

Improved Optimal RIVC Estimation of Continuous-Time Transfer Function Models

New optimal `rivcbjdd` and `sf rivcbjddid/sf rivcbjdd` tools in the CAPTAIN Toolbox: supplements to the existing `rivcbj` and `srivcarma` tools

Peter C. Young
Lancaster University

Lancaster Environment Centre and Data Science Institute

1. Introduction

The `rivcbj` tool in CAPTAIN implements the optimal RIVC estimation algorithm [see 8, 9] for continuous-time, linear *Transfer Function* (TF) models. This is derived by transforming the system and noise sub-models of the complete TF model into ‘Pseudo-Linear Regression’ (PLR) model form [see e.g. 7], whose iterative estimation within the RIV framework yields maximum likelihood estimates of the full model parameters. This algorithm is rapidly convergent and very reliable but it limits the individual TF models associated with multiple inputs to having common denominators. Clearly, this is not a desirable limitation *in some circumstances* and the `rivcdd` tool is a device that attempts to correct this deficiency by an alternative procedure. While this normally converges to a reasonable but sub-optimal estimate of a ‘different denominator’ (dd) model, the sub-optimality is obviously undesirable.

One approach to improving the `rivcdd` performance is to use its estimated parameters as initial conditions for `lsqnonlin`-based optimization of the model. This does not prevent the possible non-convergence of `rivcdd` but, given that `rivcdd` convergence occurs most of the time, it should normally lead to convergence of the associated `lsqnonlin` routine. Then, because the `rivcdd` model can contain an ARMA noise model if this is specified by the user, the result obtained will be optimal in maximum likelihood terms because it is based on *Prediction-Error Minimization* (PEM): [see e.g. 4, and the previous optimal approaches of [3] and [1]]. In other words, although convergence to this optimal solution is not totally guaranteed, it should be attained most of the time, provided `rivcdd` converges. And experience has shown that such non-convergence of `rivcdd` is normally associated with a poorly identified model structure. Indeed, it is most often an indicator of this, suggesting a requirement for re-identification.

Finally, the optimal PEM estimates are returned by `lsqnonlin` together with the error covariance matrix associated with these estimates, from which the estimated *Standard Errors* (SEs) can be computed as the square root of its diagonal elements. The covariance matrix itself is then available for uncertainty studies, such as *Monte-Carlo Simulation* (MCS) analysis used to derive the uncertainty in the full stochastic model response, as well as on any parameters that are derived from the estimated parameters. It would appear that the `tfest` routine in the Matlab Identification Toolbox works in a similar manner to this but it does not appear to contain an additive noise model and, if so, it is formally limited to a situation where this response error is serially uncorrelated white noise. If this is not the case, then its estimates are sub-optimal and the estimated covariance matrix will be in error, so affecting any procedure, such as MCS analysis, that utilizes this matrix.

This technical note describes the results obtained by application of a new `rivcbjdd` tool I have developed for CAPTAIN which implements the above procedure and is so entitled because it is effectively an extension of the existing `rivcbj` tool that allows for the optimal estimation of multi-input, different denominator models. However it does not replace `rivcbj`, which remains the primary tool for continuous-time models be-

cause of its guaranteed convergence and wide applicability, as demonstrated over many years of practical application. The performance of this new CAPTAIN tool is illustrated by the seven examples that are considered in the subsequent sections of this note: two based on simulated data and the other five using measured data that are considered in much greater detail within chapters of my forthcoming book [10]. The rivcbjdd results obtained in this manner are also compared with the results obtained by the Matlab routine tfest, as generated directly using the standard calls to this routine.

2. A Simulation Example

The simulated data used in this example are plotted in figures 1 and 2, where the first shows the three *Pseudo-Random Binary Signal* (PRBS) inputs and the latter is a plot of the noisy output response series, shown as the dotted line, together with various model response series, as discussed below. Note that the three inputs $u_i(t)$, $i = 1 : 3$ are of short length and so this input-output data set presents a rather difficult estimation problem.

The first simulated model takes the following hybrid form:

$$\begin{aligned} x(t) &= \frac{s+0.1}{s+0.5}u_1(t) + \frac{0.3s+0.1}{s+0.1}u_2(t) + \frac{1.2}{s+0.7}u_3(t) \\ y(k) &= x(k) + \xi(k) \\ \xi(k) &= \frac{1+0.2z^{-1}}{1+0.9z^{-1}}e(k) \quad e(k) = \mathcal{N}(0, \sigma^2) \end{aligned} \quad (1)$$

where $e(k)$ is a zero mean, serially uncorrelated, white noise series with variance $\sigma^2 = 0.124$. Here the model structure is known and, using this information, the call to rivcbjdd is shown in lines 1, 2 and 3 of the verbatim code on the next page. Line 4 is then the code for listing the estimated parameters, with their estimated standard errors below them, as reported reported on lines 5 and 6.

```

1 [Mdd,Mnd,edd,eefd,Pard,Psd,Pnd,ym,Ym,Mn,D]=...
2   rivcbjdd([ymn u1 u2 u3],[1 2 0 1 1;1 2 0 1 1;1 1 0 1 1],...
3   [0.0001 1 1 0],[-0.05]);
4 [Pard;sqrt(diag(Psd))']
5 0.9820  0.08394  0.4678  0.2873  0.08630  0.07964  1.2060  0.7
6 0.0236  0.02346  0.0401  0.0204  0.00960  0.00934  0.0380  0.0

```

The ‘help’ file, obtained by the request ‘help rivcbjdd’ on the command line in Matlab, is listed in the Appendix at the end of this note and this shows what all the terms in lines 1 and 2 denote. The results show that all the parameters are estimated well, with standard errors on line 6 smaller than the estimated parameter values on line 5. The estimated model Mdd is reported as:

```

1      0.982 s + 0.08394
2      -----
3          s + 0.4678
4 From input 2 to output :
5      0.2873 s + 0.0863
6      -----
7          s + 0.07964
8 From input 3 to output :
9          1.206
10     -----
11          s + 0.723

```

and the estimated noise model is given as:

```

1 z + 0.1581
2 -----
3 z + 0.8137

```

Here both of the models are shown in the form of Matab model ‘objects’. The response of this model is plotted as the black line in Fig. 2.

The standard tfest model estimate is returned as follows:

```

1 SYS=tfest(D,[1 1 1],[1 1 0], 'InputDelay',[0 0 0])
2 SYS
3 From input "u1" to output "y1":
4 0.8218 s - 0.05332
5 -----
6          s + 0.04089
7 From input "u2" to output "y1":
8 0.2505 s + 0.0891
9 -----
10          s + 0.04092
11 From input "u3" to output "y1":
12 1.201
13 -----
14          s + 0.6596

```

and its response is plotted as the full red line in Fig. 2 where it does not explain the data quite as well as the rivcbjdd estimated model, with the coefficient of determination R_T^2 based on the response error [see e.g. 8], computed as 0.814 compared with $R_T^2 = 0.863$ for the rivcbjdd estimated model. The reason for this difference is not clear but it seems likely that tfest is suffering from the effects of the short data size (see also in the next section 3 which investigates a simulation example with second order component models).

Note that the user-specified nnr setting, which controls the model structure required for the initiation of the rivcdd routine, is set to the default ‘2’ (see the Appendix) in this and the other examples. All three possible settings for nnr yield the same

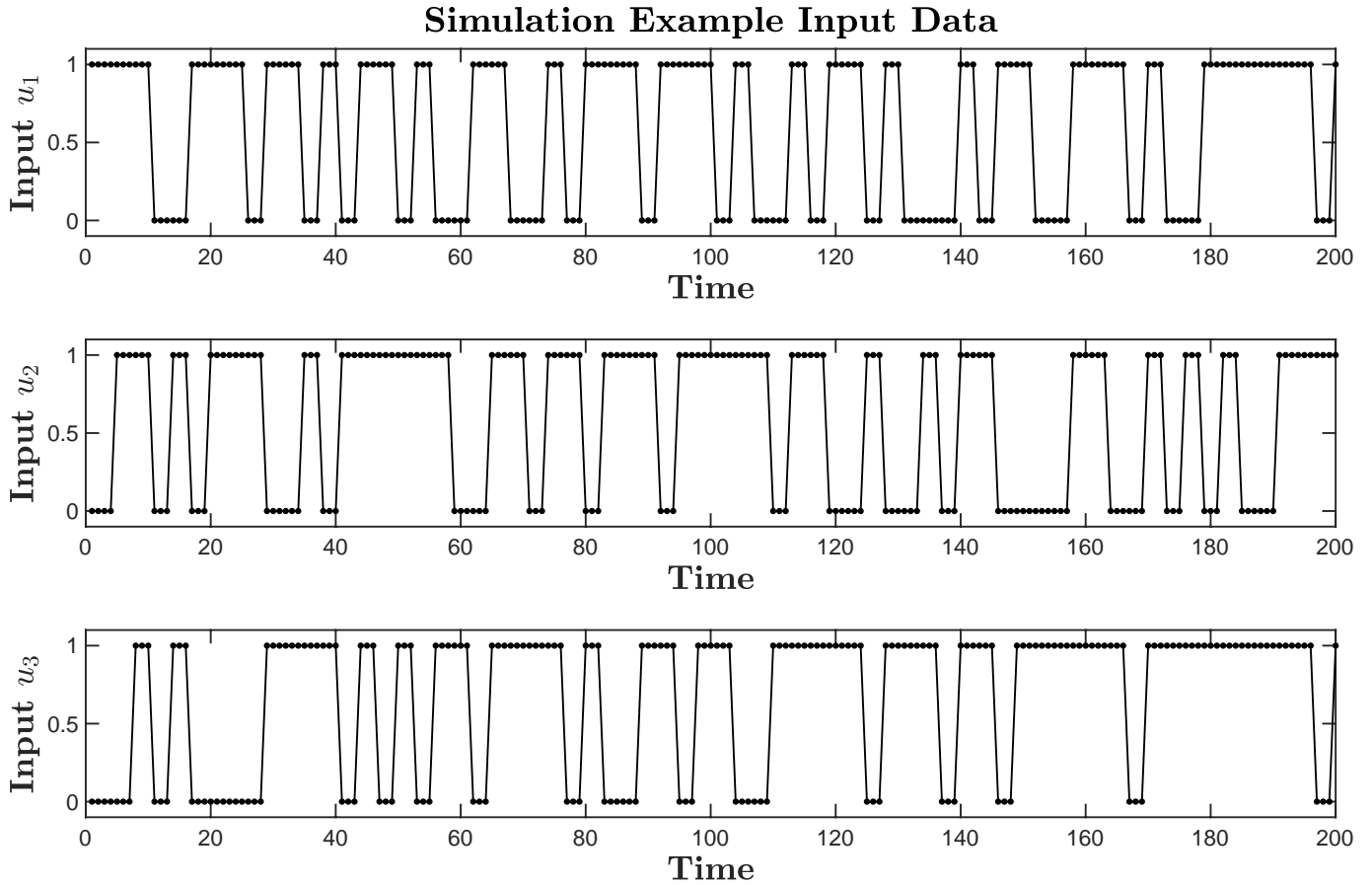


Figure 1: Inputs for the first simulated data example.

results in this case and setting 2, which is the default, happens to provide slightly more rapid convergence (8 seconds compared with 14 seconds for the 1 and zero settings). Note these fairly long computational times, which are due to a combination of the relatively slow `rivcdd` and `lsqnonlin` routines. They contrast with the rapid convergence of `tfest` (1.2 seconds) and the reason for this disparity is discussed later in section 6.

Not surprisingly neither `rivcbj` nor `srivcarma` produce acceptable common denominator models in the above example or in the second simulation example considered in the next section 3.

3. A Higher Order Simulated System

The model components in the previous section, as well as those shown in the subsequent sections 4 to 6 are all first order. Therefore, as a further check that `rivcbjdd` works satisfactorily with a mixed order system, this section considers the following three-input system which is characterised by TF models where

one is first order and two are now second order:

$$x(t) = \frac{s + 0.1}{s^2 + 0.8s + 0.15} u_1(t) + \frac{0.6s + 0.2}{s^2 + 0.3s + 0.02} u_2(t) + \frac{1.2}{s + 0.7} u_3(t)$$

$$y(k) = x(k) + \xi(k)$$

$$\xi(k) = \frac{1 + 0.2z^{-1}}{1 + 0.9z^{-1}} e(k) \quad e(k) = \mathcal{N}(0, \sigma^2) \quad (2)$$

Note that this is a difficult model to estimate because the observation interval is a very short 200 samples and there is no ‘lead in’ of zeros to assist the estimation (see later comments). The verbatim code and plotted results obtained in this case are given below. The model output response explains the data very well, as we see in Fig.3, with $R_T^2 = 0.979$, increasing to $R_T^2 = 0.990$ when this is compared with the simulated noise-free model response. However, probably because of the estimation problems mentioned above, the second parameter in the numerator of the first TF is poorly identified. In fact, it is a little surprising that `rivcbjdd` produces a model that explains the data so well and this suggests that the algorithm is reasonably robust. But clearly, testing with more examples will be required in order to establish that this is indeed the case.

```
1 [Mdd,Mnd,edd,eed,Pard,Psd,Pnd,yh,Ym,M,Mn,D]=...
2 rivcbjddt([ymn u1 u2 u3],[2 2 0 1 1;2 2 0 1 1;1 1 0 1 1],
3 [0.0001 1 1 0],-0.05);
```

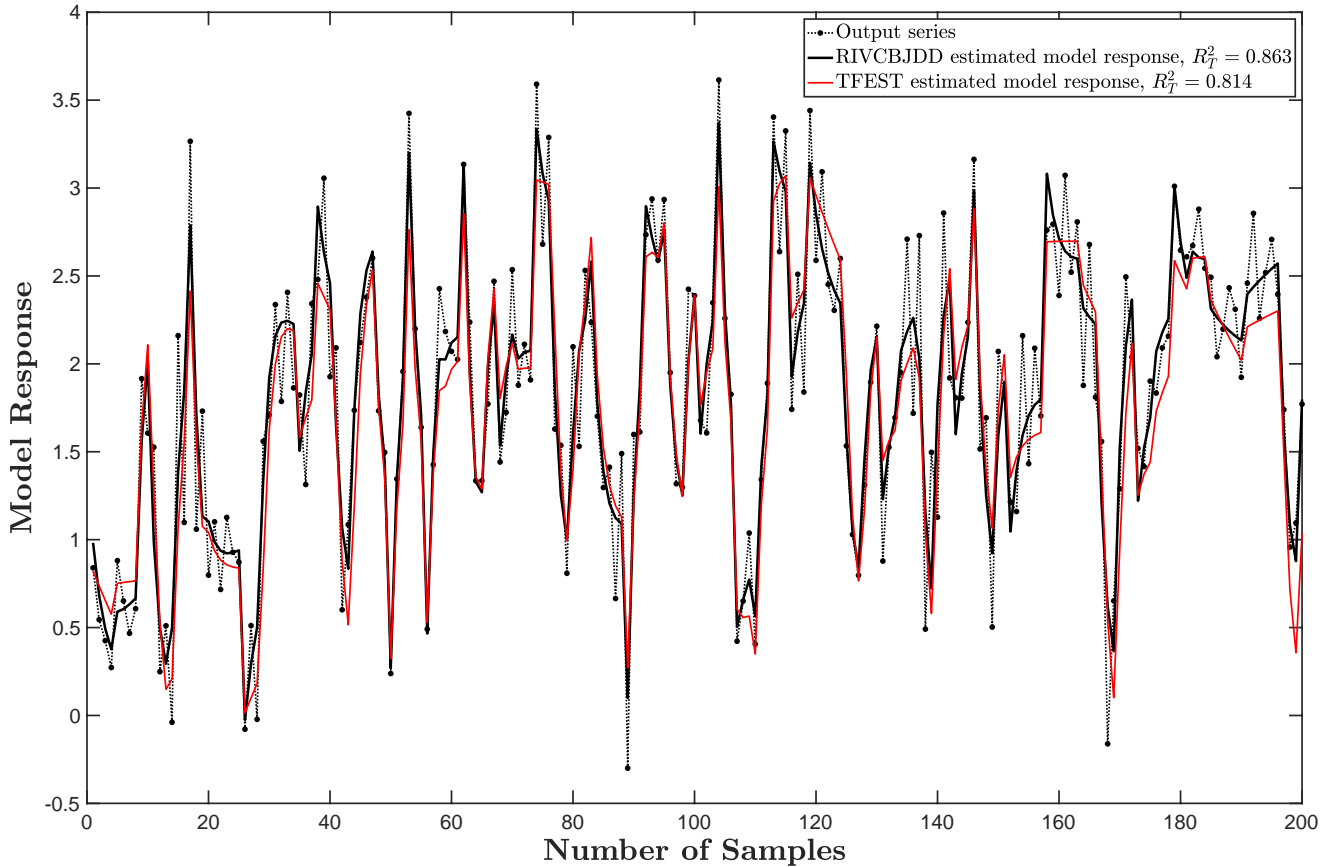


Figure 2: RIVCBJDD estimation results for the first simulated data example.

```

4 Mdd
5 From input 1 to output:
6   1.1026   -0.00589
7   -----
8   s^2 + 0.8107 s + 0.0479
9   From input 2 to output:
10  0.4778 s + 0.5872
11  -----
12  s^2 + 0.8017 s + 0.0540
13  From input 3 to output:
14  1.1582
15  -----
16  s + 0.6304
17 [Pard;sqrt(diag(Psd))']
18 1.1026 -0.00589 0.8107 0.0479 0.4778 0.5872 0.8017 0.0540 1.1582
19 0.0557 0.00799 0.0491 0.00568 0.0527 0.1216 0.1588 0.01088 0.0899 0.046
20 Mnd
21 z + 0.4655
22 -----
23 z + 0.9086

```

```
14 s + 0.01559
```

As shown by the red dash-dot line in Fig. 3, the model's explanation of the data is poor with a lower $R_T^2 = 0.829$, although this is improved to $R_T^2 = 0.858$ (full red line) if the Matlab compare routine is used rather than `lsim` (which is the simulation routine incorporated in `rivcbjdd`). It would appear, therefore, that the optimization in `tfest` has converged on a local minimum, rather than the optimal one.

Note that the `rivcbjdd` estimation results might be improved a little if the IC effects are accounted for by including the $U_i(t)$ unitary vector input, as in the USA unemployment and Global Surface Temperature examples described in the next three sections. This is indeed the case, with the R_T^2 increasing to marginally 0.999. And when this measure is based on the simulated noise-free output $R_T^2 = 0.999$, showing an excellent 99.9% of the underlying simulated output is explained by the `rivcbjdd` estimated model. Of course if the above improvement in the data set is combined with an increase in the data length, then very good results are obtained. For example, with a data length of 2000 samples, the estimation `rivcbjdd` results are shown verbatim below:

The `tfest` estimated model is given verbatim below:

```

1 SYS=tfest(D,[2 2 1],[1 1 0],'InputDelay',[0 0 0])
2 SYS=
3 From input "u1" to output "y1":
4   0.1291 s - 0.001467
5   -----
6   s^2 + 0.03243 s + 0.0015
7   From input "u2" to output "y1":
8   0.4641 s - 0.002797
9   -----
10  s^2 + 0.05961 s + 0.00404
11  From input "u3" to output "y1":
12  0.2694
13  -----

```

```

1 [Pard;sqrt(diag(Psd))']
2 1.0056 0.1013 0.8077 0.1485 1.2057 0.3900 0.02935 0.1952 1.2184
3 0.0135 0.0053 0.0118 0.0054 0.0047 0.0066 0.00400 0.0003 0.0110

```

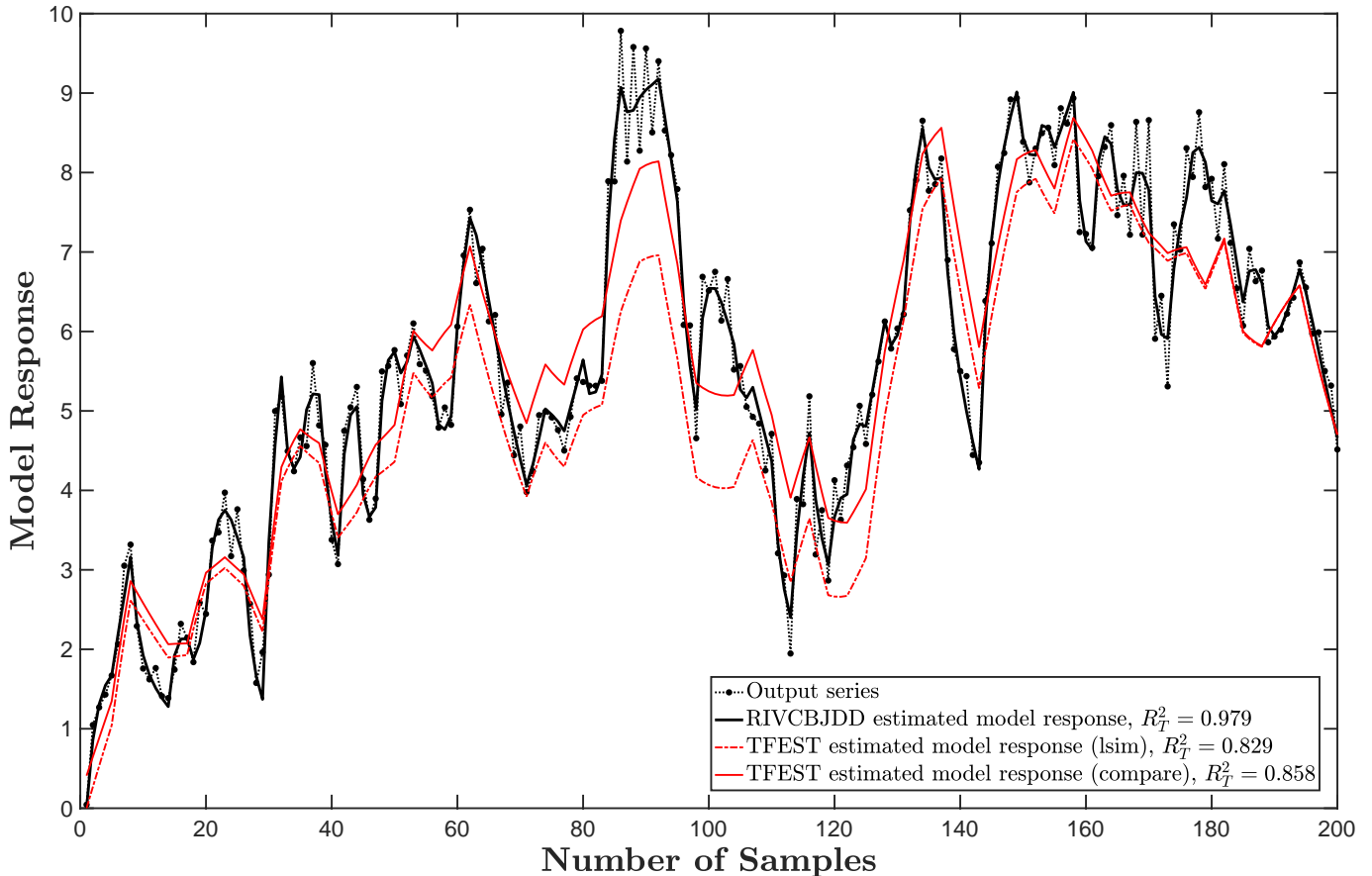


Figure 3: RIVCBJDD estimation results for the second order simulated data example.

Surprisingly, despite this longer data set, `tfest` again fails to obtain a good estimate of the model, with an $R_T^2 = 0.917$ and poor parameter estimates. However, if its estimation is started with the `rivcbjdd` estimate using the code:

```
1 SYS=tfest(D,Mdd)
```

it converges to this true optimum with exactly the same parameter estimates as `rivcbjdd`, once again confirming that it originally converged on a false optimum.

Finally, Fig.11 shows the results of *Monte-Carlo Simulation* (MCS) analysis based on the initial estimation results for the model (2), as shown in the verbatim panel above Fig.3. This analysis is based on 10,000 stochastic realisations, using the estimated covariance matrices for both the continuous-time model and the associated discrete-time noise model. As required the total 95% confidence bounds, including the noise effects, encompass a large proportion of the measured data points, while the confidence bounds based only on the parametric estimation error (dark shading) are narrow. The mean responses from both are very similar, however, so the latter red line obscures the blue line beneath it.

4. USA Unemployment Rate, 1983-2015

The input-output data for this three-input example are plotted in figures 5 and 6 on the next page, where the former only shows two inputs because the third is a unitary vector (all ones) introduced to estimate and account for the considerable *Initial Condition* (IC) on the measured output data [see 8, particularly section A.1.1.3]. This output is the unemployment rate, measured as a percentage, which is plotted in Fig. 6 and shown, once again, as the black dotted line.

The inputs are *Relative Government Investment* (RGI) and *Relative Private Investment* (RPI), where these are both made relative to the *Gross National Product* (GNP) by simple division. In this way, the continuing upward movements of all three macro-economic measures are eliminated, producing the inputs plotted in Fig. 5 that amplify the movement in both the investment variables as they react to changes in the macro-economy. And this, in turn, allows for more straightforward and revealing model identification and estimation¹.

Earlier data such as those in these plots but over the period 1945 to 1999, were first analyzed by [11]. In the present note,

¹i.e. *identification* of a statistically identifiable model structure, followed by the *estimation* of the statistically well-defined parameter estimates that characterise this structure.

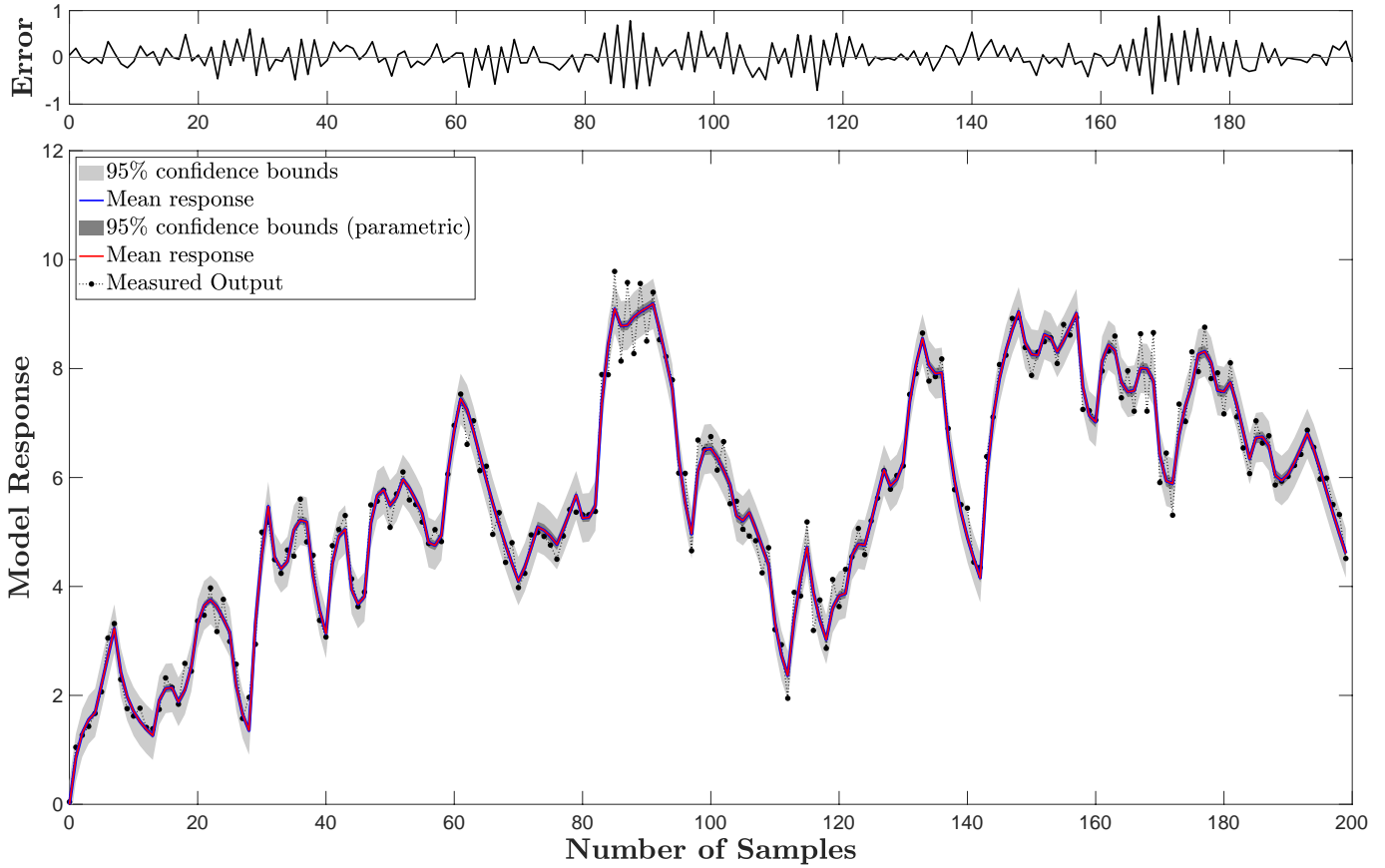


Figure 4: MCS simulation results for the second order simulated data example based on the rivcbjdd estimated model.

however, the focus is on modelling the dynamic behaviour of these variables between 1983 and 2015, based on later data, so revealing the serious troubles in the US economy and elsewhere that occurred after the turn of the Millenium, including the great crash of 2008 [as considered fully in chapter 4 of 10].

The call to rivcbjdd in this case is similar to that in the first example, except that the three input model structure is now $[1 \ 2 \ 0 \ 8 \ 0; 1 \ 2 \ 0 \ 8 \ 0; 1 \ 2 \ 0 \ 8 \ 0]$, where the noise model is identified by the ivarmaid tool in CAPTAIN as an AR(8) process. The full model produced in this manner takes the following form:

$$x(t) = \frac{136s - 17.06}{s + 0.218} RGI(t) + \frac{-9.847s - 53.8}{s + 0.2809} RPI(t) + \frac{-16.83s + 13.8}{s + 0.276} U_i(t)$$

$$y(k) = x(k) + \xi(k)$$

(3)

where U_i is the unitary vector required for IC estimation; $x(t)$ is the modelled unemployment rate and $x(k)$ are its quarterly sampled values; $y(k)$ is the quarterly measured unemployment rate, contaminated by the noise $\xi(k)$. The random input to the AR(8) noise model is a white noise series $e(k)$ with variance $\sigma^2 = 0.037$, where the autoregressive AR(8) noise model with its parameters c_1 to c_8 is not shown here to save space. Note that this noise model is not clearly identified: e.g. in [10] an AR(5)

process is identified using the aic tool but this is reduced to an AR(1) process because the the other parameters are not well identified despite the aic result. The ivarmaid identification is used here because it has a statistically significant parameter at lag 8 and this higher order noise model provides a good test for rivcbjdd.

The verbatim estimates and standard errors computed from the covariance matrix, Psd , are returned as follows from rivcbjdd and, as we see from Fig.6, the model explains the data well with $R_7^2 = 0.975$; and despite the additional parameterization required for IC estimation, the parameter estimates are all reasonably defined in statistical terms.

```

1 [Mdd, Mnd, edd, eefd, Pard, Psd, Pnd, yh, Ym, Mn, D]=...
2 rivcbjdd([yn(1:lg) ur1n(1:lg) ur2n(1:lg) ones(size(yn(1:lg)
3 [1 2 0 8 0; 1 2 0 8 0; 1 2 0 8 0], [0.0001 0.25 1 0], -0.1, 0);
4 [Pard; sqrt(diag(Psd))'])
5 136.0 -17.057 0.2176 -9.8472 -53.797 0.2809 -16.632 13.802 0.21
6 11.1 11.221 0.0947 6.2954 5.9871 0.065 3.351 2.206 0.0

```

This result means that the model is a reliable vehicle for studies into the behaviour of the USA economy in these terms.

On the other hand, it is worth noting that the common denominator alternative to this model, which is estimated very simply and reliably by the application of the srivcarma tool in CAPTAIN, has two less parameters. Moreover it has very similar response characteristics and performs in a very similar



Figure 5: Inputs for the USA unemployment data example.

manner to the above model for the applications used in chapter 4 of [10]. In this alternative study, therefore, it provided a quicker (0.09 *cf* 13.7 seconds) and quite acceptable alternative to `rivcbjdd`. The reason for this is revealed if the estimated time constants of the models are considered: in this `rivcbjdd` case, the time constants are estimated as $\hat{T}_1 = 1/0.2176 = 4.6$ and $\hat{T}_2 = 1/0.2809 = 3.6$ years; while the common denominator `srivcarma` case it is estimated as $\hat{T} = 1/0.215 = 4.65$ years. These are clearly quite close and the differences are not statistically significant.

This is a good example of the ‘art and craft’ required for *Data-Based Mechanistic* (DBM) modelling [see 10]; namely, for any defined task, choosing the most suitable tool for the job. In this case, the rapid and robust action of the `srivcarma` tool, combined with its ability to generate a similar response to the `rivcbjdd` estimated model, make it more suitable for tasks such as model updating and forecasting regularly, say every quarter, over an extended period of time. But if the objective is the estimation of the model parameters, or parameters derived from these, then `rivcbjdd` or `tfest` may be more appropriate tools (see the next section 5). for instance, in this example, it is interesting to investigate which type of investment, RGI or RPI, is most effective in reducing unemployment, thus necessitating the best possible statistical estimates of derived parameters, such as the steady state gain and time constant of the two TF models.

As shown in Fig.6, the `tfest` results are not particularly good, with a poorer explanation of the unemployment rate series. This is partly an IC estimation problem because the response of the model computed by the Matlab `lsim` routine and shown as a red line is clearly not accounting for this. However, if the `compare` routine is used instead, then the explanation of the IC effect is similar to that achieved by the `rivcbjdd` analysis because `compare` corrects for any IC errors. Why the `tfest` estimation is not exploiting the presence of the U_i input in this example is not clear, as is its inability to do as well as `rivcbjdd` over the rest of the series. Note also that because `tfest` produces a very similar model to `rivcbjdd` if initiated with the latter’s estimates, one presumes that the poorer performance is because, once more, it has converged on a false minimum, possibly because of the identifiability issues mentioned above. This is, perhaps, reflected in the returned covariance matrix, which has extremely large elements producing very wide standard errors on the parameters.

There is another alternative to estimating a model with `tfest` by accepting that the use of `compare` will correct for the IC estimation and so only estimating a two-input model without the U_i component. This produces a marginally worse explanation of the data although, of course, the model is then more parsimonious with three less parameters. Unfortunately, this does not improve the returned covariance matrix which again suggests very large uncertainty in the estimates.

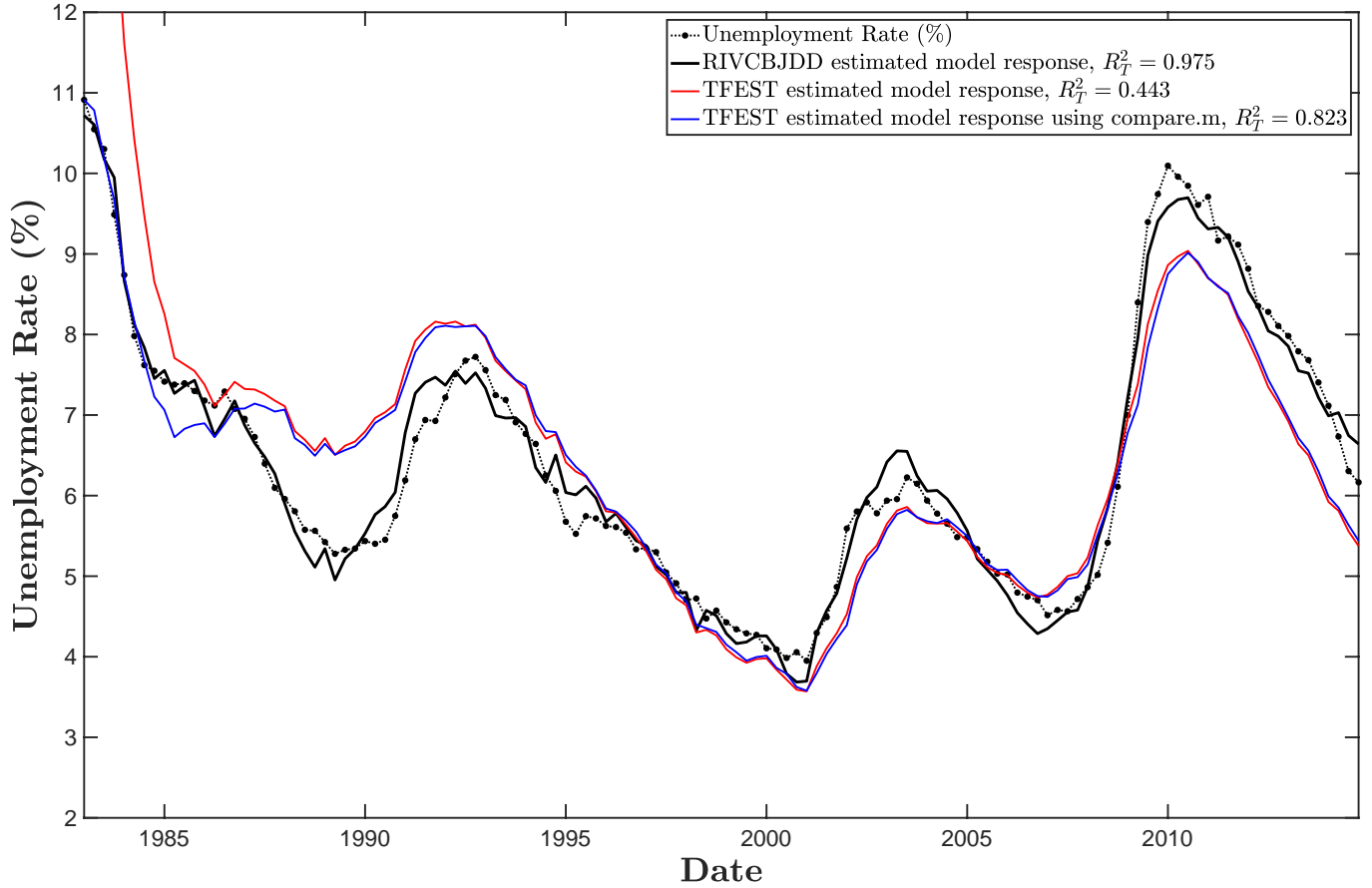


Figure 6: RIVCBIJDD estimation results for the USA unemployment data example.

5. Global Surface Temperature 1856-2015

The input-output data for this three-input example are plotted in figures 7 and 8 where the former only shows two inputs because, once again, the third is a unitary vector required to account for the ICs on the measured global temperature anomaly series, as plotted in Fig. 8 on the next page. The first input in the top panel of Fig. 7 is the *Total Radiative Forcing* (TRF), defined as the difference between the incoming and outgoing radiation at the top of the atmosphere². This is the sum of all the components that affect the radiation falling on the Earth, including the atmospheric CO₂ that is a primary anthropogenic stimulant of global warming. But the TRF is the aggregation of all these positive *and negative* components where, for example, the relatively large magnitudes of the sharp downward spikes (that lower the surface temperature) are due to volcanic activity.

The second input is the *Atlantic Multi-decadel Oscillation* (AMO), which is a less well known variable identified by [6] and usually defined from the patterns of *Sea Surface Temperature* (SST) variability in the North Atlantic, once a trend has

been removed. This is not often considered as an input in climate modelling but figures prominently in such a role for the revealing DBM modelling analysis described in [10].

Once again, the call to `rivcbjdd` in this example is similar to that in the previous example, except that the three input model structure is now identified with an AR(1) residual noise process. The full hybrid model produced in this manner takes the following form:

$$x(t) = \frac{0.0306s + 0.0196}{s + 0.0273} TRF(t) + \frac{0.359s - 0.0236}{s - 0.00574} AMO(t) + \frac{-0.407s + 0.00131}{s + 0.0618} U_i(t)$$

$$y(k) = x(k) + \xi(k)$$

$$\xi(k) = \frac{1}{1 + 0.236z^{-1}} e(k) \quad e(k) = \mathcal{N}(0, 0.0060)$$
(4)

and the verbatim copy of the estimation results is as follows:

```
1 [Pard;sqrt(diag(Psd))']
2 0.0306 0.0196 0.0273 0.359 -0.0236 -0.00574 -0.407 0.00131
3 0.0234 0.00670 0.0114 0.0438 0.0168 0.0483 0.0479 0.00647
```

The explanation of the data is good, with $R_T^2=0.924$ and the *Equilibrium Climate Sensitivity* [ECS, see e.g. 10] derived from the model is 3.7°C multiplied by the steady-state gain, $G =$

²See <https://www.metoffice.gov.uk/binaries/content/assets/metofficegovuk/pdf/research/ukcp/ukcp18-guidance-representative-concentration-pathways.pdf>

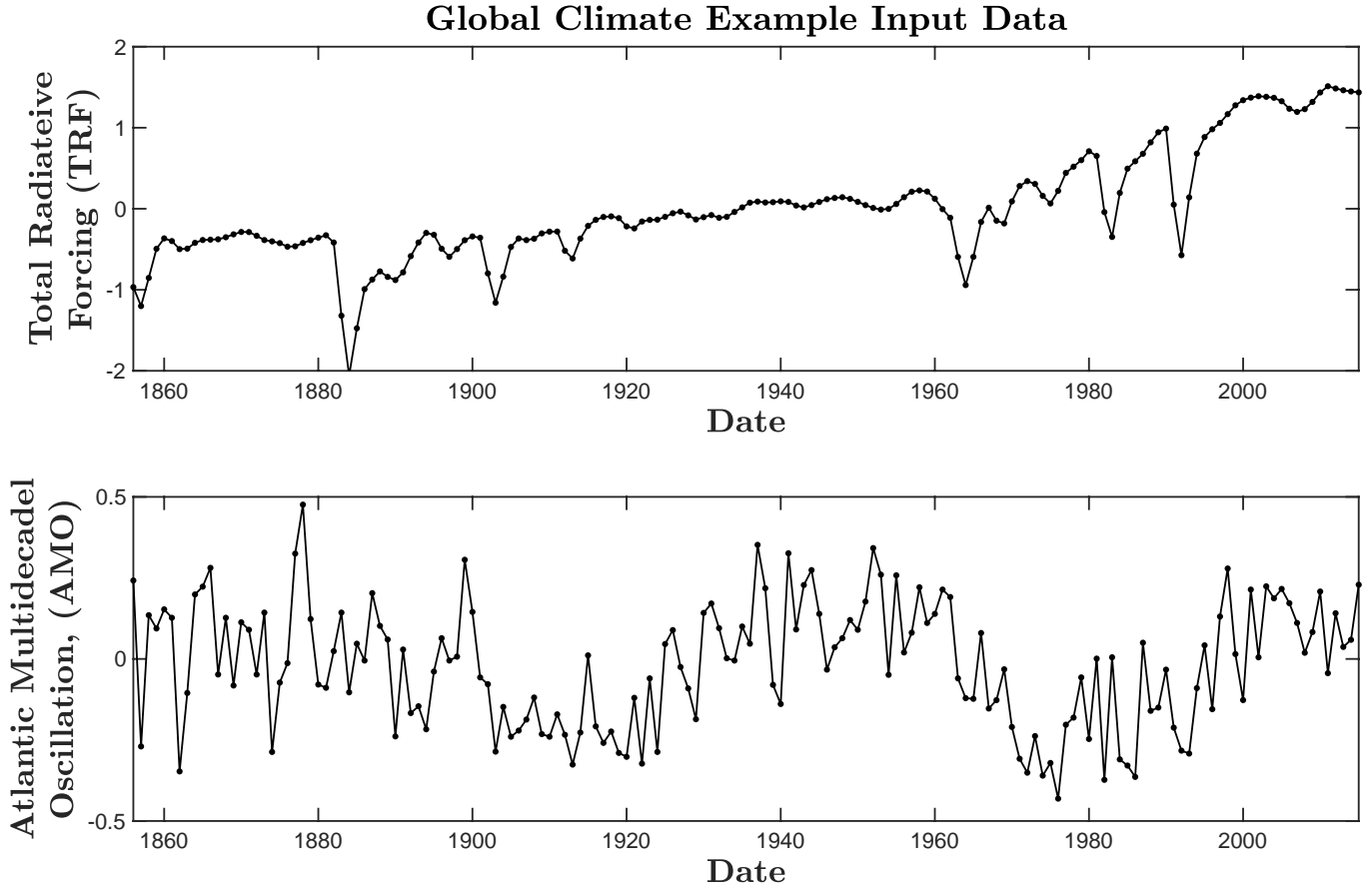


Figure 7: Inputs for the global climate data example.

0.0196/0.0273, of the first TF in (4), yielding $ECS=2.66^{\circ}C$, which is acceptable from a climate science standpoint. While this suggests that the model is adequate, it is not necessarily the best possible model because there are two poorly defined parameters, with their standard errors much larger than the estimated value (see above verbatim results). This is important because a major objective in this example is the best estimation of this ECS and the associated time constant of the TF. In fact a simpler model, with all its parameters statistically well-defined, can be identified and estimated using Simulink-based model [for details, see 10] and it takes the form :

$$x_1(t) = \frac{b_{10}}{s + a_{11}} u_1(t) \quad (i)$$

$$x_2(t) = G u_2(t - 1) \quad (ii)$$

$$x(t) = x_1(t) + x_2(t) \quad (iii)$$

$$y(k) = x(k) + \xi(k) \quad (iv)$$

$$\xi(k) = \frac{1}{1 + c_1 z^{-1}} e(k) \quad e(k) = \mathcal{N}(0, \sigma^2) \quad (v)$$

(5)

where G is a simple static gain, effectively the steady state gain estimated directly, and the parameters estimates plus their esti-

mated standard errors, in parentheses, are as follows:

$$\begin{aligned} \hat{a}_{11} &= 0.062(0.007); \quad \hat{b}_{10} = 0.036(0.003); \\ \hat{G} &= 0.308(0.040); \quad \hat{IC} = -0.362(0.055) \\ \hat{c}_1 &= -0.282(0.077); \quad \sigma^2 = 0.0065 \end{aligned} \quad (6)$$

Although the $R_T^2 = 0.911$ is a little less in this case than the rivcbjdd estimated model (4), the $ECS=3.7 * 0.036/0.062 = 2.15^{\circ}C$ makes sense in climatic terms and is now more precisely estimated. Moreover, it is obtained with only 4 estimated parameters, one of which, \hat{IC} , is the initial condition on the integrator in the Simulink model, with the model for the AMO input reduced to a static gain with a one year time delay. Consequently, the model (4) can be considered a step towards the final identification of this much simpler and statistically better defined model.

In this case, the tfest estimated model returned by rivcbjdd and plotted as the red line in Fig. 8 is visually not very good, mainly because because the IC is not being estimated well. This is a similar situation to that in the previous USA unemployment example and, as in that case, a much better explanation of the data is obtained if the compare routine is used to compute the model output, rather than the lsim routine used in rivcbjdd. And the improved performance is, once again, due to the IC being estimated within the compare routine. On the other hand, this

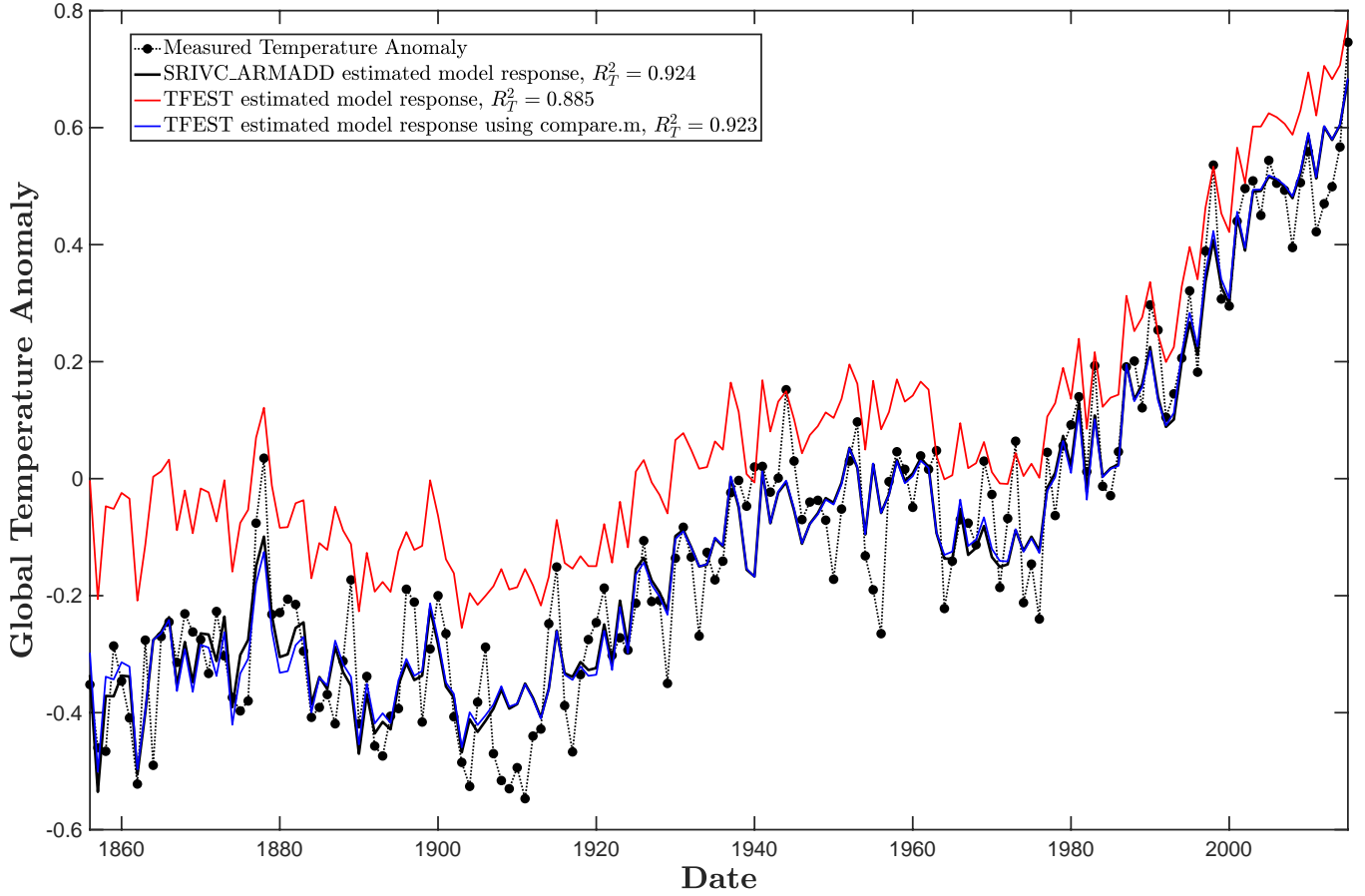


Figure 8: RIVCBJDD estimation results for the global climate data model.

tfest model yields a very large and unlikely value of 6.16°C for the ECS and the model would probably be rejected on these grounds.

One advantage of tfest is that it can estimate models with simple static gains and, given the Simulink-based model (5) above, it makes sense to consider the tfest estimation of a similar model. The best model using this approach takes the form shown verbatim below, which is the same form as the (5) but without the unity time delay on the AMO variable (including this delay produced worse results). Using compare, which is now essential in this case because lsim cannot account for the IC, this has $R_T^2 = 0.905$, not quite as good as the above Simulink-based model (6). However, the model parameters appear statistically well-defined and the ECS estimate of 2.13°C is very similar to that obtained from the estimates in (6). Clearly, therefore, the ability of tfest to estimate a model with a simple gain component is attractive and it is important that the TF estimation procedures in CAPTAIN, including rivcbjdd, should be upgraded in this regard in any future development.

```

1 From input "u1" to output "y1":
2   0.03787
3   -----
4   s + 0.0658
5 From input "u2" to output "y1":
6   0.2878

```

Finally, rivcbj and srivcarma estimation both fail to produce acceptable models in this example, with much reduced explanatory ability ($R_T^2 = 0.798$).

6. An Alternative Climate Model

The climate model in the previous section assumes that the quasi-periodic component in the globally averaged surface temperature is related to the AMO variable. An alternative hypothesis is to consider that it may be an oscillatory mode triggered by the TRF. The DBM modelling reported in [10] shows that this is possible if a second input is introduced in the form of the TRF substantially delayed by a τ years, i.e.,

$$x_1(t) = \frac{b_{10}s + b_{11}}{s + a_{11}}u(t) \quad (i)$$

$$x_2(t) = \frac{b_{20}s + b_{21}}{s^2 + a_{21}s + a_{22}}u(t - \tau) \quad (ii)$$

$$x(t) = x_1(t) + x_2(t) \quad (iii)$$

$$y(k) = x(k) + \xi(k); \quad (iv)$$

$$\xi(k) = \frac{1}{1 + c_1z^{-1}}e(k) \quad e(k) = \mathcal{N}(0, \sigma^2) \quad (v)$$

(7)

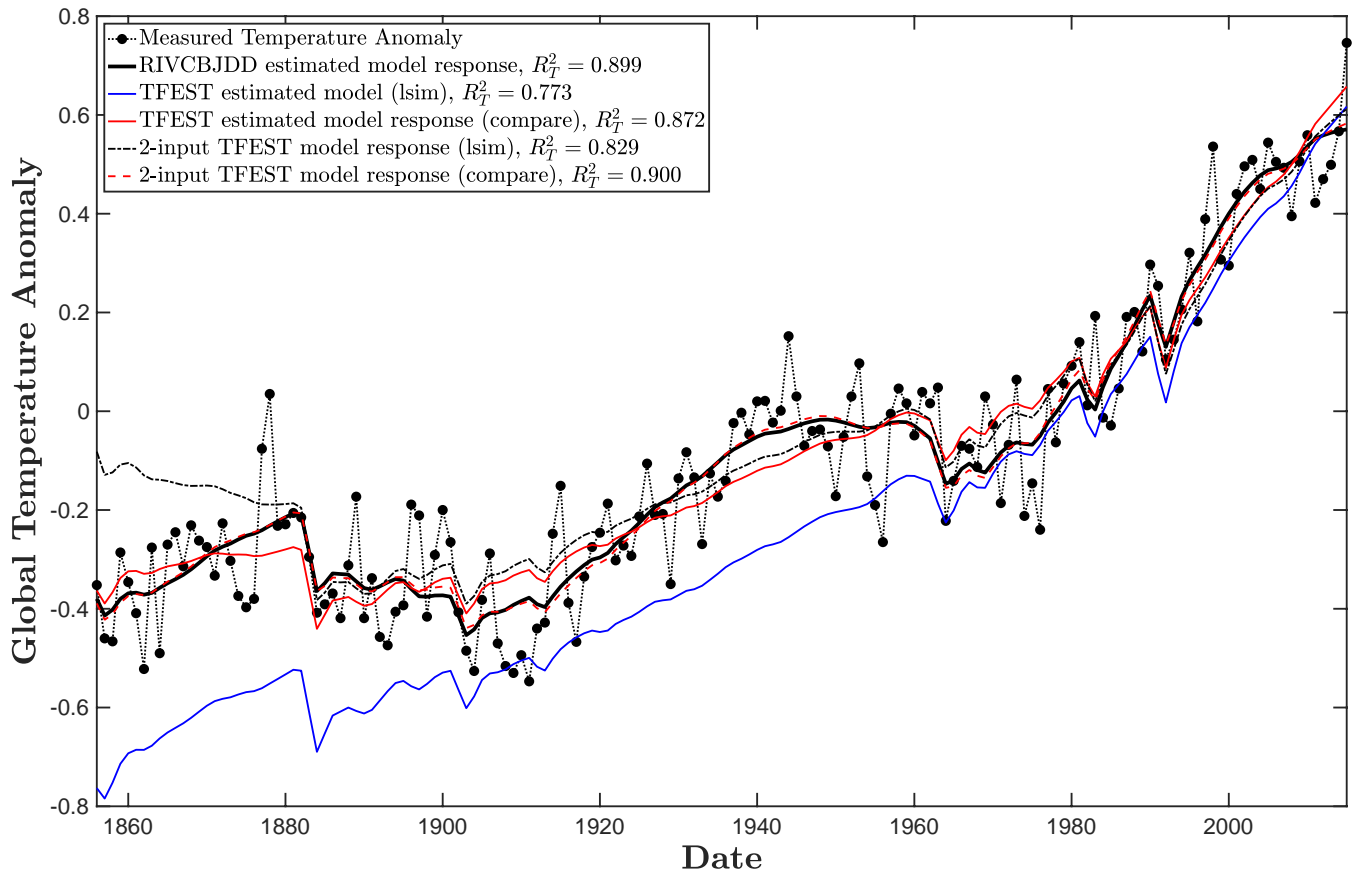


Figure 9: RIVCBJDD estimation results for the second global climate data example.

```

1 [snip(SYS1.par', '0'); snip(sqrt(diag(SYS1.CovarianceMatrix)), '0')]
2 0.0847 0.02957 0.0465 -9.32e-05 0.00179 0.0125
3 0.028 0.0031 0.0040 0.00036 0.0082 0.0010

```

7. Dispersion of a Pollutant in a Wetland area

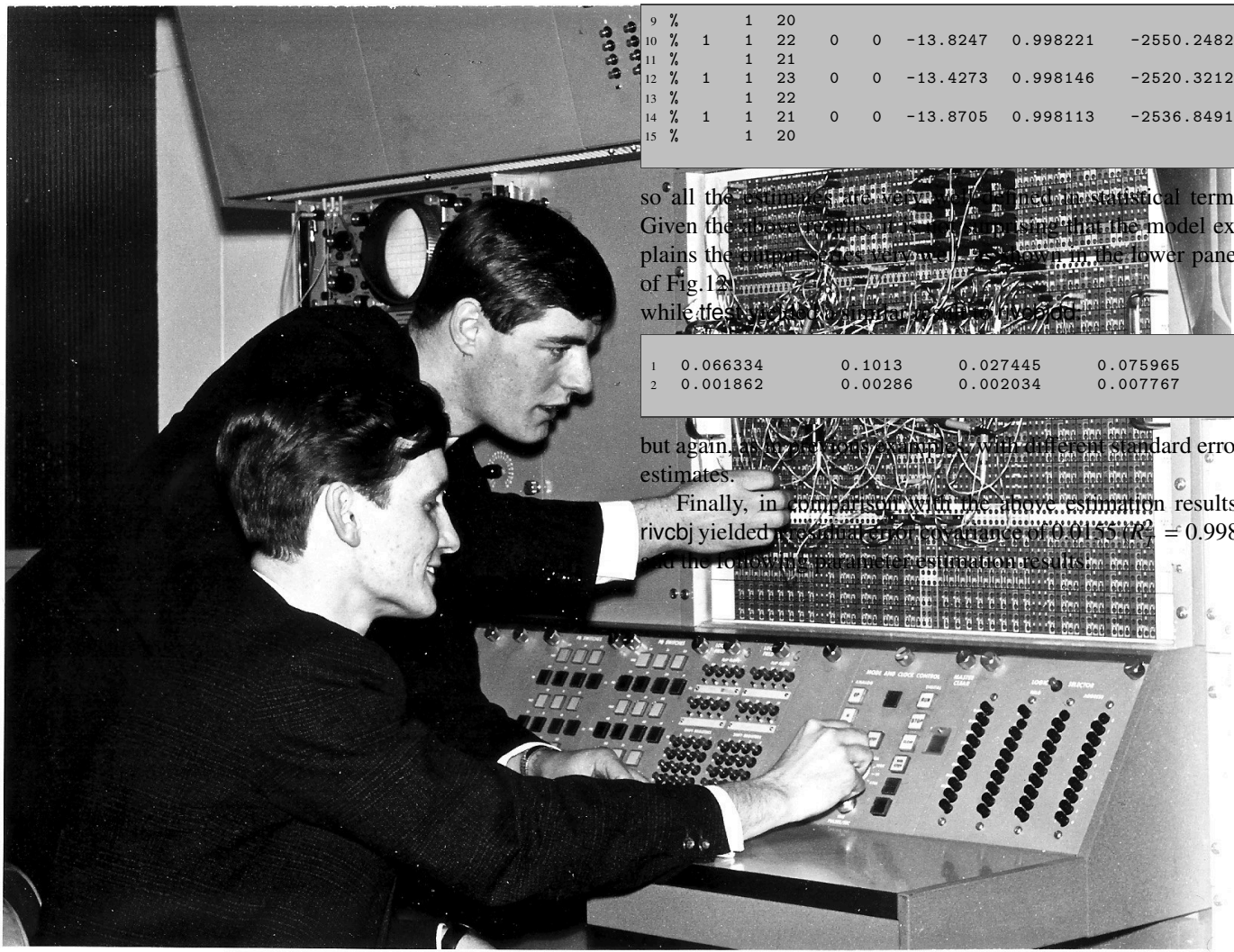
This example considers the analysis of bromide tracer concentrations collected at locations 2-X, 2-Z and 6-Y shown in Fig.11, as discussed fully in [5]. Here, the locations of the first two upstream sites are considered as inputs with samples $u_1(k)$ and $u_2(k)$ with 6-Y as the downstream output with samples $y(k)$. The sample size is 433 and the sampling interval is $dt = 2$ minutes. Of course, as this is a wetland area and not a narrow stream, not all of the dye at the inputs necessarily reaches the output location but the model should provide some insight into what is happening from a physical standpoint.

One limitation of the present `rivcbjdd` tool is that it does not allow inherently for the user to specify input time delays, which are important in this application. Consequently, the user must identify these prior to estimation and adjust the input time series accordingly. In this case, such identification employed the `rivcbjid` tool in CAPTAIN and the verbatim output showing the best identified common denominator models of this kind is given at the top of the next page. Clearly, all these models explain the data well with $R_T^2 > 0.9980$, so it is difficult to choose between them. In this case, the model best identified by the YIC was selected, i.e. on lines 14 and 15, with time delays of 21 and 20 samples (42 and 40 minutes). The input data were then adjusted accordingly to produce the equivalent zero-delay data shown in the middle panel of Fig.12. Using these adjusted

Note that these are the results returned by `tfest`, so it is not clear if an unreported noise model was included in estimating the covariance matrix and standard errors. If not, then this would be a limitation of the model.

In this example, therefore, that the `tfest` estimation has advantages in relation to `rivcbjdd`: it is more parsimonious with, therefore, better defined estimates of its parameters; and it could be used as a quick estimation of the model instead of the more complicated formulation of the Simulink model. On the other hand, a noise model is required for statistical optimality and the accuracy of estimated uncertainty in the model parameters. This is essential for many applications: for example, the Simulink model is used in [10] for scenario, forecasting and management studies, all of which require the presence of the noise model and are also assisted by the practical flexibility of the Simulink model.

Finally, it is not surprising that, in this example, `rivcbj` and `srvcarma` both completely fail to explain the data and so there is no alternative common denominator model.



```

9 % 1 20
10 % 1 1 22 0 0 -13.8247 0.998221 -2550.2482 -6.520
11 % 1 21
12 % 1 1 23 0 0 -13.4273 0.998146 -2520.3212 -6.479
13 % 1 22
14 % 1 1 21 0 0 -13.8705 0.998113 -2536.8491 -6.461
15 % 1 20

```

so all the... Given the... explains the... while the...
 In statistical terms, the model explains the... the lower panel of Fig. 12... while the...
 but again... with... standard error estimates.

```

1 0.066334 0.1013 0.027445 0.075965
2 0.001862 0.00286 0.002034 0.007767

```

but again... with... standard error estimates.

Finally, in comparison with the above estimation results, rivcbj yielded a variance of 0.015 (R² = 0.9981) and the following parameter estimation results:

```

3 0.0635 0.0956 0.0236 0.0673
4 0.002639 0.0042358 0.0026542 0.010498

```

Figure 10: The Simulink diagram of a DBM model based on (7), as used in [10], is shown the top, with a connected diagram for an AMO model below it.

data for rivcbjdd estimation, with an aic identified AR(9) model for the noise, produced the following model:

$$x(t) = \frac{0.0635}{s + 0.0956}u_1(t - 42) + \frac{0.0236}{s + 0.0673}u_2(t - 40) \quad (8)$$

$$y(k) = x(k) + e(k)$$

The parameter estimates with their standard errors are obtained as shown verbatim below:

```

1 [Pard;sqrt(diag(Psd))']
2 % 0.063509 0.095617 0.023588 0.067336
3 % 0.002639 0.0042358 0.0026542 0.010498

```

```

1 [thc,Mc,Mnc,D,statsc,ec,Rrc,eefc,varc,Psc] =...
2 rivcbjid([y u1 u2],[1 1 1 15 15 0 0;1 1 1 25 25 0 0],2,...
3 [0.00001 2 1 0],[-0.01]);
4 % ----- BEST 20 models (Rt2) -----
5 % den num del AR MA YIC Rt2 BIC AIC
6 % 1 1 23 0 0 -13.5667 0.998262 -2554.2172 -6.5440
7 % 1 1 21
8 % 1 1 22 0 0 -13.8649 0.998246 -2562.5243 -6.5351

```

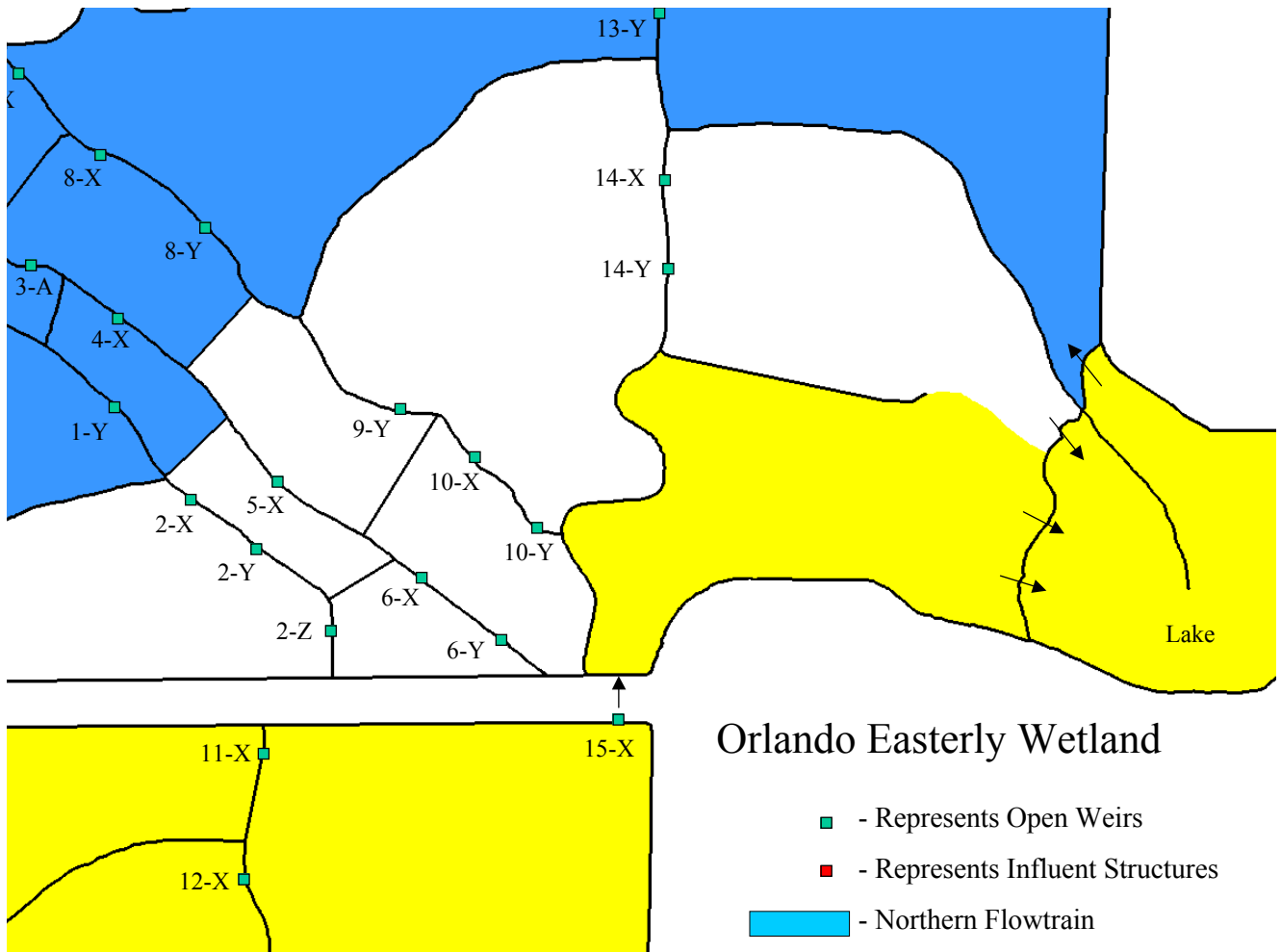


Figure 11: Map of the Orlando Easterly Wetland area (provided by [5]) showing the sampling locations 2-X, 2-Z and Y-6 considered in this example.

```

1 [M(1).f(2:end) M(1).b M(2).b;sqrt(diag(Ps))']
2 % 0.089734 0.061521 0.028942
3 % 0.001997 0.0017273 0.0009037

```

demo example ctmoddemo for the identification and estimation of continuous-time TF models. The results below are extracted from this demo example.

The final rivcbjdd estimation is based on a simplified model and takes the form:

$$\begin{aligned}
 x(t) &= \frac{0.9895}{s^2 + 1.238s + 7.238} u_1 t + \frac{95.28}{s^2 + 27.83s + 79.06} u_2 t \dots \\
 &\quad + \frac{10.06}{s^2 + 6.913s + 46.5} u_3 t \\
 y(k) &= x(k) + \xi(k)
 \end{aligned}
 \tag{9}$$

where the noise is identified and estimated as an AR(11) process. This model is compared with srivcarma and tfest estimated models of the same form in a plot generated by the Matlab compare routine. An improved version of this plot is shown below in Fig.13 at the top of the next page, where we see that the explanation of the measured output is reasonable but not outstanding, probably because the winding system is nonlinear (which is not investigated in the demo). The parameter estimates and standard errors are produced as before:

In this case, therefore, there is no justification for the different denominator model. But, of course, it was necessary to consider this in order to eliminate the possibility.

8. A Three Input-Winding Process

This demo is a concerned with a more conventional systems engineering system: a three input-three output winding process. The model identification and estimation analysis is based on only the first output and uses 3-input, single output data collected by [2]. In general, winding systems like this are continuous, processes with some nonlinear elements. They are encountered in a wide variety of industrial plants, such as rolling mills in the steel industry, plants involving web conveyance including coating, paper making and polymer film extrusion processes. In this case, the data come from experiments associated with a winding pilot system. The rivcbjdd identification and estimation results are generated in a new CAPTAIN

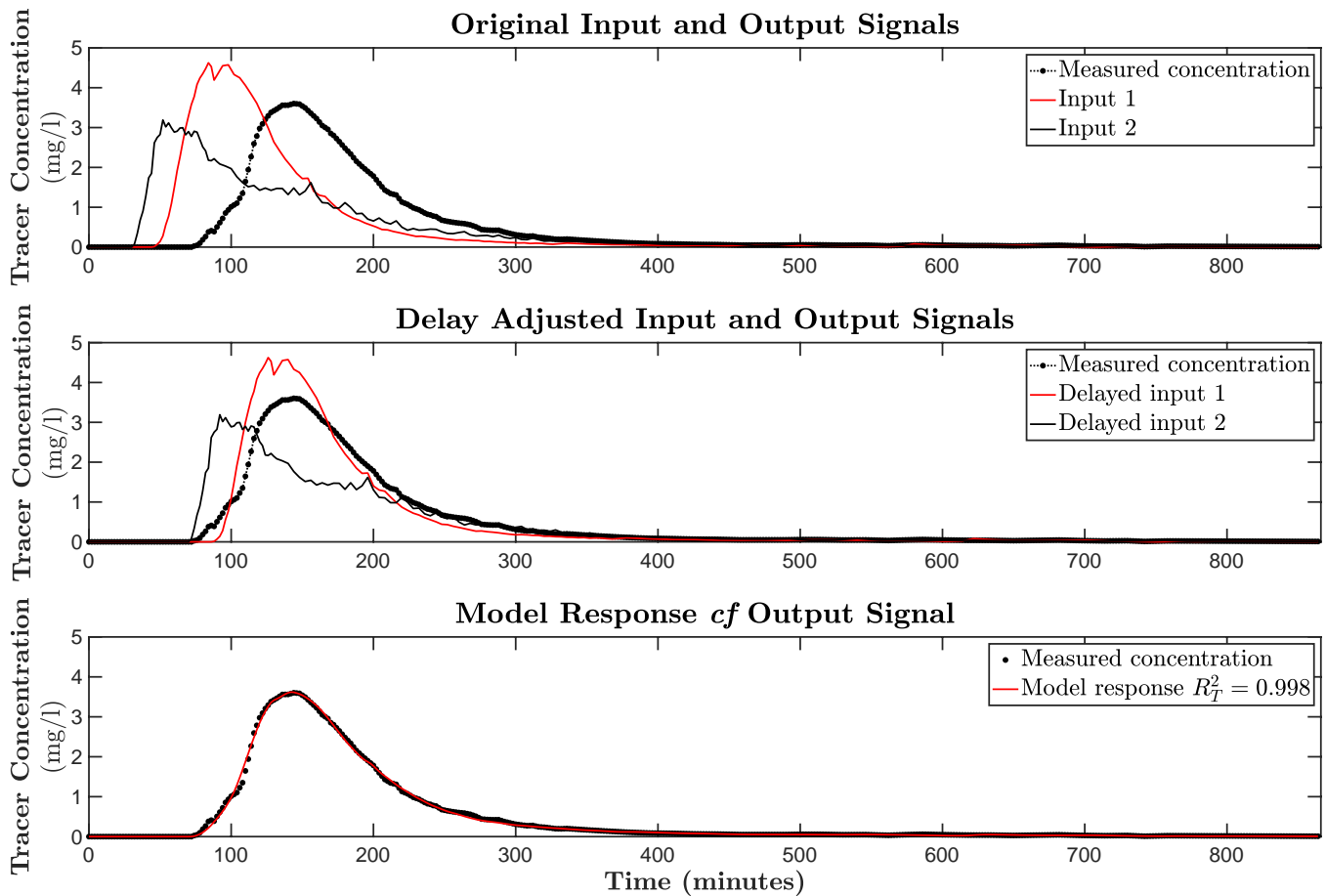


Figure 12: Original Data (top panel), time-delay adjusted data (middle panel) and rivcbjdd estimation result for the two input model (lower panel).

```

1 [Pard; sqrt(diag(Psd))']
2 0.98947 1.238 7.2381 95.279 27.831 79.062 10.061 6.9135 46.504
3 0.13827 0.2806 0.5654 14.690 4.315 11.696 3.585 2.8301 14.115

```

9. Conclusions

The results presented in this Technical Note suggest that the new rivcbjdd tool that has been developed for the CAPTAIN Toolbox works well under standard conditions, as exemplified by the first two simulation examples. Indeed it seems, at first sight, to have some advantages in this regard over the alternative, but related, tfest tool in the Matlab Identification Toolbox, although further evaluation is required before any conclusions about its relative performance will be possible. Most importantly, it overtly incorporates an associated noise model and, therefore, has built-in statistical and stochastic simulation advantages. However, it does not function so well when applied to data that have pronounced initial conditions, as in the second three examples. Here, it is able to explain the data well but the necessary inclusion of an additional unitary input to account for the initial conditions means that there are additional parameters to estimate, so that the parameters are not as well-defined statistically as when the model is estimated using an alternative Simulink model.

Currently, rivcbjdd estimation has the following three main limitations.

1. The first is the much longer computation time than tfest, which is clearly due to this Matlab routine being com-

Once again, however, the rivcbjdd estimation yields better results than tfest and common denominator srivcarma estimation exercises based on the same identified model form.

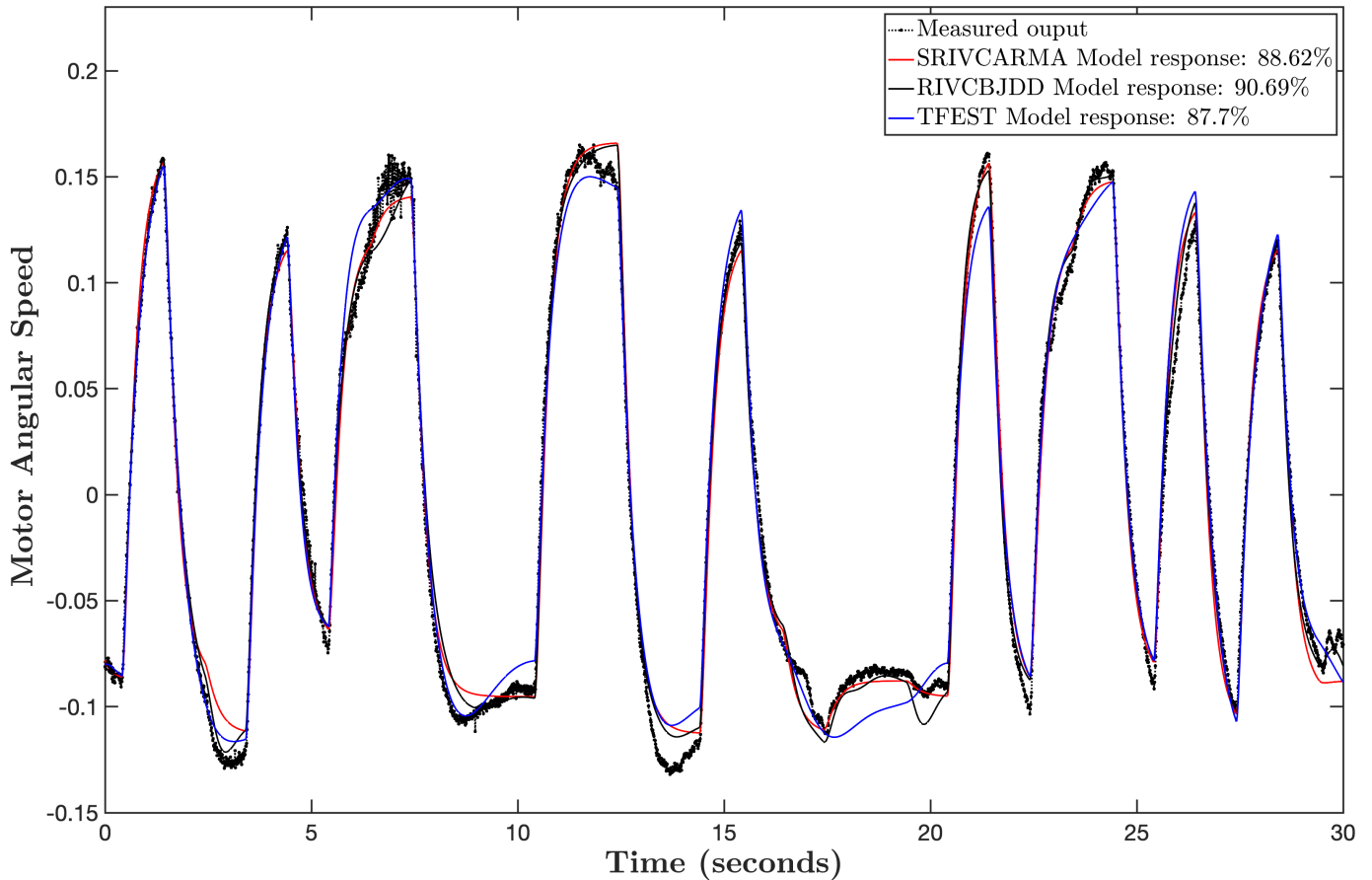


Figure 13: Improved version on the plot produced by the compare routine in the CAPTAIN ctmoddemo demo.

piled using Matlab C++ and it suggests³ that rivcbjdd would be similarly rapid if coded and compiled in the same manner. Unfortunately, the CAPTAIN Toolbox team of two no longer has the resources to pursue this possibility. But the aim of DBM modelling is the identification and estimation of an optimal model that incorporates a noise model to permit uncertainty analysis and can be interpreted in physically meaningful terms. In this context, therefore, the rivcbjdd model is more satisfactory than tfest and so, it is felt, worth waiting the longer computational time.

2. The second disadvantage is that, at present, it cannot handle models where there are input delays, so that the user must adjust the input data series to remove such delays.
3. The final disadvantage is its inability to handle a situation, such as that in the first global climate model, where one of the component TFs can be replaced by a simple static gain.

Despite these limitations, which it is hoped will be corrected in future upgrades, the rivcbjdd tool explains the measured data

well: in most of the examples considered here better than tfest, with a statistically well-defined model and reliable measures on the associated uncertainty in the model parameters.

Finally, note the rivcbjdd tool is now available in CAPTAIN. However, it is still at β -testing stage and has not yet been fully 'CAPTAINized'. Consequently, some errors will not be captured and reported during its use, so it would help if any problems encountered in this, or any other, regard are reported, in order to aid in the final development.

³see for example the discussion in <https://stackoverflow.com/questions/20513071/performance-tradeoff-when-is-matlab-better-slower-than-c-c>.

10. Appendix: CAPTAIN ‘Help’ for RIVCBJDD

[11] Young, P. C. and Pedregal, D. J. (1999). Macro-economic relativity: government spending, private capital investment and unemployment in the USA 1948-1998. *Structural Change and Economic Dynamics*, 10:359–380.

```

1 function [Mdd,Mnd,edd,eefd,Pard,Psd,Pnd,yh,Ym,Mn,DD]=rivcbjdd(Z,nn,flags,c,cf,nnr,Tol,Iter);
2 % RIVCBJDD Optimized continuous-time MISO TF with different denominators
3 %
4 % [Mdd,Mnd,edd,eefd,Pard,Psd,Pnd,yh,Ym,Mn,DD]=rivcbjdd(Z,nn,flags,c,cf,nnr,Tol,Iter);
5 %
6 % BETA TEST version - please report any problems encountered
7 %
8 % Z: I=0 data, [Y,Ui,...,Unu] where Y and Ui are column vectors (*)
9 % nn: Component model structures where each of nu rows represents a
10 % SISO model associated with each input: [na, nb, ntd, nc, nd] (*)
11 %   na: denominator, nb: numerator, ntd: time delay
12 %   nc: noise model denominator (AR), nd: numerator (MA)
13 % CURRENT BETA TEST FNC REQUIRES ZERO TIME DELAYS (ndt=0)
14 % Remove estimated time delays from Z before using fnc
15 % (this limitation will be removed in a later release)
16 % flags: Additional parameters vector [Mit, Ni, dt, ddt, cd, Rc]
17 %   Missing value or -1 implies use default value in brackets
18 %   (1) Ni<1: convergence criterion to compare with quadratic norm
19 %       of parameter vector change between RIV iterations (1e-4)
20 %   Ni>1: Number (maximum) of SRIV iterations
21 %   (3) dt: Sampling interval for continuous-time estimation (1)
22 %   (5) cf: Constant (1) or adaptive (0-default) pre-filter flag
23 %   (6) Rc: Block (0-default) or recursive algorithm (1-slower)
24 % c: Prefilter polynomial parameter (1)
25 %   Scalar <=0: Polynomial created automatically from pole value
26 %   e.g. zero to utilise multiple integrators, <0 for
27 %   stable filter (chosen so that 1/C roughly matches
28 %   the bandwidth of the system being estimated) - note
29 %   relatively insensitive to this parameter, so it is
30 %   useful as default (e.g. -0.1, -0.05)
31 %   Polynomial: Polynomial with order 'na'
32 %   1: Estimate discrete-time filter and convert to continuous-time
33 %   Normally C<=0 is best for fast and C=1 for coarse sampled data
34 %   (or, if fast sampled, can use C=1 with ddt set at a more coarse
35 %   subsampling rate for initial discrete-time estimation)
36 % cf: Cost function: prediction error (1-default) or response error (0)
37 % nnr: Model structure for backfitting starting estimate (1)
38 %   [na,nb(1:nu),ntd(1:nu),nc,nd]: common denominator model
39 %   na: denominator, nb: numerators, ntd: time delays
40 %   nc: noise model denominator (AR), nd: numerator (MA)
41 %   e.g. use RIVCBJDD to find suitable values for nnr
42 %   1: automatically generate common denominator model nnr from nn
43 %   0: use SISO models based on nn above
44 % Tol: 'TolFun' setting for lsqnonlin.m (0.001)
45 % Iter: 'MaxIter' setting for lsqnonlin.m (200)
46 %
47 % Outputs
48 % Mdd: Estimated model object (SID Toolbox required) or empty
49 % Mnd: ARMA noise model (SID Toolbox required) or empty
50 % edd: system model output errors (y=fit+edd)
51 % eefd: Residuals from noise model (stochastic model residuals)
52 % Pard: Covariance matrix for I=0 parameter estimates
53 % Psd: Covariance matrix for noise model parameters
54 % yh: Estimated model output response
55 % Ym: Estimated responses of each transfer function to their associated
56 % inputs (matrix dimension: length(Y), nu)
57 % Mn: redundant - to be removed at next update
58 % DD: Data object in IDDATA form
59 % See also CTMDDDEMO, RIVCDD, RIVCBJ, RIVCBJDD, SRIVCARMA
60 % Copyright 2025 by Lancaster University, United Kingdom
61 % Peter Young and James Taylor

```

Note that the above ‘Help’ is slightly different to that currently in CAPTAIN: it is the version associated with an update of the rivcbjdd tool used in this TN.

- [1] Astrom, K. J. (1980). Maximum likelihood and prediction error methods. *Automatica*, 16:551–574.
- [2] Bastogne, T., Noura, H., Sibille, P., and Richard, A. (1998). Multivariable identification of a winding process by subspace methods for a tension control. *Control Engineering Practice*, 6(9):1077–1088.
- [3] Box, G. E. P. and Jenkins, G. M. (1970). *Time Series Analysis Forecasting and Control*. Holden-Day: San Francisco.
- [4] Ljung, L. (1999). *System identification. Theory for the user*. Prentice Hall, Upper Saddle River, N.J., 2nd edition.
- [5] Martinez, C. J. and Wise, W. R. (2003). Analysis of constructed treatment wetland hydraulics with the transient storage model OTIS. *Ecological Engineering*, 20(3):211 – 222.
- [6] Schlesinger, M. and Ramankutty, N. (1994). An oscillation in the global climate system of period 65-70 years. *Nature*, 367:723–726.
- [7] Solo, V. (1980). Some aspects of recursive parameter estimation. *Int. Jnl. Control*, 32:395–410.
- [8] Young, P. C. (2011). *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. Springer-Verlag, Berlin.
- [9] Young, P. C. (2015). Refined instrumental variable estimation: Maximum likelihood optimization of a unified Box-Jenkins model. *Automatica*, 52:35–46.
- [10] Young, P. C. (2025). *The Art and Craft of Data-Based Mechanistic Modelling*.