms, *in which both are initialized using the method proposed in this paper*. This has the advantage that it is ten time faster than the FD initialization procedure (e.g. 0.5 compared with 5 secs. on a Mac OSX 10.10.5, with 2 x 2.66 GHz 6-Core Intel Xeon) and the effect is the same. It is clear from the tabulated results, that both routines are performing reliably and achieving optimal estimation performance.

Also shown as the last item in the tables are the MI results and, once again, the results are quite good with consistent parameter estimates and average standard deviations not much greater those of RIVBJinit and BJ2.

Parameter		a_1	a_2	a_3	a_4	a_5	a_6
	True	0.9468	1.7955	0.8304	1.7348	0.8808	0.9558
	SE Opt	0.00012	0.00020	0.00030	0.00030	0.00020	0.00012
RIVBJinit	Mean	0.9468	1.7955	0.8304	1.7347	0.8808	0.9558
	SD	0.00013	0.00022	0.00033	0.00033	0.00022	0.00013
	SE	0.00014	0.00022	0.00033	0.00033	0.00022	0.00014
BJ2	Mean	0.9468	1.7955	0.8304	1.7348	0.8808	0.9558
	SD	0.00014	0.00022	0.00033	0.00033	0.00022	0.00013
	SE	0.00014	0.00022	0.00033	0.00033	0.00022	0.00014
MI	Mean	0.9468	1.7953	0.8301	1.7339	0.8802	0.9553
	SD	0.00012	0.00029	0.00042	0.00028	0.00017	0.00013

Table 3: MCS estimation results for the $A(z^{-1})$ polynomial parameters obtained by the RIVBJinit and BJ routines, with system model parameters initialized using proposed method. SD and SE defined as in Table 1.

Parameter		b_0	b_1	b_2	b_3	b_4	b_5	b_6
	True	0.0040	0.0241	0.0604	0.0805	0.0604	0.0241	0.0040
	SE Opt	0.00012	0.00017	0.00031	0.00028	0.00032	0.00018	0.00012
RIVBJinit	Mean	0.0040	0.0241	0.0604	0.0805	0.0603	0.0241	0.0040
	SD	0.00013	0.00018	0.00033	0.00030	0.00035	0.00020	0.00014
	SE	0.00012	0.00017	0.00031	0.00028	0.00032	0.00018	0.00012
BJ2	Mean	0.0040	0.0241	0.0604	0.0805	0.0603	0.0241	0.0040
	SD	0.00013	0.00018	0.00033	0.00030	0.00035	0.00020	0.00014
	SE	0.00012	0.00017	0.00031	0.00028	0.00032	0.00018	0.00012
MI	Mean	0.0040	0.0242	0.0604	0.0806	0.0604	0.0242	0.0040
	SD	0.00026	0.00030	0.00058	0.00046	0.00067	0.00039	0.00032

Table 4: MCS estimation results for the $B(z^{-1})$ polynomial parameters obtained by the RIVBinit and BJ algorithms, with system model parameters initialized using proposed method. SD and SE defined as in Table 1.

Parameter		c_1	c_2	d_1
	True	-1.4	0.7	0.2
	SE Opt	0.009	0.009	0.012
RIVBJinit	Mean	-1.501	0.701	0.198
	SD	0.010	0.010	0.013
	SE	0.009	0.009	0.012
BJ2	Mean	-1.501	0.701	0.198
	SD	0.010	0.010	0.013
	SE	0.009	0.009	0.013

Table 5: MCS estimation results for the $C(z^{-1})$ and $D(z^{-1})$ noise polynomial parameters obtained by the RIVBJ and BJ algorithms, both with system model parameters initialized using proposed method: based on 100 realizations using 10,000 data samples. SD and SE defined as in Table 1.

5 Conclusions

This paper has shown how it is possible to modify the standard method of initialization for the *Refined Instrumental Variable* (RIV) class of identification algorithms in those rare situations where the algorithms encounter convergence problems. The efficacy of the new initialisation is demonstrated by Monte Carlo simulation analysis applied to a discrete-time simulation example suggested by Schoukens et al. (2011). This uses a suitably modified version RIVBJinit of the RIVBJ routine in the CAPTAIN Toolbox for Matlab. However, the same approach can be used in RIV algorithms for hybrid continuous-time models, such as the RIVCBJ routine in CAPTAIN or the equivalent SRIVC routine in the CONTSID Toolbox, although it has not yet been evaluated in this regard. In addition, the results show that it can be used also for initializing the BJ routine in the Matlab SID Toolbox.

The paper has revealed also that the noise model in Schoukens et al example introduces a problem for the optimal prefilters used in RIV-type identification and that this problem is best avoided by approximating the noise model with a high order AR model. This can be considered as a quasi-nonparametric noise model and, therefore, an alternative to the Schoukens et al's use of a nonparametric noise model, but one that does not require estimation in the frequency domain and so is computationally more efficient. However, both methods suffer because they require a reasonable sample size to facilitate the non-parametric estimation. There is also a need for more research on the identification of the high order ARX model order n_{arx} because the AIC-based approach used here may not be the best.

Finally, the paper has only addressed the initialization of the main system model and has not considered additional noise model initialization, which would be possible in RIVBJ but adds complexity and does not appear to be required. Moreover, from a practical standpoint, RIVBJ works well almost all of the time using the standard least squares estimation of a low order ARX model for initialization, so the special initialization based on initial high order ARX model identification, such as that described in this paper, does not need to be invoked very often and so can be utilized as an option, rather than the default approach.

Acknowledgements

The comparative MCS analysis reported in section 4 of the paper was greatly facilitated by Professor Johan Schoukens, who kindly provided the Matlab code used to generate their results in their paper. I am greatly indebted to Professor Schoukens for this assistance, as well as the provision of their excellent FDIDENT Toolbox required for the frequency domain estimation.

References

Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control 19*, 716–723.

Akaike, H. (1979). A Bayesian extension of the minimum AIC procedure of autoregressive model fitting. *Biometrika 66*, 237–242.

Åström, K. J. and P. Eykhoff (1971). System identification - a survey. *Automatica 7*, 123–162.

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

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

Schoukens, J., Y. Rolain, G. Vandersteen, and R. Pintelon (2011). User friendly box-jenkins identification using nonparametric noise models. In *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pp. 2148–2153.

Schwarz, G. (1978). Estimating the dimension of a model. *Annal. Statist. 6*, 461–464.

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

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

Young, P. C. (1999). Data-based mechanistic modelling, generalised sensitivity and dominant mode analysis. *Computer Physics Communications 117*, 113–129.

Young, P. C. (2011). *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. Springer-Verlag, Berlin.

Young, P. C. (2014). Comment on ‘projection-based identification algorithm for grey-box continuous-time models’. *Systems and Control Letters 69*, 62–64.

Young, P. C. (2015). Refined instrumental variable estimation: Maximum likelihood optimization of a unified Box-Jenkins model. *Automatica 52*, 35–46.

Young, P. C. and M. Ratto (2011). Statistical emulation of large linear dynamic models. *Technometrics 53*, 29–43.