Recent Developments in the CAPTAIN Toolbox for Matlab

Peter C. Young * **C. James Taylor** **

* *Systems and Control Group, Lancaster Environment Centre, Lancaster University, UK; Fenner School of Environment and Society, Australian National University, Canberra; Australia; e-mail: p.young@lancaster.ac.uk.*
** *Department of Engineering, Lancaster University, UK.*

Abstract: The paper briefly reviews the main features of the CAPTAIN Toolbox, outlines some recent developments and presents a number of examples that demonstrate the performance of these new routines.

Keywords: identification, estimation, forecasting, signal processing, control design

1. INTRODUCTION

The **C***omputer-***A***ided* **P***rogram for* **T***ime Series* **A***nalysis and* **I***dentification of* **N***oisy Systems* (CAPTAIN) Toolbox provides access to novel algorithms for various important aspects of identification, estimation, nonstationary time series analysis and signal processing, adaptive forecasting and automatic control system design, all of which have been developed by the authors and their colleagues over many years. In addition to on-line help with each routine, the CAPTAIN Handbook [Pedregal et al., 2005] lists and gives advice on the use of the various algorithms. It provides a useful companion to the recently published book *Recursive Estimation and Time Series Analysis* [Young, 2011], which contains information on the derivation and use of all of the *en-bloc* and recursive algorithms in CAPTAIN. A web site (`http://captaintoolbox.co.uk/Captain_Toolbox.html`) is devoted to the CAPTAIN Toolbox: it reports recent news and developments, as well as providing short discussions on technical issues. The Toolbox is now free to non-commercial users and can be downloaded from a linked web site (`http://www.es.lancs.ac.uk/cres/captain/download.html`).

Essentially, the CAPTAIN toolbox is a collection of routines (m-files) for the identification and estimation of the various model types and their subsequent use with various application routines available in the Toolbox. These are organized around the core *Refined Instrumental Variable* algorithms for discrete-time (RIV) and continuous-time (RIVC) transfer function model identification (see [Young, 2011]); and various time and state-dependent parameter estimation algorithms that exploit the recursive *Kalman Filter* (KF) and *Fixed Interval Smoothing* (FIS) algorithms. In addition, there are *Non-Minimal State Space* (NMSS) control system design routines (see [Taylor et al., 2000]) that can utilize the RIV and RIVC estimated models for control system deign. In order to allow for straightforward user access, CAPTAIN consists of numerous 'shells' i.e. top level functions that automatically generate the necessary model structures with default options based on the authors' and their colleagues' extensive practical experience with the algorithms (but user-adjustable to override the default settings). In this regard, the main areas of functionality are:

- The functional pairs **rivbjid/rivbj** (for discrete-time TF models) and **rivcbjcid/rivcbj** (for continuous-time TF models estimated from discrete-time sampled data) are provided for order/structure identification (the routines ending in '**id**') and final parameter estimation based on the identified model structure. These functions include, as options, both off-line, recursive and en-bloc versions of the algorithms, in addition to conventional least squares-based approaches. All of the routines produce the estimation results in an 'object' coded form that is compatible with routines in the Matlab SID Toolbox.
- In all of the order/structure identification routines, the model listing, for the selection of models chosen, can be reported in the order of various statistics, such as the coefficient of determination R_T^2 and various identification statistics, including the AIC, BIC and our own YIC [Young, 1989].
- Nonlinear modelling is based around the **sdp** function (see e.g. [Young, 2000, 2011]), which yields non-parametric (graphical) estimates of state-dependent parameters. If required, the user can parameterize these graphically defined nonlinearities using specified nonlinear functions (*e.g.* exponential, power law, radial basis functions etc.) that can then be optimized using standard Matlab functions (see Technical Matter '*Special TF Optimization in CAPTAIN*' in `http://captaintoolbox.co.uk/Captain_Toolbox.html`).
- The *Dynamic Harmonic Regression* (DHR) model (a harmonic regression with time variable parameters [Young et al., 1999]). This is estimated using the hyper-parameter optimization function **dhropt** followed by the DHR algorithm **dhr** primed with these optimized hyper-parameters. It is a particularly useful algorithm for signal extraction, forecasting and back-casting of periodic or quasi-periodic series.
- Other related TVP models are: *Dynamic Linear Regression* (DLR), *Dynamic Auto-Regression* (DAR),

Dynamic AR with eXogenous inputs (DARX), and *Dynamic Transfer Function* (DTF). These are accessible via the functions **dlr**, **dar**, **darx** and **dtfm** and their associated optimization functions **dlropt**, **daropt**, **darxopt** and **dtfmopt**. As discussed in numerous earlier publications, all of these routines are useful for nonstationary time series analysis and forecasting over a wide range of engineering, environmental, biological and socio-economic systems. The DARX algorithm can be used for time-frequency analysis (see [Young, 2011]).

- Various conventional models, identification tools and auxiliary functions, too numerous to list individually here. For example, **kalmanfis** provides a shell to the KF/FIS algorithms for state-space filtering, smoothing and forecasting purposes.
- The PIP control system routines include the **pip** algorithm for pole assignment design and the **pipopt** for PIP-LQ design, together with all other required support routines.
- New routines for closed loop and multiple input TF model system identification and estimation, as outlined in the next section, as well as routines under development or undergoing β testing.

Note that almost all of the recursive estimation and smoothing procedures outlined above will automatically handle missing data in the time series, represented in Matlab by special Not-a-Number (NaN) variables. Indeed, by appending or pre-pending such variables to the data set using the **fcast** function, the toolbox will forecast, interpolate or back-cast as appropriate, without requiring further user intervention.

2. RECENT DEVELOPMENTS

The recently introduced routines are for the estimation of *Multiple Input, Single Output* (MISO) transfer function models, where each elemental transfer function has a different denominator; and for the estimation of a transfer function model in a closed loop. On-line, real-time versions RRIV and RRIVC versions of the RIV and RIVC algorithms are being developed for time variable parameter estimation but these are not yet available. These developments are outlined below and the full details are given in Young [2011].

2.1 RIV/RIVC MISO Model Estimation

One limitation of the standard **rivbj/rivcbj** routines is that they can only handle transfer function models in which the transfer functions associated with each input share a common denominator polynomial. Within an RIV context, estimation is not straightforward when the denominator polynomials are different because the model cannot be put in the required pseudo-regression form, except by forming an overly parameterized ARMAX model form. The new **rivdd** and **rivcdd** routines are based on the 'back-fitting' approach suggested by Young and Jakeman [1980], evaluated fully by Jakeman et al. [1980] and used recently in a continuous-time context by Garnier et al. [2007]. Further information on these routines, including important practical issues and examples, is given in Taylor et al. [2012 (submitted for SYSID 2012].

2.2 Closed Loop RIV/RIVC Estimation

The identification and estimation of transfer function models in a closed-loop situation has received a lot of attention in the control systems literature: see e.g. Ljung [1999], Söderström and Stoica [1989], Verhaegen [1993], Van den Hof [1998], Gilson and Van den Hof [2005]. Provided there is an external command input signal, simple, sub-optimal transfer function estimation within a closed automatic control loop has always been straightforward when using *Instrumental Variable* (IV) estimation methodology [Young, 1970]. However, recent RIV and RIVC algorithms provide a new stimulus to the development of statistically optimal methods of closed-loop estimation and a number of possible solutions are discussed by Gilson et al (2008, 2009), within a continuous-time setting.

The new **clriv** and **clrivc** routines are 'three-stage' algorithms that derive from an earlier two-stage algorithm [Young et al., 2009]. While the two-stage approach yields consistent, but statistically inefficient parameter estimates, the three-stage algorithm allows for statistically efficient estimation of the enclosed TF model parameters when the system within the closed-loop is stable. The attraction of this new approach is its relative simplicity: in particular, the resulting algorithms are straightforward to implement since they use the existing **rivbj** and **rivcbj** estimation algorithms in CAPTAIN. As a result, the coding of the algorithms in CAPTAIN is straightforward, requiring only three calls to these existing algorithms. Moreover, the simple two stage options are still able to obtain reasonable estimates of the transfer function parameters even if the enclosed system is unstable (but stabilized by the closed loop system).

2.3 Real-Time Recursive Estimation

One advantage of RIV estimation is that it allows easily for recursive estimation and the recursive estimates are exactly the same as those obtained by stage-wise estimation, i.e. where the estimates are obtained by repeated en-bloc estimation. Also, the algorithm can be converted to a *Time-Varying Parameter* (TVP) form by introducing simple stochastic models for parameter variation and modifying the recursive algorithms into a prediction-correction form. However, because the RIV algorithm is iterative, these estimates can only be generated by analysing a block of data in an 'off-line' manner. The *Recursive Refined Instrumental Variable* (RRIV and RRIVC) algorithms currently being developed for inclusion in the CAPTAIN Toolbox allows for 'on-line' real-time recursive estimation by combining the recursive and iterative processes.

The basic idea behind the RRIV and RRIVC algorithms is that, at each recursive update, the latest batch of *ruw* input-output samples (the 'recursive update window' which can be any size defined by the user) is imported, together with the estimated parameter vector $\hat{\boldsymbol{\rho}}$ and associated covariance matrix $\hat{\mathbf{P}}_\rho$ obtained at the previous recursive step. Then the algorithm is applied for *it* iterations in order to estimate the time variable system and noise model parameters over this data window, starting the first iteration recursions with the prior estimate $\hat{\boldsymbol{\rho}}(0)$, and matrix $\hat{\mathbf{P}}_\rho(0)$ set to $\hat{\boldsymbol{\rho}}$ and $\hat{\mathbf{P}}_\rho$ from the previous recursion.

3. ILLUSTRATIVE EXAMPLES

Numerous examples of CAPTAIN-based analysis are described in the CAPTAIN Handbook and Young [2011], as well as many published papers over the past 20 years. The examples considered here simply illustrate the performance of the recent additions to the Toolbox, as outlined in previous sections.

3.1 RIV/RIVC MISO Model Estimation

Here, the RIVDD algorithm is applied first to 2942 input-output samples of simulated data, as generated by the following 3-input model, which is a simulated version of the real winding system considered in Garnier et al. [2007] and Young [2011]:

$$
\begin{aligned}
x(k) &= x_1(k) + x_2(k) + x_3(k) \\
&= \frac{B_1(z^{-1})}{A_1(z^{-1})}u_1(k) \; + \; \frac{B_2(z^{-1})}{A_2(z^{-1})}u_2(k) \; + \\
&\quad \frac{B_3(z^{-1})}{A_3(z^{-1})}u_3(k) \\
y(k) &= x(k) + \xi(k)
\end{aligned}
$$

where z^{-1} is the backward shift operator and

$$
\begin{aligned}
&A_1(z^{-1}) = 1 - 1.9868z^{-1} + 0.9875z^{-2} \\
&A_2(z^{-1}) = 1 - 1.7641z^{-1} + 0.7702z^{-2} \\
&A_3(z^{-1}) = 1 - 1.9229z^{-1} + 0.9267z^{-2} \\
&B_1(z^{-1}) = 8.8108 \times 10^{-5}; \quad B_2(z^{-1}) = 7.689 \times 10^{-3} \\
&B_3(z^{-1}) = 8.396 \times 10^{-4}
\end{aligned}
$$

and the inputs $u_1(k)$, $u_2(k)$ and $u_3(k)$ are three independent PRBS signals. The heavily coloured additive noise $\xi(k)$ is generated by an ARMA(14,1) noise model, driven by zero mean white noise with variance $\sigma^2 = 6.07 \times 10^{-6}$. This level and degree of colour are realistic since they are defined by the estimation results obtained from the analysis of the real data. This example is also selected because it presents a reasonable estimation challenge. From the above parameter values, it will be seen that the steady state gains of the TFs associated with the three inputs $u_1(k)$, $u_2(k)$ and $u_3(k)$ are 0.132, 1.252 and 0.219, respectively. Consequently, the effects of the $u_1(k)$ and $u_3(k)$ on $y(k)$ are much less that $u_2(k)$: indeed, a SRIV estimated 2^{nd} order model between $u_2(k)$ and $y(k)$ explains 97.7% of the variance in $y(k)$. As a result, the effective noise/signal level on the first and third TF components of the MISO model is relatively high and their parameters are more difficult to estimate than those of the second TF component.

For comparison, two other estimation algorithms are applied to the same data set: the PEM algorithm from the Matlab SID Toolbox, using the same model structure and its default initiation; and the standard, common denominator, RIV algorithm. The full estimation results are given in Young [2011] and Figure 1 compares the simulated deterministic model output with the noise-free output $x(k)$. It is clear from these results that the RIVDD estimated model matches the simulated data extremely closely with $R_T^2 = 0.9998$ and the parameter estimates are defined well in relation to the true values. Rather surprisingly, given that these are simulated data, the PEM results ($R_T^2 = 0.9863$ and quite large errors) are not as good

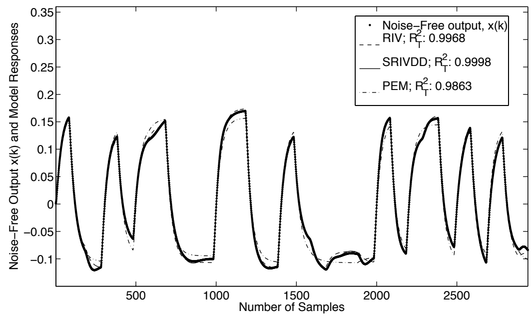

Fig. 1. MISO simulation example: comparison of RIV, RIVDD and PEM estimated model responses.

as those obtained from the standard RIV with common denominator TFs, which does quite well with $R_T^2 = 0.9968$. As expected and shown in Figure 2, the estimation of the first (top panel) and third (bottom panel) component TF models is not as good as the estimation of the most important, second TF (middle panel) due to their much lower steady state gains (see earlier comments).This also seems to be the reason for the poorer PEM estimation results, where the first and third TF models are not estimated well.

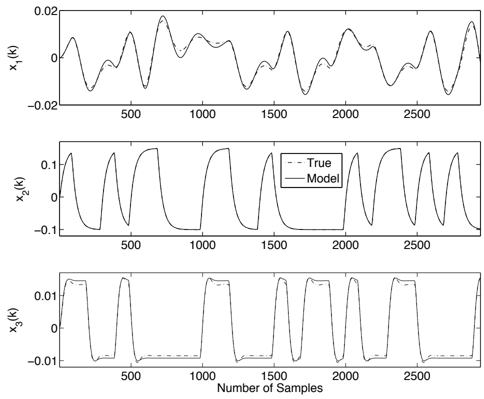

Fig. 2. MISO simulation example: RIVDD estimated component model outputs compared with true, noise-free, simulated outputs.

The results obtained from the **rivdd** analysis of the real winder data are given in Young [2011].

3.2 Closed Loop RIV/RIVC Estimation

Here, the **clrivc** routine is used to estimate a TF model for a system contained within a digital *Proportional-Integral-Plus* (PIP) closed-loop control system (see e.g. Taylor *et al*, 2000, and the prior references therein), with the control system designed using the PIP design routines in the CAPTAIN Toolbox. The controlled system is the following second order, non-minimum phase, continuous-time model:

$$\frac{B(s)}{A(s)} = \frac{-0.5s + 1}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1}$$

where ζ and ω_n take on various values; while the discrete-time additive noise is generated by the following ARMA noise process,

$$\xi(k) = \frac{1 + 0.5z^{-1}}{1 - 0.85z^{-1}}e(k); \quad e(k) = \mathcal{N}(0, \sigma^2) \tag{2}$$

Table 1. 3-Stage estimation and other results for the simulated continuous-time model.

Parameter	a_1	a_2	b_0	b_1
True	0.141	2.0	-0.5	1.0
3-stage RIVC (SR)	0.141	2.002	-0.509	0.981
SE	0.0021	0.0028	0.0028	0.0055
3-stage RIVC (MCS)	0.140	2.000	-0.509	0.978
SD	0.0025	0.0030	0.0030	0.0055
Open-loop (MCS)	0.140	1.999	-0.509	0.978
SD	0.0022	0.0024	0.0030	0.0055
Simple (MCS)†	0.140	1.999	-0.509	0.983
SD	0.0122	0.0154	0.0040	0.0093

SE, standard error; SD, standard deviation; †, two stage

and σ^2 is adjusted to provide different levels of additive noise. The system is simulated in continuous-time within Simulink for 300 secs., using the variable step length ODE45 (Dormer and Prince) solver and enclosed within a discrete-time, PIP control loop based on optimal *Linear-Quadratic* (LQ) design (i.e. optimization of a linear system control performance based on a quadratic performance index).

In this example, the command input signal $r(t_k)$ is a ± 1.0 PRBS signal of length 15000 samples. The MCS analysis is based on 100 realizations, with the white noise variance $\sigma^2 = 0.05$, yielding a noise-to-signal ratio on the control input of 0.30 by standard deviation. However, the noise level at the output is reduced because of the closed loop characteristics and has a noise-to-signal ratio of 0.22.

At the first stage in a typical single run of the three-stage algorithm, the *Bayesian Information Criterion* (BIC) [Schwarz, 1978] clearly identifies a third order, [3 3 0 0 0] model between $r(t_k)$ and $u(t_k)$; and the coefficient of determination between the model output and the noise-free input is $R_T^2 = 0.999$. The MCS results are summarized in Table 1 (note that the noise model parameter estimates are not shown because the text column is not wide enough to accommodate a wider Table): we see that the single run and associated standard error estimates match those obtained from the MCS analysis; and the estimation results also match the optimal results obtained in the equivalent open-loop situation using the **rivcbj** algorithm, demonstrating the optimality of the **clrivc** estimation. Figure 3 shows the ensemble of impulse responses obtained in this example.

emulation_BDrev_long.pdf

Fig. 3. MCS-based comparison of the true and CLRIVC estimated model impulse responses.

On the negative side, the computation time required for each realization is 35 times greater than that required in the equivalent discrete-time case (see Young [2011]), although this is partly explained by the sample length being ten times larger to accommodate the smaller sampling interval used in the continuous-time estimation.

Finally, note that the variance of the estimates obtained with the simple two-stage algorithms is considerably higher than that obtained by adding the third stage, demonstrating the advantages of statistically optimal algorithm design in the case of stable systems. However, these simple algorithms have advantages when the system contained in the closed-loop is unstable or marginally stable, as shown in Young [2011].

3.3 Real-Time Recursive Estimation

This simulation example considers the following second order, discrete-time TVP model, with two variable parameters and one constant parameter:

$$
\begin{aligned}
x(k) &= \frac{b_0(k)}{1 + a_1(k)z^{-1} + 0.7z^{-2}}u(k) \\
\xi(k) &= \frac{1 + 0.2z^{-1}}{1 - 0.85z^{-1}}e(k) \qquad e(k) = \mathcal{N}(0,1) \\
y(k) &= x(k) + \xi(k) \\
u(k) &= \frac{0.5}{1 - 0.5z^{-1}}e_u(k) \qquad e_u(k) = \mathcal{N}(0,1)
\end{aligned}
\tag{3}
$$

where $a_1(k)$ is a pseudo random binary sequence of ± 0.5 centered on -1.4, while $b_0(k) = 1.5 + 0.5sin(0.02k)$. The estimation results were obtained over a data set of 3000 samples and the noise $\xi(k)$ yields a noise/signal ratio by standard deviation of 0.22.

The ML optimized diagonal elements of the $\mathbf{Q}_{nvr}$ matrix, based on a window of $iwh = 1000$ prior data samples, are $\text{NVR}_{a_1} = 0.000035$, $\text{NVR}_{a_2} = 5 \times 10^{-18}$ and $\text{NVR}_{b_0} = 0.00092$, showing that the ML optimization has correctly inferred that the a_2 parameter is constant. The constant parameter SRIV initialization estimates, based on the iwp window of 100 data samples, provide the following initial values for the elements of $\hat{\boldsymbol{\rho}}(0)$ and $\mathbf{P}_\rho(0)$:

$$\hat{a}_1(0) = -1.282;\ \hat{a}_2(0) = 0.699;\ \hat{b}_0(0) = 1.816$$

$$\mathbf{P}_\rho(0) = 10^{-4}\begin{bmatrix} 4.3 & -3.8 & 0.002 \\ -3.8 & 4.16 & -0.002 \\ 0.001 & -0.002 & 102.1 \end{bmatrix}$$

Figure 4 shows the RRIV results obtained from these starting values, with a recursive update interval of 2 samples and 3 iterations for each recursive update. Clearly, the two TVPs and the constant parameter are all estimated well; and the TVP model output closely matches the noise-free output $x(k)$, with $R_T^2 = 0.97$.

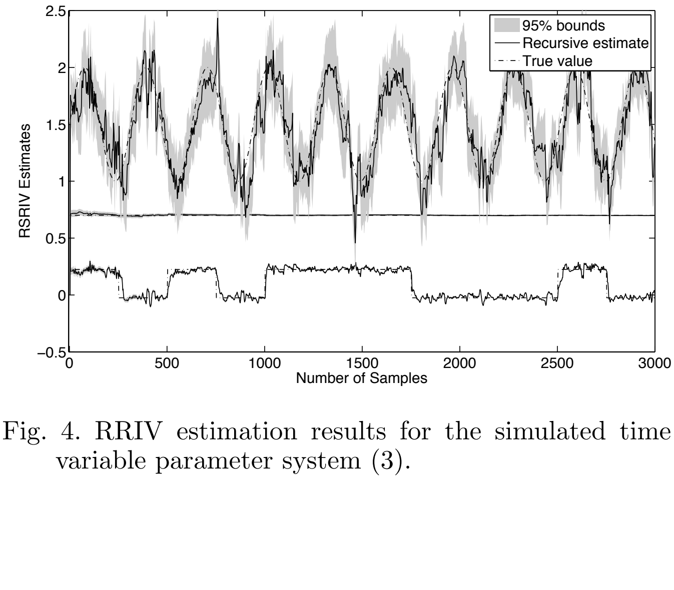

Fig. 4. RRIV estimation results for the simulated time variable parameter system (3).

RRIV estimation has been used in research on the self tuning and adaptive forecasting of flow in rivers. One example [Young, 2010] is concerned with daily rainfall-flow data from the Leaf River catchment, a humid watershed with an area of 1944 km^2 located north of Collins, Mississippi, USA. The *Data-Based Mechanistic* (DBM) model between 'effective rainfall' rainfall $u(k)$ and flow $y(k)$ [see e.g. Young, 2003] is identified, initially, from an estimation data set of 366 days over 1952-1953 and takes the form:

$$y(k) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{(1 + f_1 z^{-1})(1 + f_2 z^{-1})^2} u(k) + e(k) \qquad (4)$$

where this constrained form of the TF model is estimated using special optimization routine that exploits the ability of the rivbj routine to constrain the parameter estimates [1].

For the purposes of forecasting, this TF model is decomposed into physically meaningful, first order TF elements connected in series and parallel, as shown in Figure 5, which then defines the following state space model:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} = \begin{bmatrix} f_{11} & 0 & 0 & 0 \\ 0 & f_{22} & 0 & 0 \\ 0 & 0 & f_{33} & f_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ x_4(k-1) \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ 0 \end{bmatrix} u(k-1) + \begin{bmatrix} \eta_1(k) \\ \eta_2(k) \\ \eta_3(k) \\ \eta_4(k) \end{bmatrix} \qquad (5)$$

with the associated observation equation:

$$y(k) = [\,1\;\,1\;\,0\;\,1\,] \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + g_i u(k) + e(k) \qquad (6)$$

Here, the state variables $x_i(k),\ i = 1, 2, 4$ are, respectively (see Figure 5), the 'slow flow' $x_1(k)$, normally associated with the groundwater processes; and the two 'quick' flows, $x_2(k)$ and $x_4(k)$, normally associated with the surface and near surface processes. The fourth state, $x_3(k)$, is an intermediate state arising from the transfer function decomposition.

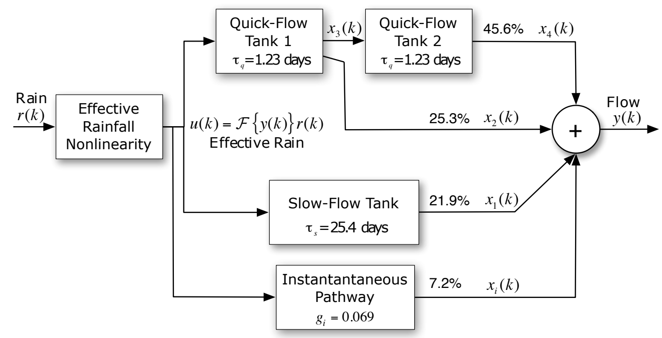

Fig. 5. The DBM model of the Leaf River: here, τ_q and τ_s are the quick and slow residence times, while the percentage figures denote the 'partition percentages' i.e. the percentage of flow passing down the indicated pathways.

[1] see http://captaintoolbox.co.uk/Technical_Matters/Entries/2011/4/29_Special_TF_Optimization_in_CAPTAIN.html

Typical adaptive forecasting results are presented in Figures 6 which shows three years of real-time updating, following initiation after 50 days. Here, the initial covariance matrix for the RRIV parameter estimation is set to reflect some considerable uncertainty in the parameters and so the estimates are rather volatile when the first large rainfall and flow events occur.

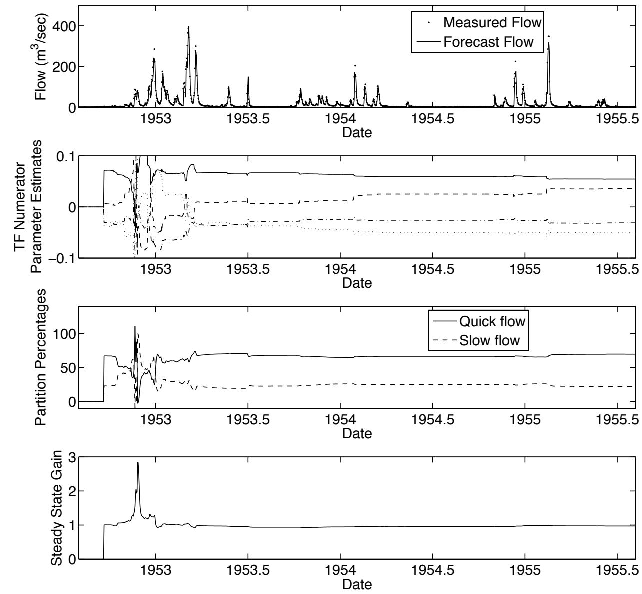

Fig. 6. Leaf River example: three years of real-time updating following initiation after 50 days.

The recursively updated parameter estimates, in this case, are the coefficients of the numerator polynomial in the transfer function since these reflect well the changes in the model characteristics that, when they occur, are linked mainly with the gain of the model. In previous projects of this type, only a single gain parameter has been updated [Lees et al., 1993, 1994, Romanowicz et al., 2006] but this has been modified in this example to illustrate how multiple parameter estimates can be updated if this is necessary. The resulting estimates, as plotted in the upper middle panel, vary quite a lot while the RRIV estimation algorithm is self-tuning the model parameters from the rainfall-flow data. However, after this is completed early in 1953, they then settle down to become fairly stable when sufficient information has been processed to engender confidence in the estimates. Note that the associated changes in the parameters of the state space model (5) are inferred straightforwardly from the changes in these estimated transfer function parameters and the same estimation behaviour is reflected in the associated changes in the partition percentages, shown in the lower middle panel. Here, for clarity, the quick-flow percentage is obtained by aggregating the percentages of the two estimated quick-flow pathways. Figure 7 presents a more detailed view of the adaptive forecasting performance.

REFERENCES

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

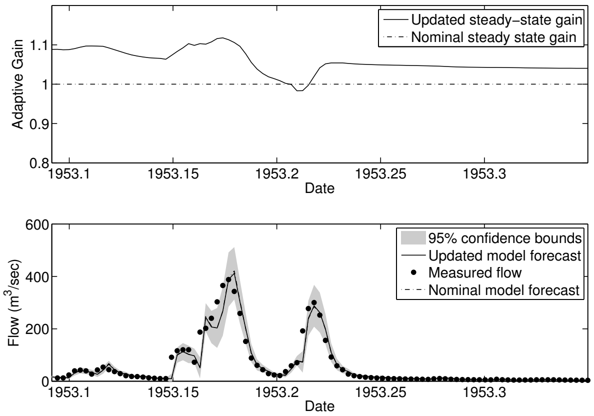

Fig. 7. Leaf River example: the lower panel is a short part of Figure 6 showing more clearly the 24 h ahead forecasting performance; the continually updated estimate of the model steady-state gain is shown in the top panel.

M. Gilson and P. Van den Hof. Instrumental variable methods for closed-loop system identification. *Automatica*, 41:241–249, 2005.

M. Gilson, H. Garnier, P. C. Young, and P. Van den Hof. Instrumental variable methods for closed-loop continuous-time model identification. In Hugues Garnier and Liuping Wang, editors, *Identification of Continuous-Time Models from Sampled Data*, pages 133–160. Springer-Verlag: London, 2008.

M. Gilson, H. Garnier, P. C. Young, and P. Van den Hof. Refined instrumental variable methods for closed-loop system identification. In *Proceedings 15th IFAC Symposium on System Identification SYSID09, St Malo, France*, 2009.

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

M. Lees, P. C. Young, and S. Ferguson. Adaptive flood warning. In P. C. Young, editor, *Concise Encycolpedia of Environmental Systems*, pages 234–236, Oxford, 1993. Pergamon.

M. J. Lees, P. C. Young, S. Ferguson, K. J. Beven, and J. Burns. An adaptive flood warning scheme for the River Nith at Dumfries. In W. R. White and J. Watts, editors, *International Conference on River Flood Hydraulics*, pages 65–77. J. Wiley: Chichester, 1994.

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

D. J. Pedregal, C. J. Taylor, and P. C. Young. *The CAPTAIN Handbook.* Centre for Research on Environmental Systems and Statistics, Lancaster University, 2005.

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.

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

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

C. J. Taylor, A. Chotai, and P. C. Young. State space control system design based on non-minimal state variable feedback: further generalization and unification results. *International Journal of Control*, 73:1329–1345, 2000.

C. J. Taylor, P. C. Young, and D Robertson. Practical experience with unified discrete and continuous–time, multi–input identification for control system design. 2012 (submitted for SYSID 2012).

P. M. J. Van den Hof. Closed-loop issues in system identification. *Annual Reviews in Control*, 22:173–186, 1998.

M. Verhaegen. Application of a subspace model identification technique to identify LTI systems operating in closed-loop. *Automatica*, 29:1027–1040, 1993.

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

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.

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.

P. C. Young. Top-down and data-based mechanistic modelling of rainfall-flow dynamics at the catchment scale. *Hydrological Processes*, 17:2195–2217, 2003.

P. C. Young. Gauss, Kalman and advances in recursive parameter estimation. *Journal of Forecasting (special issue celebrating 50 years of the Kalman Filter)*, 30:104–146, 2010.

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

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.

P. C. Young, D. J. Pedregal, and W. Tych. Dynamic harmonic regression. *Jnl. of Forecasting*, 18:369–394, 1999.

P. C. Young, H. Garnier, and M. Gilson. A simple refined IV method of closed-loop system identification. In *Proceedings 15th IFAC Symposium on System Identification SYSID09, St. Malo, France*, 2009.