Leaf River Data: DBM Modelling Fixed Interval Smoothing and Flow Forecasting

Prepared for PUB-IAHS international Workshop
Lugano, July 2004

Peter Young and Renata Romanowicz
p.young@lancaster.ac.uk

Centre for Research on Environmental Systems and Statistics
Lancaster University
and
Centre for Resource and Environmental Studies
Australian National University

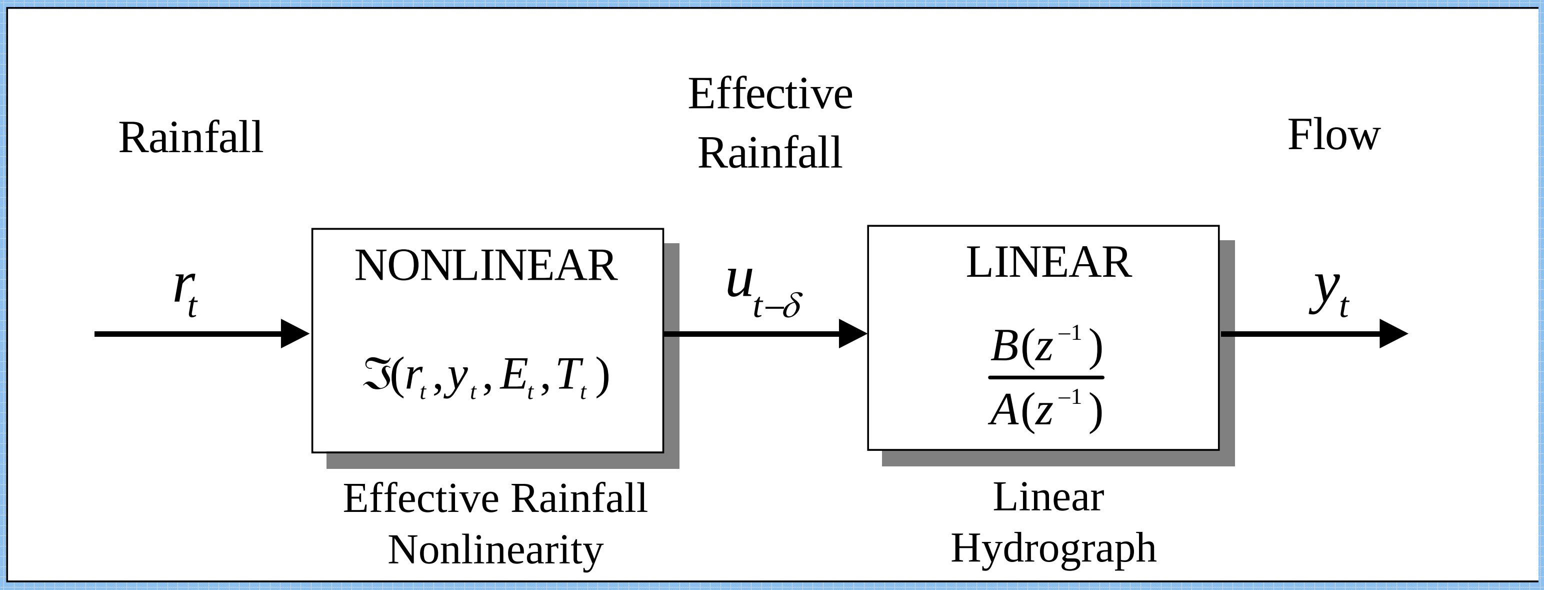
- The Leaf River Data: Inductive DBM Modelling
- Non-Parametric State Dependent Parameter (SDP) Analysis
- Parametric Estimation of the DBM Model
- Kalman Filter-based Forecasting and Fixed Interval Smoothing
    1. Based on prior data analysis
    2. *Initial* study of optimization by prediction error decomposition
- Conclusions

The Leaf River: DBM Modelling

An inductive model based on the minimum of prior assumptions

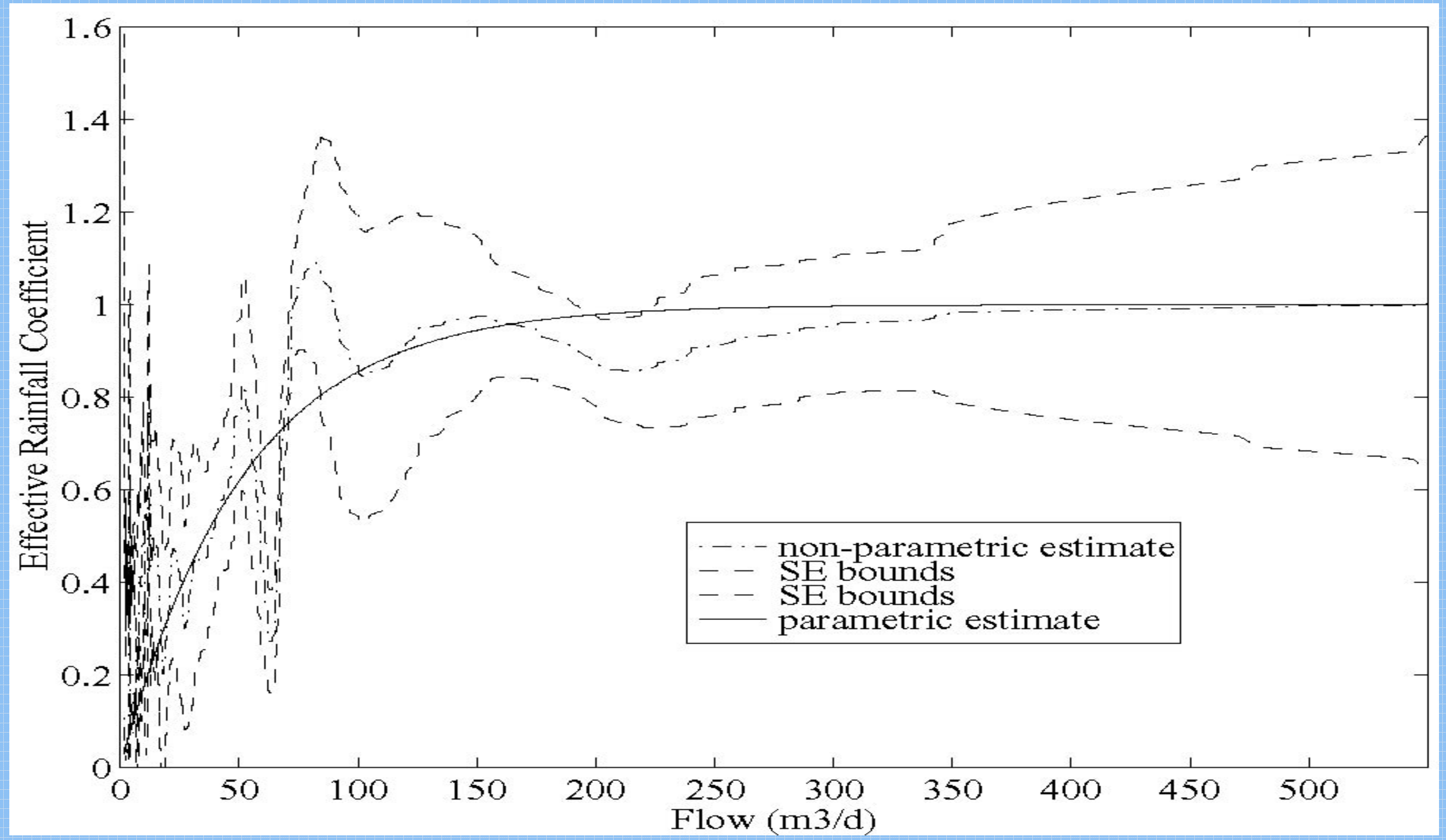
Note: only rainfall and flow measurements are used in this analysis

A Generic Form for DBM & Most Top-Down, Rainfall-Flow Models



- In conceptual models, this generic form arises from hydrological considerations and the modelling is based on a hypothetico-deductive approach.
- In DBM models, the generic form is identified directly from the rainfall-flow data by an inductive approach, without any prior assumption about the model form. Only after the model form has been identified in this manner is the model interpreted in hydrological terms.

SDP Estimation of the Effective Rainfall Nonlinearity



ER nonlinearity estimated initially by non-parametric SDP estimation;then finally with the ER nonlinearity parameterized as a rising exponential function.

Initial Unconstrained RIV Identification and Estimation of TF Model

Refined Instrumental Variable (RIV) estimation of the TF model, with effective rainfall u_t as input, yields the following model

$$y_t = \frac{0.0497 + 0.0590 z^{-1} - 0.0926 z^{-2}}{1 - 2.2115 z^{-1} + 1.6993 z^{-2} - 0.4715 z^{-3}} u_t + \frac{1}{C(z^{-1})} e_t \quad \mathrm{var}(e_t) = 123.4$$

where $C(z^{-1})$ is an AR(7) autoregressive model.

This model explains the data quite well (simulation $R_T^2 = 0.86$) but:

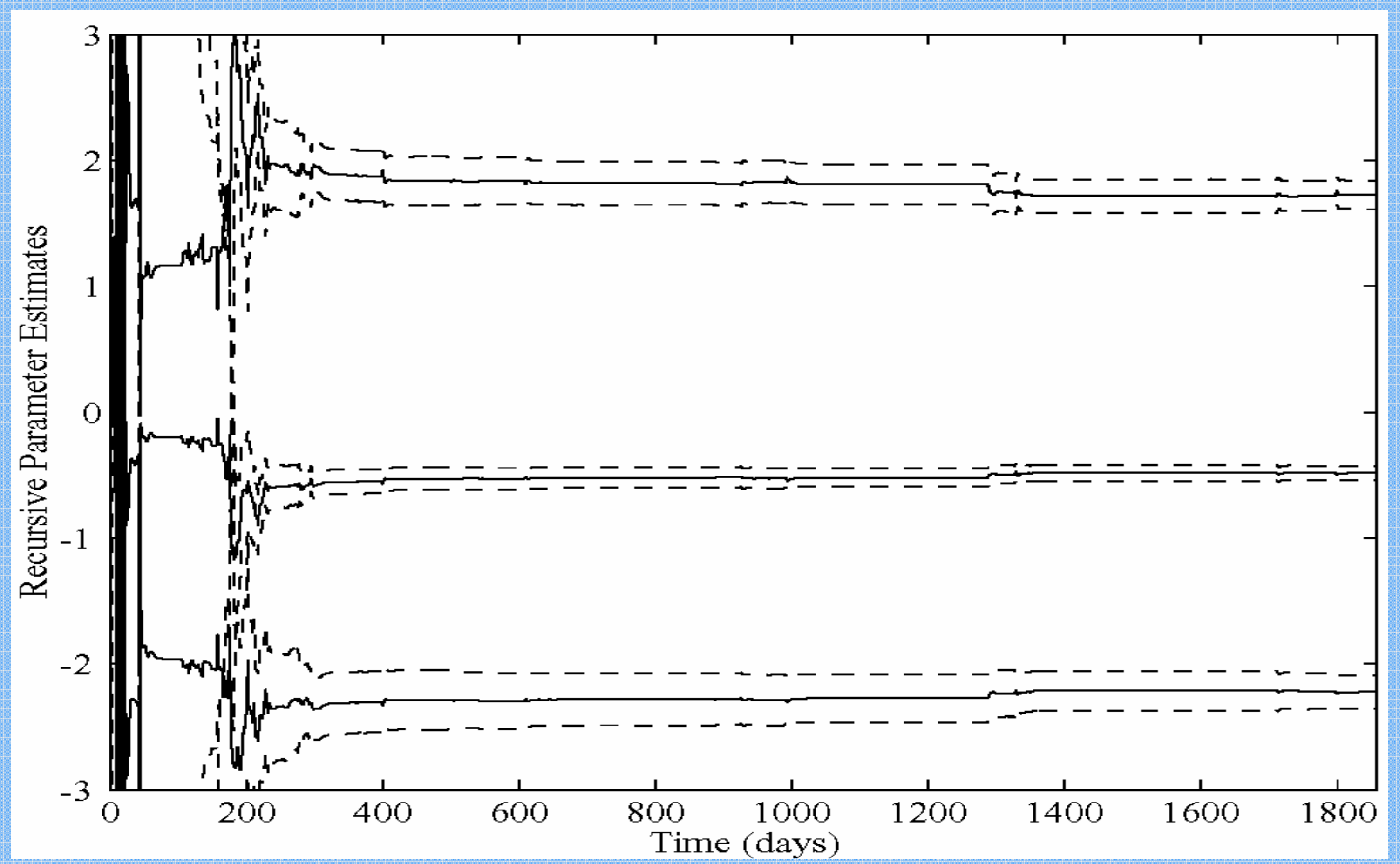
- the eigenvalues are complex (not acceptable for DBM modelling);
- the residual error is heteroscedastic.

However, the variance of the one-step-ahead forecasts is relatively low, yielding $R^2 = 0.95$.

Note: The TF model and the parameterized ER nonlinearity are estimated concurrently

CRESearch
Key to Success in Environmental and Biological Systems Analysis

Recursive RIV Estimates of unconstrained TF parameters (revealing the Bayesian nature of RIV estimation)



Constrained RIV Identification and Estimation of TF Model

Constrained RIV estimation of the TF model, with the eigenvalues constrained to be real, yields the following model:

$$y_t = \frac{0.0572 + 0.0562z^{-1} - 0.1079z^{-2}}{(1-0.969z^{-1})(1-0.573z^{-1})^2} u_t + \frac{1}{C(z^{-1})} e_t \qquad \mathrm{var}(e_t) = 124.2$$

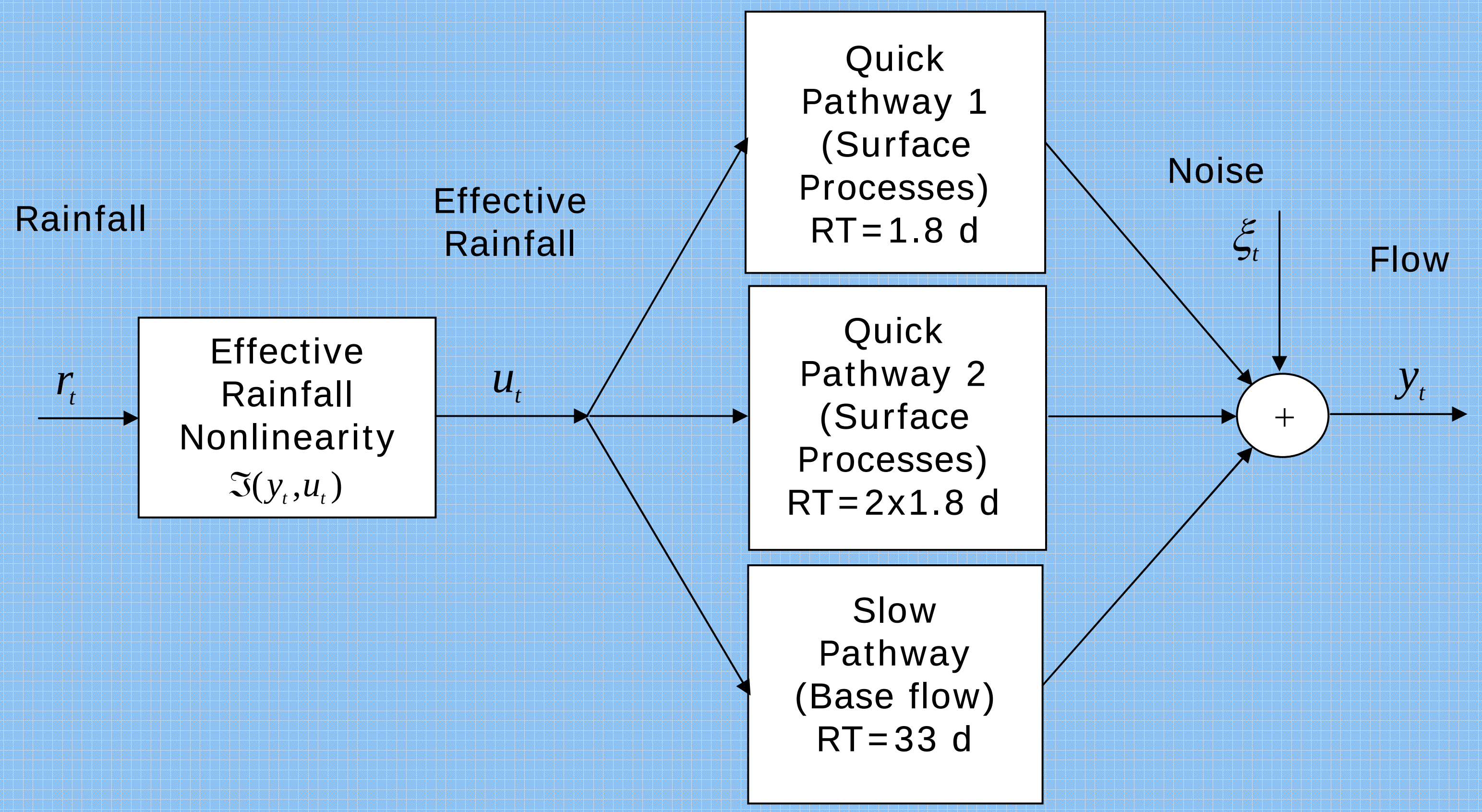
where $C(z^{-1})$ is again an AR(7) autoregressive model.

This model also explains the data quite well ($R_T^2 = 0.84$) but, of course, the residual error is still heteroscedastic.

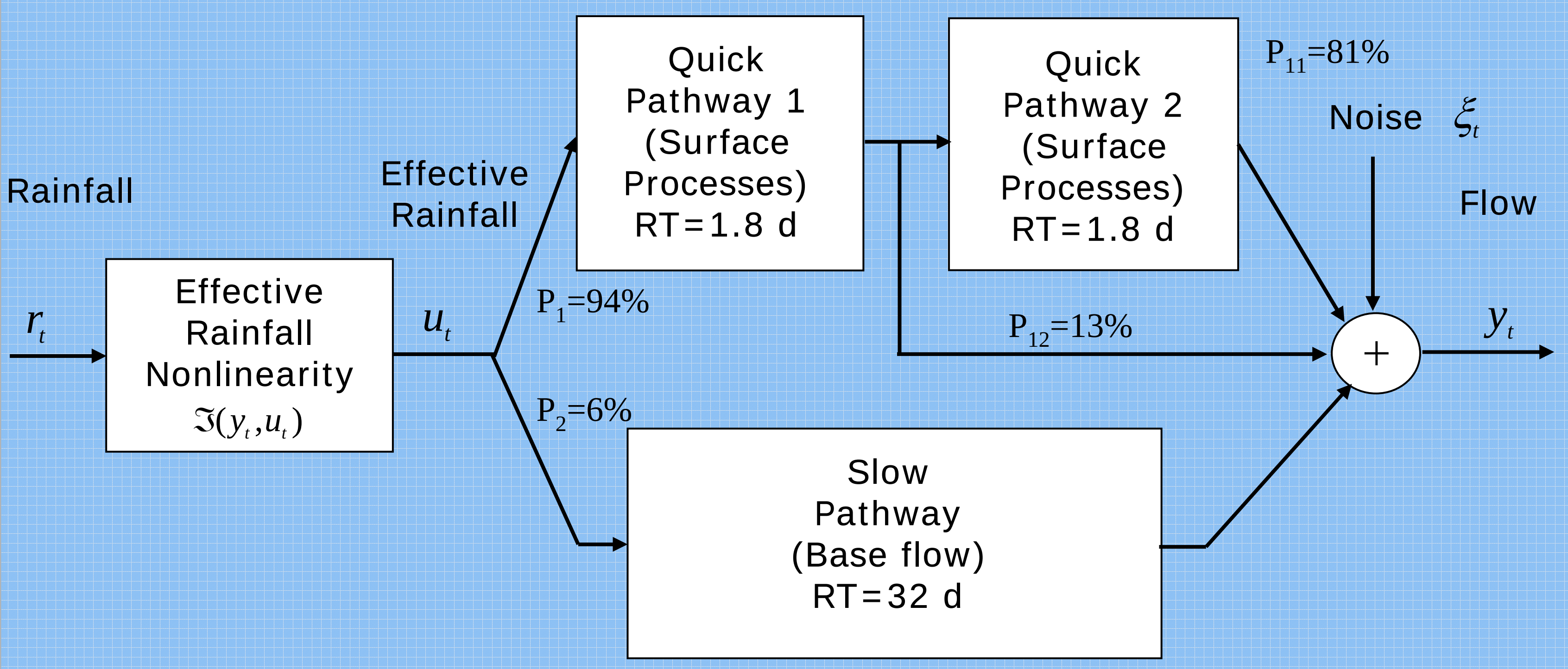
The variance of the one-step-ahead forecast errors is relatively low, yielding $R^2 = 0.95$.

Note: The TF model and the parameterized ER nonlinearity are estimated concurrently

Leaf River DBM Model: Parallel-Flow Decomposition and Physical Interpretation I



Leaf River DBM Model: Parallel-Flow Decomposition and Physical Interpretation II



P_i, i=1,2 are partition percentages; P_{1i},i=1,2 are sub-partitions in 'Quick' pathway

CRESearch
Key to Success in Environmental and Biological Systems Analysis

Validation of DBM Model on Second Half of the Data

- The validation data set (second half of the data) has much larger flow perturbations and so provides a good test for the DBM model

- The constrained DBM model actually explains the validation data set better than the estimation data set, with $R_T^2 = 0.86$.

- Also, the variance of the one-step-ahead forecast errors remains relatively low yielding $R^2 = 0.94$.

Leaf River DBM Model: Stochastic State Space Representation for Forecasting and FIS Estimation

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{bmatrix} = \begin{bmatrix} p_1 & 0 & 0 \\ 0 & p_2 & 0 \\ 0 & 1 & p_3 \end{bmatrix} \begin{bmatrix} x_{1,t-1} \\ x_{2,t-1} \\ x_{3,t-1} \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} u_t + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{w}_t \qquad y_t = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{bmatrix} + \xi_t$$

$$\xi_t = N(0, g y_t \sigma^2) \qquad \mathbf{w}_t = \begin{bmatrix} w_{1,t} \\ w_{2,t} \\ w_{3,t} \end{bmatrix} = N(\mathbf{0}, \mathbf{Q}) \qquad \mathbf{Q} = \begin{bmatrix} q_{11} & 0 & 0 \\ 0 & q_{22} & 0 \\ 0 & 0 & q_{33} \end{bmatrix}$$

$$u_t^* = (1 - \exp(-\lambda y_t)).r_t \qquad u_t = (\sum y_t / \sum u_t^*) u_t^*$$

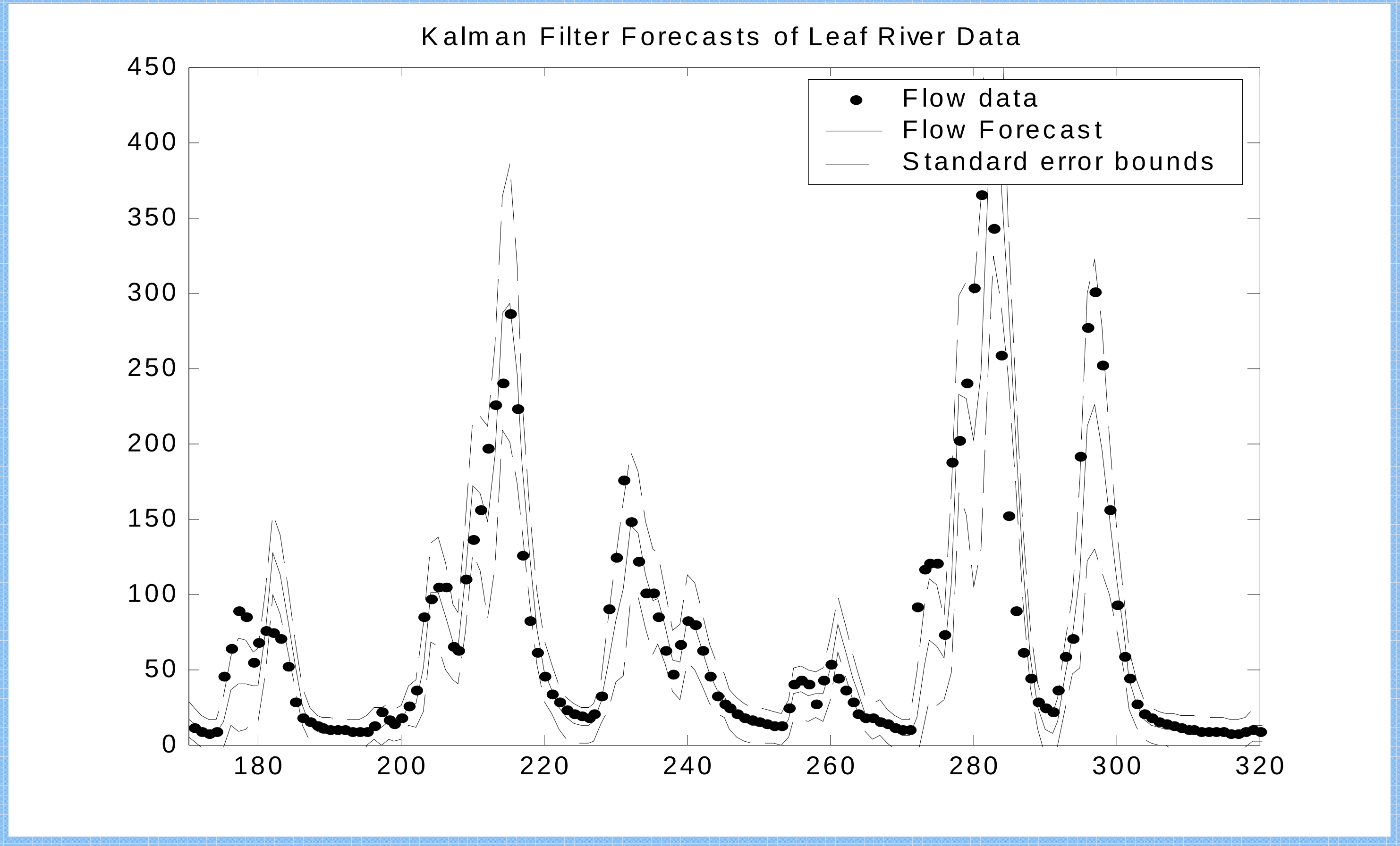
Note that the AR(7) noise model is not included here

The Leaf River: SDP-KF Forecasting and Fixed Interval Smoothing

1. Preliminary estimation results based on prior data analysis without inclusion of the AR(7) noise model

Note: not parameter adaptive

Kalman Filter Forecasts: DBM model with flow-dependent heteroscedastic observation noise



The Leaf River: SDP-KF Forecasting and Fixed Interval Smoothing

2. Optimization based on prediction error decomposition but without inclusion of AR(7) noise model

Note: not parameter adaptive

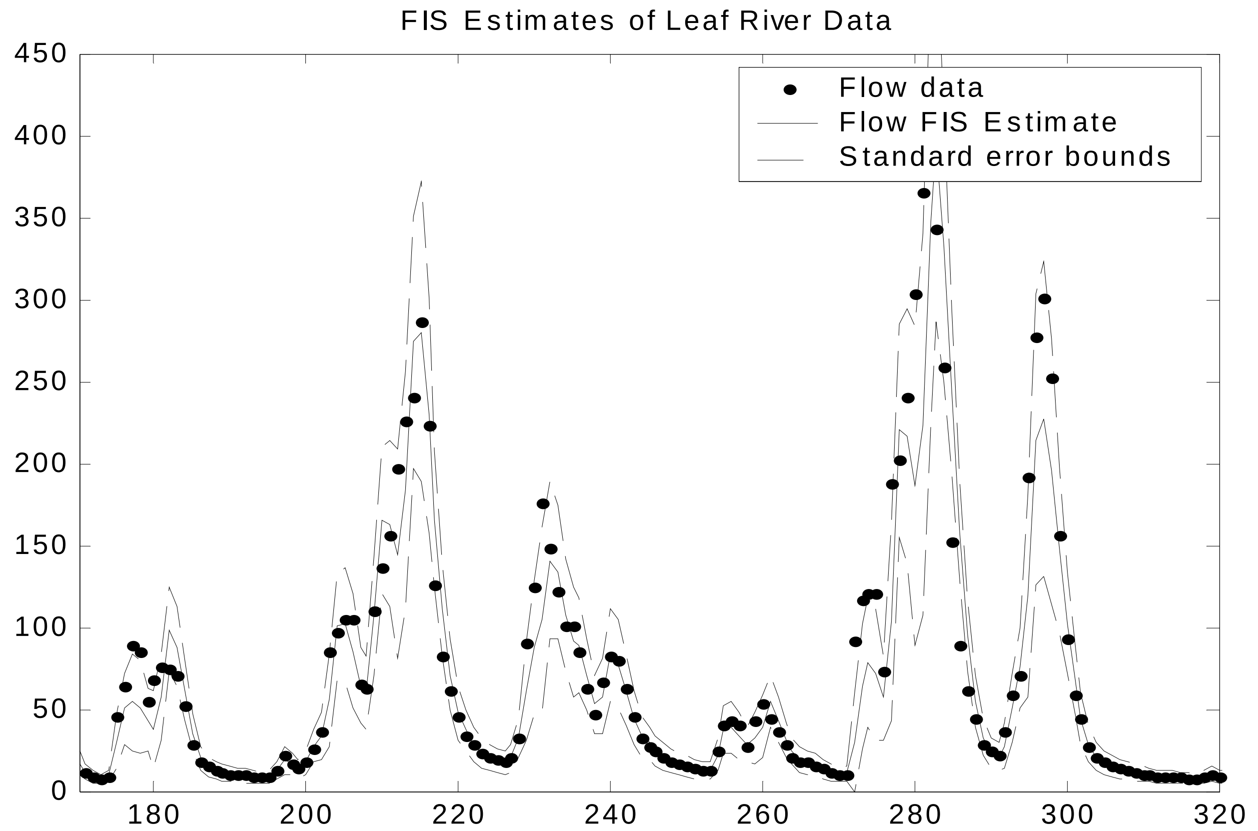
ML Optimization by Prediction Error Decomposition

- Here, the *hyper-parameters* (noise variances that define the stochastic inputs to the SS mode and the observation noise) are optimized to maximize the Likelihood function defined in relation to the innovations of the KF (Schweppe, 1964).

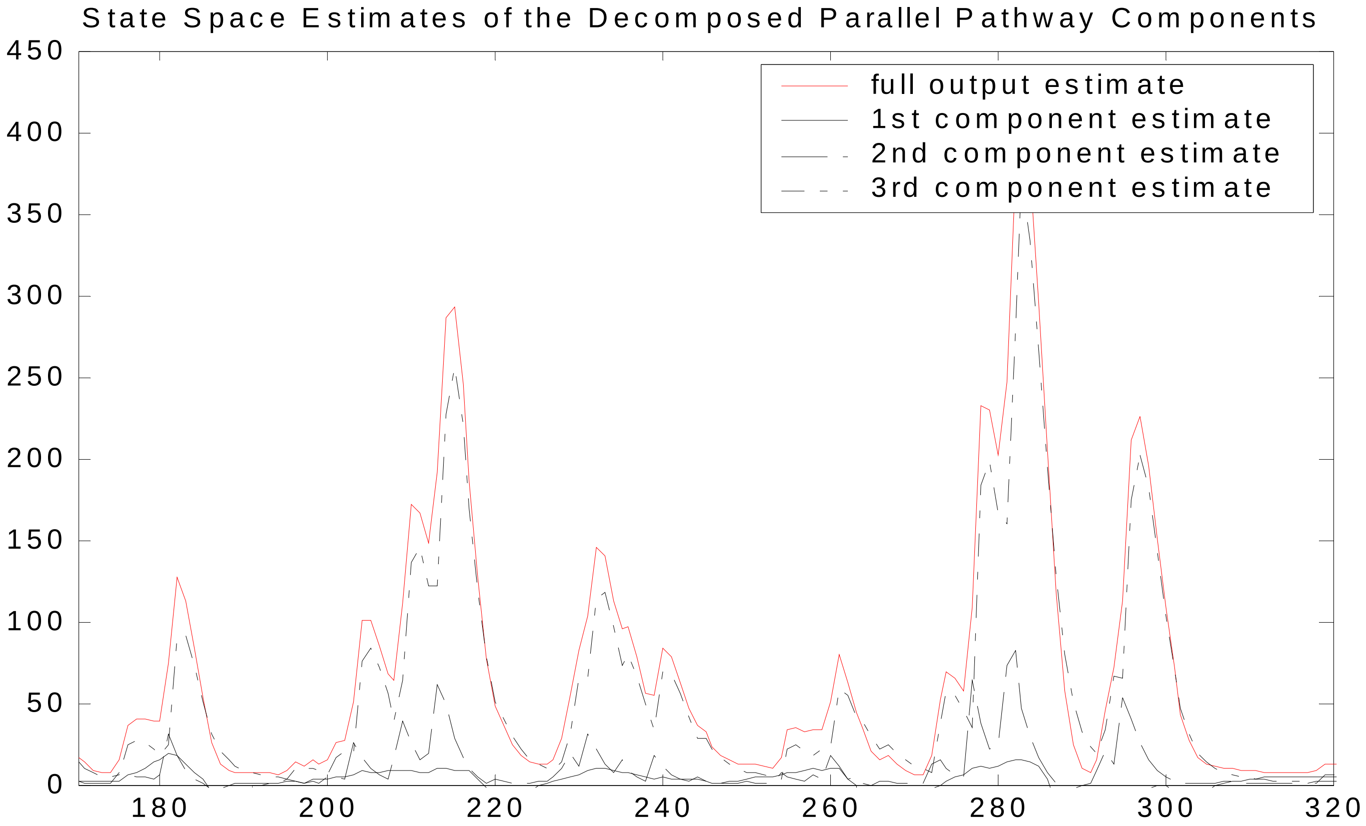
- This is not as straightforward as it seems at first sight because optimization can be difficult if there are too many unknown hyper-parameters and so it may be necessary to restrict the number in some manner.

- In the following, therefore, the $\mathbf{Q}$ matrix used in the preliminary forecasting analysis is retained and a scalar multiplier q is optimized: i.e $\mathbf{Q}_1 = q\mathbf{Q}$. In the case of the observation noise variance, the scalar multiplier g in $gy_t\sigma^2$ is optimized.

FIS Estimation Results

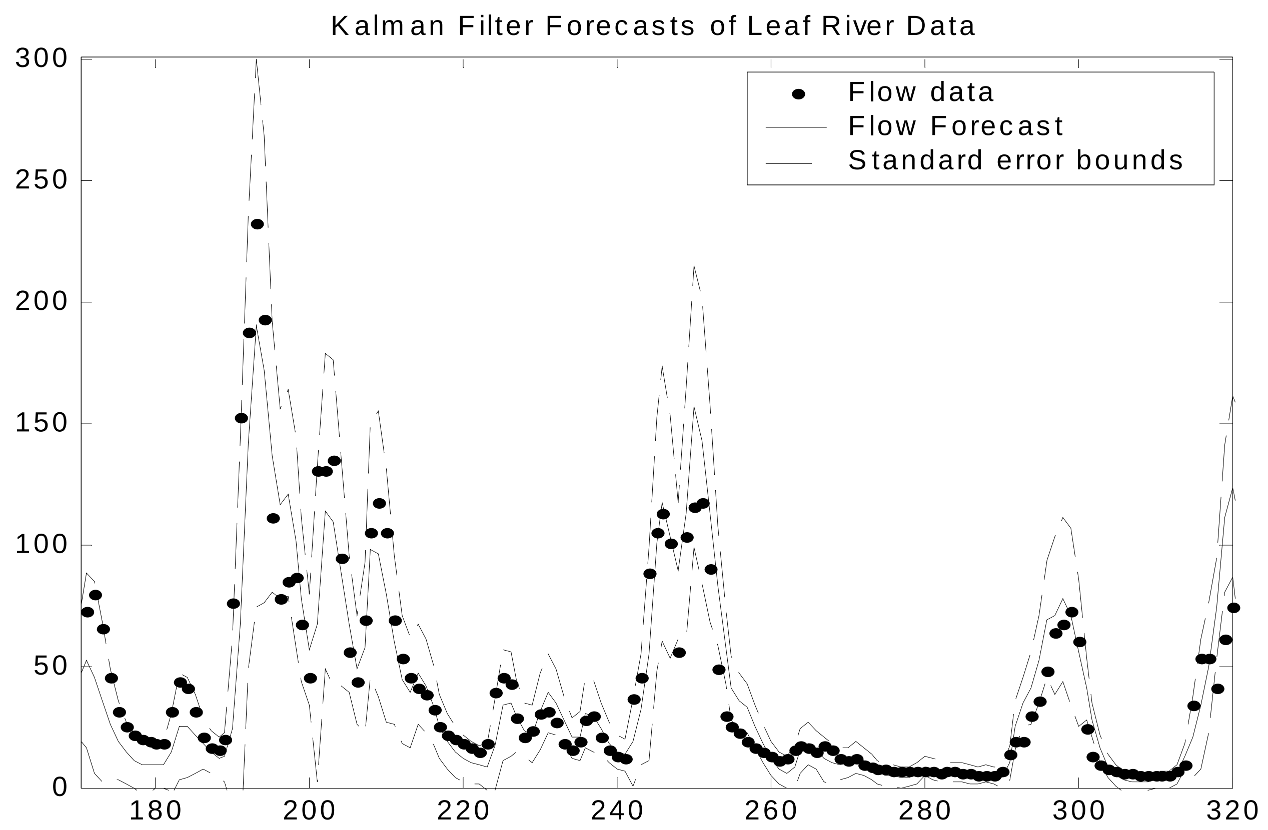


FIS Estimated States (Component Flows)



Note poor estimate of base-flow component (noise model not included in SS model)

Predictive Validation on Second Half of the Data



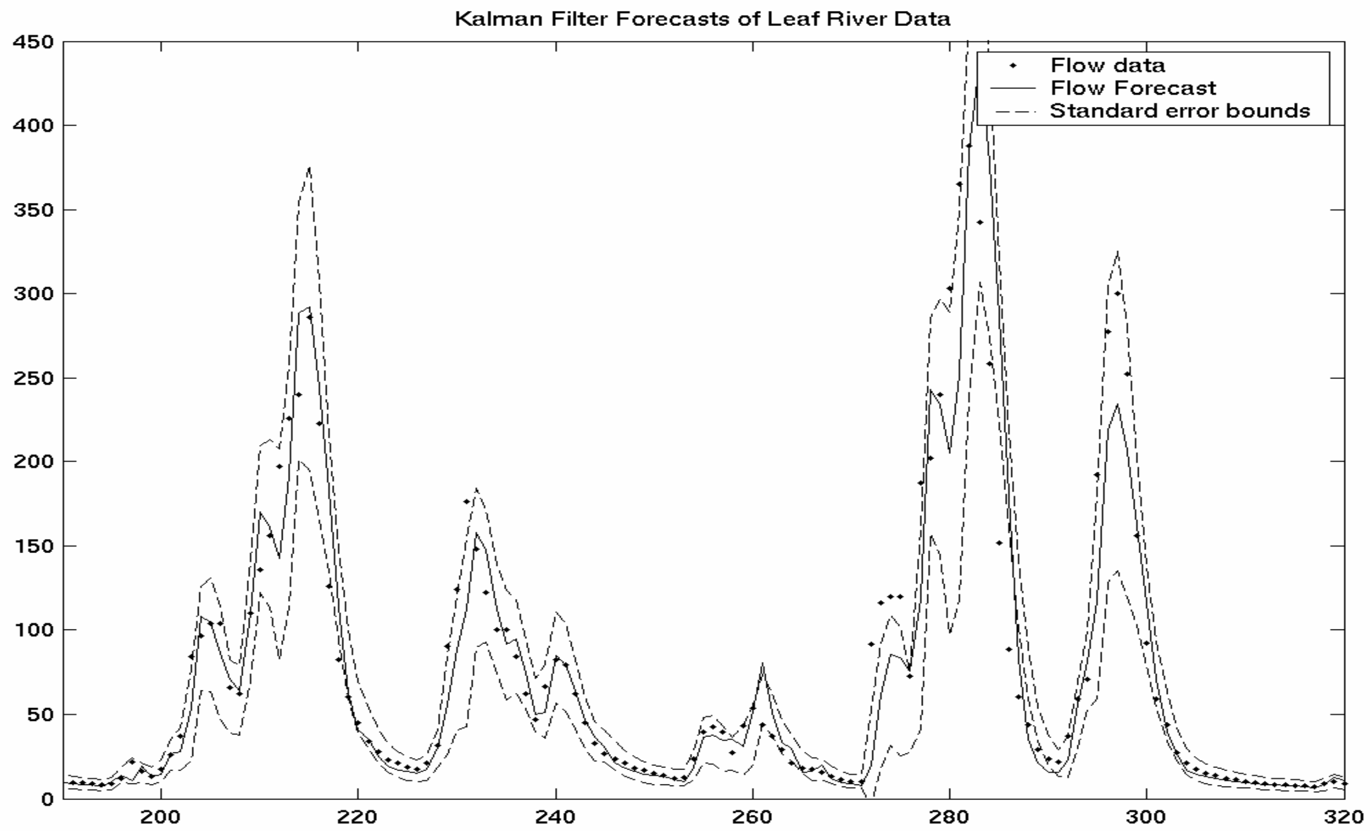
The Leaf River: SDP-KF Forecasting

LATEST RESULTS

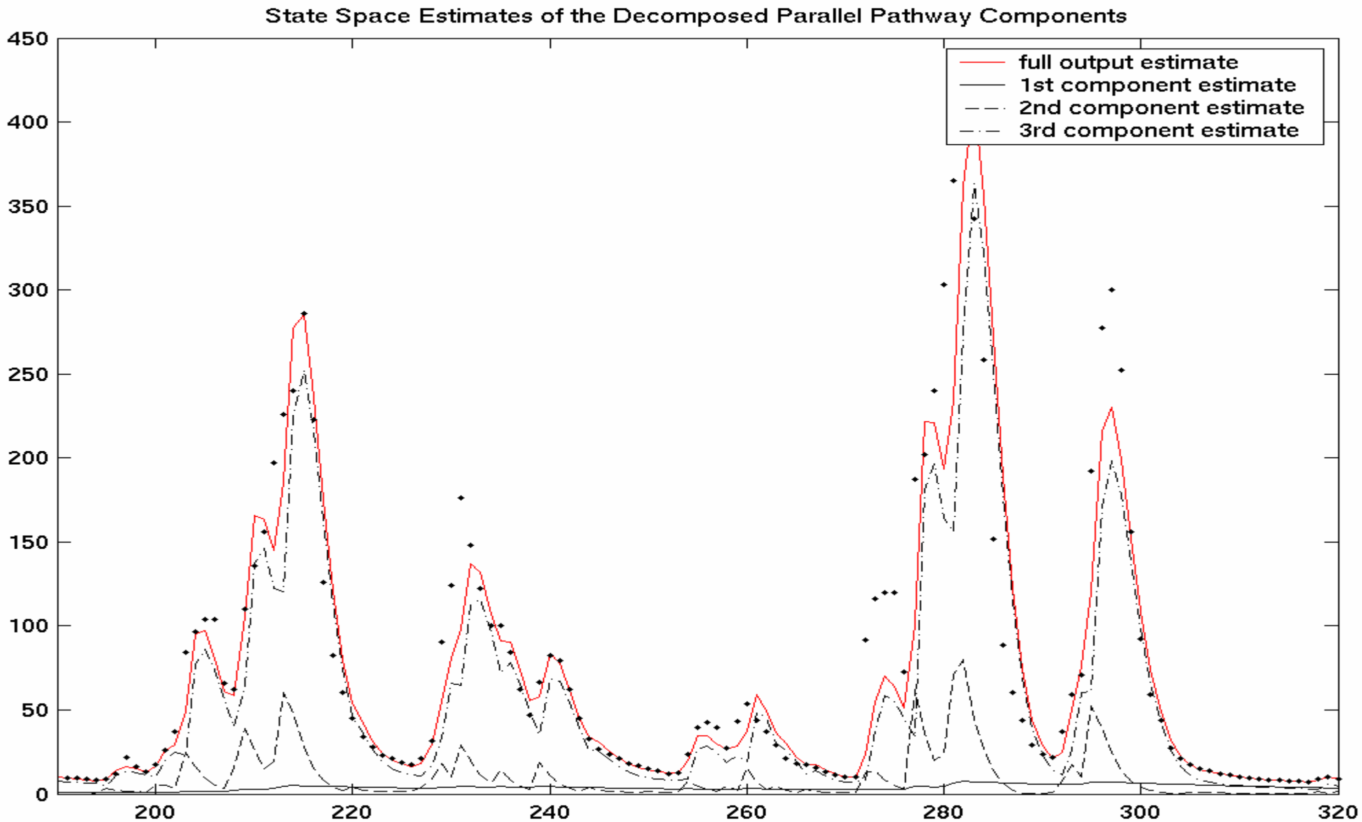
These are preliminary results based on prior data analysis but now with the AR(7) noise model included

Note: not parameter adaptive

Kalman Filter Forecasts: DBM State Space Model including AR(7) Noise



DBM State Space Model including AR(7) Noise: Decomposition



Note good estimate of base-flow component (noise model included in SS model)

CONCLUSIONS

- Time series analysis of most rainfall-flow data suggests a 'Hammerstein' model structure in which an 'effective rainfall' nonlinear process is in series with a low order linear transfer function that characterizes the underlying unit hydrograph of the catchment.
- The *en-bloc or* recursive (analytic Bayesian) *Refined Instrumental variable* (RIV) algorithm provides a simple but statistically sound method of identification and estimation for this model, which can be interpreted in a *Data-Based Mechanistic (DBM)* model form.
- The SDP-KF and SDP-FIS algorithms, based on the state-space form of the SDP-DBM model, with flow dependent heteroscedastic noise, provide the main engines for recursive (if necessary adaptive) forecasting, data assimilation and smoothing.
- More complex and computationally intensive numerical Bayesian methods do not appear essential for this kind of simple DBM model.
- *All* rainfall-flow models will benefit from more research on the nature of the effective rainfall nonlinearity and the best ways to incorporate distributed (spatial) information in order to produce spatial forecasts.
- The results reported here were obtained using SDP, RIV, and FIS algorithms available in the CAPTAIN Toolbox for Matlab (see our web page: but *kalmanfis* algorithm used here not yet in Toolbox).