Neurofuzzy Network State Space Identification Algorithms using Prefiltering for Non-linear Kalman Filtering

X. Hong, C. J. Harris and P. A. Wilson†

Image, Speech and Intelligent System Group
Department of Electronics and Computer Science
University of Southampton, Highfield, Southampton SO17 1BJ

†Department of Ship Science, University of Southampton

Abstract — A new state estimator algorithm is introduced based on a neurofuzzy network and the Kalman filter algorithm. The major contribution of the paper is recognition of a bias problem in the parameter estimation of the state space model and the introduction of a simple effective prefiltering method to achieve unbiased parameter estimates in the state space model which will then be applied for state estimation using the Kalman filtering algorithm. Fundamental to this method is a simple prefiltering procedure using a non-linear principal component analysis (PCA) method based on the neuro-fuzzy basis set. This prefiltering procedure can be performed without *prior* system structure information. Some numerical examples are included to demonstrate the effectiveness of the new approach.

Keywords — neurofuzzy networks, Kalman filter, bias, state-space model, PCA, identification.

1 Introduction

The reconstruction of real system state from noisy observations is important in system modelling, control and prediction (Goodwin and Sin, 1984, Gibson, *et al*, 1992, Chui and Chen, 1987). A state space model is widely applied for calculating the state estimator using Kalman filter related algorithms which forms the framework of most current state estimation and tracking algorithms, and these algorithms have been widely applied in industry (Chui and Chen, 1987). Given a known state space model together with a knowledge of the statistics of noise, additive state and observation, the traditional Kalman filter algorithm, based on a linear state space model, provides an optimal recursive estimate of the unknown state vector and associated covariance matrices. However, if the system is non-linear the Kalman filter is no longer optimal since the linear assumption is inappropriate. An extended version based on the first order Taylor series expansion provides only a local linear approximation to the system, potentially resulting in divergence and stability problems.

Non-linear system dynamical modelling using neural networks including a B-spline neuro-fuzzy network has been widely studied in recent years (Brown and Harris, 1994). Neural networks provide a highly sophisticated capability for non-linear function approximation and the B-spline neurofuzzy network has desirable properties such as a compact support, a partition of unity, locality, logicality and transparency. The B-spline neuro-fuzzy network and algorithms have been extensively developed enabling the applicability of state estimation. The extended Kalman filter algorithm has been applied to the B-spline neuro-fuzzy network structure to render a new neuro-fuzzy multi-sensor data fusion algorithm with applications in helicopter guidance (Doyle and Harris, 1996). Alternatively, neuro-fuzzy state estimator approaches have been developed in which a neuro-fuzzy system input-output model is converted into state-space representation followed by state estimation using Kalman filter algorithms (Wu and Harris, 1997).

The identification of minimal state space models is essential for effective state estimation. In this work, a new state estimator is introduced based on a neurofuzzy network and the Kalman filter algorithm. The paper's major contribution is a recognition of a bias problem in the parameter estimation of the state space model and the introduction of a simple effective prefiltering method to achieve unbiased parameter estimates in the state space model which will then be utilised for state estimation using the classical Kalman filtering algorithm. The new prefiltering procedure has been introduced based on a non-linear principal component analysis (PCA) (A. Papoulis, 1965, A. R. Webb, 1993) method using the B-spline neuro-fuzzy basis function. A bias in state space model was initially investigated based on the neuro-fuzzy operating point dependent structure (Feng and Harris, 1997). It was found that a direct convertion from an identified input-output model could result in a parameter estimate bias in the state space model. To overcome this problem, it may be necessary to construct a system input-output model with a noise model in the structure, but this would complicate the model construction and parameter training process. This paper then proposes a simple method to overcome these disadvantages. A preprocessing procedure of prefiltering over the noisy observation data was performed followed by the identification of the system which is based on the preprocessed data instead of the noisy observations. In this mode, the achieved parameter estimates for the state space model can be justified to possess good statistical properties. The new non-linear prefiltering procedure utilized only noisy system observations without *prior* knowledge of the actual system structure. Numerical examples are included to show the effectiveness of the new approach.

2 Problem formulation

Consider a non-linear system which is described by

$$z(t) = f(z(t-1), \cdots, z(t-n_y), u(t-1), \cdots, u(t-n_u), \mathbf{\Theta}) \quad (1)$$

$$y(t) = z(t) + \xi(t) \quad (2)$$

where $t = 1, 2, 3, \cdots$; n_u, n_y are positive intergers representing lag in system input $u(t)$ and system output $y(t)$. $f(\bullet)$ is an unknown non-linear function. $z(t)$ is an unobservable system state variable that is to be estimated from $y(t)$. The observation noise $\xi(t)$ is an uncorrelated sequence with variance σ_ξ^2 (denoted as $\mathbf{R}$ in Appendix B). $\mathbf{\Theta}$ represents parameter vector of an appropriate dimension that is associated with the selected model structure. Two aspects of system identification are the determination of model structure $f(\bullet)$ and the estimation of the parameter $\mathbf{\Theta}$.

Substituting Eq.(1) into Eq.(2), the system can be represented by

$$y(t) = f(z(t-1), \cdots, z(t-n_y), u(t-1), \cdots, u(t-n_u), \mathbf{\Theta}) + \xi(t) \quad (3)$$

From Eq.(2), it is noted that $z(t-1), z(t-2), \cdots, z(t-n_y)$ contains $y(t-1), y(t-2), \cdots, y(t-n_y)$ and $\xi(t-1), \xi(t-2), \cdots, \xi(t-n_y)$. Therefore in order to obtain an unbiased estimate for $\mathbf{\Theta}$ in Eq.(1) using Eq.(3), the model should incorporate noise terms into the model structure and take the form

$$\begin{aligned} y(t) = {} & f(y(t-1), \cdots, y(t-n_y), u(t-1), \cdots, u(t-n_u), \xi(t-1), \cdots, \\ & \xi(t-n_\xi), \mathbf{\Theta}) + \xi(t) \end{aligned} \quad (4)$$

where n_ξ denotes the lag of noise term in the model. If a model

$$y(t) = f(y(t-1), \cdots, y(t-n_y), u(t-1), \cdots, u(t-n_u), \mathbf{\Theta}) + \xi(t) \quad (5)$$

is used for the identification of the model structure $f(\bullet)$ and parameter $\mathbf{\Theta}$ in Eq.(1), this could induce a bias in parameter estimates $\mathbf{\Theta}$. In the following this point will be further developed.

A neuro-fuzzy network using B-spline function as the membership function has been introduced and proven to be capable of representing complex non-linear system dynamics with optimal approximation capability (Brown and Harris, 1994). Assume that a model structure which has good approximation ability can be determined by some off-line construction algorithm (K. M. Bossley, 1997). Define an observation vector as

$$\mathbf{x}(t) = [y(t-1), \cdots, y(t-n_y), u(t-1), \cdots, u(t-n_u)]^T \in \Re^{n_y+n_u},$$

It may be shown that Eq.(1) can be expressed as a class of operating point dependent autoregressive (AR) model by a B-spline neural network (Wu and Harris, 1997, Feng and Harris, 1998).

$$z(t)=\sum_{i=1}^{n_y}\theta_i(\mathbf{x}(t))z(t-i)+\sum_{i=n_y+1}^{n_y+n_u}\theta_i(\mathbf{x}(t))u(t+n_y-i)+\varepsilon(t) \tag{6}$$

where $\varepsilon(t)$ is an uncorrelated model residual in modelling $z(t)$ using B-spline neuro-fuzzy network with a variance $\sigma^2_{\varepsilon(t)}$ (denoted as $\mathbf{Q}$ in Appendix B). Since most of the system noise is additive observation noise and a B-spline neural network structure with good approximation property is applied, it can be assumed that $\sigma^2_{\varepsilon(t)} \ll \sigma^2_{\xi(t)}$. Denote

$$\mathbf{\Theta}(\mathbf{x}(t))=[\theta_1(\mathbf{x}(t)),\theta_2(\mathbf{x}(t)),\cdots,\theta_{n_y+n_u}(\mathbf{x}(t))]^T\in\Re^{n_y+n_u}$$

as parameter vector which is modelled using neuro-fuzzy network as

$$\begin{aligned}\theta_i(\mathbf{x}(t)) &= \mathbf{\Omega}_i^T\mathbf{N}(\mathbf{x}(t))\\ &= \sum_{k=1}^{p}\omega_{ik}N_k(\mathbf{x}(t))\\ &= \sum_{k=1}^{p}\omega_{ik}\{\prod_{i=1}^{n_y}B_{k_i,m}(y(t-i))\prod_{i=n_y+1}^{n_y+n_u}B_{k_i,m}(u(t+n_y-i))\},\qquad i=1,\cdots,n_y+n_u\end{aligned} \tag{7}$$

in which p is the number of B-spline basis functions which are used to approximate the system. m denotes the given polynomial degree of B-spline function. The B-spline basis function $\mathbf{N}(\mathbf{x}(t))=[N_1(\mathbf{x}(t)),N_2(\mathbf{x}(t)),\cdots,N_p(\mathbf{x}(t))]^T\in\Re^p$ is formed as tensor products of one-dimensional B-spline models $B_{k_i}(\bullet)$ consisting of L_y (for $y(t)$) and L_u (for $u(t)$) basis functions respectively (Appendix A). Note that $k_i=1,2,\cdots,\max\{L_y,L_u\}$ depending on the position of each component of $\mathbf{x}(t)$. The size of full model set $p=(L_y)^{n_y}(L_u)^{n_u}$, where the superscript denotes polynomial degree. The coefficient vector $\mathbf{\Omega}_i=[\omega_{i1},\omega_{i2},\cdots,\omega_{ip}]^T\in\Re^p$ represents the set of adjustable parameters associated with the set of basis functions.

Let $\mathbf{z}(t)=[z(t),z(t-1),\cdots,z(t-n_y+1)]^T$ be a state vector and denote

$$\tilde{u}(t)=\sum_{i=n_y+1}^{n_y+n_u}\theta_i(\mathbf{x}(t))u(t+n_y-i) \tag{8}$$

It may be shown that Eq.(2), Eq.(6) and Eq.(8) can be written in state space form as

$$\mathbf{z}(t+1) = \mathbf{A}(t)\mathbf{z}(t)+\mathbf{B}\tilde{u}(t)+\mathbf{\Gamma}\varepsilon(t) \tag{9}$$

$$y(t) = \mathbf{C}\mathbf{z}(t)+\xi(t) \tag{10}$$

$$\mathbf{A}(t)=\begin{bmatrix}\theta_1(\mathbf{x}(t)) & \theta_2(\mathbf{x}(t)) & \cdots & \theta_{n_y-1}(\mathbf{x}(t)) & \theta_{n_y}(\mathbf{x}(t))\\ 1 & 0 & \cdots & 0 & 0\\ 0 & 1 & \cdots & 0 & 0\\ \cdots & \cdots & \cdots & \cdots & \\ 0 & 0 & \cdots & 1 & 0\end{bmatrix}\in\Re^{n_y\times n_y}$$

$\mathbf{B} = [1, 0, \cdots, 0, 0]^T \in \Re^{n_y}$, $\mathbf{C} = [1, 0, \cdots, 0, 0] \in \Re^{n_y}$ and $\mathbf{\Gamma} = [1, 0, \cdots, 0, 0]^T \in \Re^{n_y}$. After a proper model is identified, the Kalman filter algorithm (Appendix B) is readily applicable based on Eq.(9) and Eq.(10) (Goodwin and Sin, 1984).

Consider the problem of construction of the state space model Eq.(9) and Eq.(10) given information of the system output $\{y(t-1), y(t-2), \cdots\}$ and input $\{u(t-1), u(t-2), \cdots\}$. Throughout it is assumed that n_y and n_u are correctly predetermined. It is noted that Eq.(6) cannot be directly used for estimation as $z(t)$ is unobservable. Substituting Eq.(6) into Eq.(2), it may be shown that

$$y(t) = \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t)) z(t-i) + \sum_{i=n_y+1}^{n_y+n_u} \theta_i(\mathbf{x}(t)) u(t+n_y-i) + \varepsilon(t) + \xi(t) \tag{11}$$

$$\begin{aligned} y(t) &= \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t)) y(t-i) + \sum_{i=n_y+1}^{n_y+n_u} \theta_i(\mathbf{x}(t)) u(t+n_y-i) \\ &\quad - \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t)) \xi(t-i) + \varepsilon(t) + \xi(t) \\ &\approx \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t)) y(t-i) + \sum_{i=n_y+1}^{n_y+n_u} \theta_i(\mathbf{x}(t)) u(t+n_y-i) \\ &\quad - \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t)) \xi(t-i) + \xi(t) \end{aligned} \tag{12}$$

by assuming that $\sigma^2_{\varepsilon(t)} \ll \sigma^2_{\xi(t)}$. If this assumption is weak, then it can be assumed that the model takes the form

$$\begin{aligned} y(t) &= \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t)) y(t-i) + \sum_{i=n_y+1}^{n_y+n_u} \theta_i(\mathbf{x}(t)) u(t+n_y-i) \\ &\quad + \sum_{i=n_y+n_u+1}^{n_y+n_u+n_e} \theta_i(\mathbf{x}(t)) e(t+n_y+n_u-i) + e(t) \end{aligned} \tag{13}$$

where $e(t)$ denotes the model residual which is uncorrelated with $\{y(t-1), y(t-2), \cdots, y(t-n_y)\}$. Note that the appropriate dimension of parameter vector $\mathbf{\Theta}$ is now expanded to $(n_y + n_u + n_e)$ in Eq.(13).

Both Eq.(12) and Eq.(13) takes the form of a state-dependent model which has been introduced from a quite general non-linear background (Priestley, 1980), but here we will call the model as an operating dependent model to avoid confusion with the state variable $z(t)$ which is unobservable. This is because the system input $u(\bullet)$ and system out $y(\bullet)$ forms an operating point or observation vector $\mathbf{x}(t)$.

Consider the parameter estimation based on Eq.(13). It is noted that noise terms are included in the model structure and to obtain unbiased parameter estimates the extended least squares related algorithms can be used (Goodwin and Sin, 1984). In this model structure the free parameters $\mathbf{\Omega}_i$'s, which are in the first layer representing the set of parameters associated with the set of basis functions, may be determined by for example back-propagation. Eq.(13) need to be iteratively applied in training because of the included noise terms. This is a disadvantage because the model structure and training process are much more complicated. Now Eq.(13) can be rewritten as

$$y(t) = \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t))y(t-i) + \sum_{i=n_y+1}^{n_y+n_u} \theta_i(\mathbf{x}(t))u(t+n_y-i) + \tilde{\xi}(t) \tag{14}$$

where $\tilde{\xi}(t) = \sum_{i=n_y+n_u+1}^{n_y+n_u+n_e} \theta_i(\mathbf{x}(t))e(t+n_y+n_u-i) + e(t)$ is an assimulated noise terms. If this were an independent additive noise terms, the parameters could be estimated without taking into account of the noise terms based on Eq.(14) yielding $\hat{\Theta} = \arg\{\min \sum_{t=1}^{N} \tilde{\xi}(t)^2, \quad \forall \Theta\}$, where N is the number of tha data sample in the estimation data set. But as $\tilde{\xi}(t)$ will in general be correlated with the past observations $\{y(t-1), y(t-2), \cdots, y(t-n_y)\}$, the parameter estimates obtained will be biased. State estimation derived from the state space model with seriously biased parameters is unlikely to be satisfactory in practice. In the following a prefiltering procedure is proposed to overcome this bias problem.

3 Parameter estimation based on prefiltering using principal component analysis (PCA)

An alternative approach to the parameter estimation problem is presented which is based on a prefiltering procedure using principal component analysis (PCA). A preprocessing procedure of prefiltering over the noisy observation data is performed followed by the identification of the system which is based on the preprocessed data instead of the noisy observations. In this mode, the achieved parameter estimates for the state space model can be shown to possess good statistical properties. The new non-linear prefiltering procedure of principal component analysis (PCA) (A. R. Webb, 1993) method based on the B-spline neural fuzzy basis function is introduced following an initial introduction of prefiltering procedure using linear PCA (A. Papoulis, 1965). Both linear and non-linear procedures utilize only noisy system observations without *prior* knowledge of the actual system structure.

Consider the parameter estimation based on Eq.(11). According to Eq.(2) $z(t)$ may be interpreted as the noise free part of $y(t)$. This suggests that a prefiltered $y_p(t)$ can be used for the approximation of $z(t)$ in Eq.(11) and Eq.(2) as

$$y(t) = \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t))y_p(t-i) + \sum_{i=n_y+1}^{n_y+n_u} \theta_i(\mathbf{x}(t))u(t+n_y-i) + \varepsilon(t) + \xi(t) \tag{15}$$

and

$$y(t) = y_p(t) + \xi(t) \tag{16}$$

respectively, where $z(t)$ is replaced by the prefitered data $y_p(t)$. The basic idea in the new approach is that the prefitered data $\{y_p(t)\}$ is obtained followed by the parameter estimation being performed using Eq.(15) via the minimisation of $\sum_{t=1}^{t=N}[\varepsilon(t) + \xi(t)]^2$ in Eq.(15). If the $\{y_p(t)\}$ are noise free part of $y(t)$, $\varepsilon(t)+\xi(t)$ will be uncorrelated and the obtained parameter estimates will be unbiased. This is advantageous in that the iterative procedure for estimation via building a model containing noise terms in the model structure would be avoided.

The extraction of a noise free part $y_p(t)$ from noisy observation $y(t)$ can be formulated using principal component analysis (PCA). This extraction is based on the fact that noise existing in the system output $y(t)$ is much less predictable than the noise free part $y_p(t)$, and hence most of the system energy in system output y(t) should be stored in the noise free part $y_p(t)$. Then the most important principal components can be used to construct the noise free part $y_p(t)$. The significance of PCA is that it is very efficient in preserving signal energy, and it is easy to include as many principle components as necessary into the noise free part $y_p(t)$ to achieve good approximation properties, and no *prior* system structure information is needed. Linear PCA based method is initially described as follows (A. Papoulis, 1965).

3.1 Linear principal component analysis

Denote system output vector $\mathbf{y}$ as

$$\mathbf{y} = [y(1), y(2), \cdots, y(N)]^T \in \Re^N$$

The linear PCA is based on the singular value decomposition (SVD) of the covariance matrix $E[\mathbf{y}\mathbf{y}^T]$ as

$$E[\mathbf{y}\mathbf{y}^T] = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \tag{17}$$

where $\mathbf{V} \in \Re^{N\times N}$ is an orthonormal matrix consisting of N orthonormal eigenvectors $\mathbf{v}_j \in \Re^N$, $j = 1, 2, \cdots, N$

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N] \tag{18}$$

and

$$\mathbf{\Lambda} = diag\{\lambda_1, \lambda_2, \cdots, \lambda_N\} \in \Re^{N\times N} \tag{19}$$

where λ_j, $j = 1, 2, \cdots, N$ represent eigenvalues in the order of decreasing multitude, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$. Construct a vector $\mathbf{y}_p$ using the first M principal components

$$\mathbf{y}_p = \sum_{j=1}^{M} c_j \mathbf{v}_j \tag{20}$$

with

$$c_j =< \mathbf{y}, \mathbf{v}_j >$$

where $< \bullet, \bullet >$ denotes the inner product and it may be shown that

$$c_j^2 = \lambda_j \tag{21}$$

Denote $\mathbf{y}_p$ as

$$\mathbf{y}_p = [y_p(1), y_p(2), \cdots, y_p(N)]^T \in \Re^N \tag{22}$$

A prefiltered signal $y_p(t), t = 1, 2, \cdots, N$ is then achieved using Eq.(22). It may be shown that the construction of $\mathbf{y}_p$ is obtained in the sense that $||\mathbf{y} - \mathbf{y}_p||$ is minimised, where $|| \bullet ||$ denotes *Euclidean* norm. The prefiltered signal $y_p(t), t = 1, 2, \cdots, N$ can approximate the noise free part of $y(t)$ by gradually increasing the number M to achieve an appropriate size by checking the residual vector $\mathbf{y} - \mathbf{y}_p$ using an independent model validity tests (Billings and Zhu, 1996). For a given threshold of the variance of residual, the PCA based prefiltered signal $y_p(t)$ is optimal in the sense that M is minimised.

3.2 Nonlinear PCA using neuro-fuzzy B-spline basis function

Linear PCA is directly carried out on the covariance matrix of the output vector. When the system is non-linear the linear PCA may not be appropriate. A non-linear smoother or non-linear filter may be required to effectively retrieve the noise free part $y_p(t)$. As a preprocessing procedure, a simple non-linear prefiltering method with no need to select non-linear model structure would be desirable. Webb (1993) proposed a non-linear PCA approach based on radial basis function. Here a non-linear PCA based method via neuro-fuzzy B-spline basis function is introduced. The PCA analysis is performed on a regression matrix which is constructed based on the full model set of B-spline neural fuzzy basis function. No prior system structure information is needed except knowledge of the system lag n_y and n_u. If unknown, high orders are assumed *ab initio* then iteratively reduced, or they can be estimated by simple cross-correlation.

Denote a regression matrix $\mathbf{P}$ consisting of the full B-spline basis function set $\mathbf{N}(\mathbf{x}(t))$, $t = 1, 2, \cdots, N$. as

$$\mathbf{P} = [\mathbf{N}(\mathbf{x}(1))\mathbf{N}(\mathbf{x}(2)) \cdots \mathbf{N}(\mathbf{x}(N))]^T \in \Re^{N \times p} \tag{23}$$

The main idea is to select a subset of the eigenvectors of $\mathbf{P}$ as a basis in approximating the system output vector $\mathbf{y}$.

Perform the singular value decomposition (SVD) of $\mathbf{P}$

$$\mathbf{P} = \mathbf{V \Lambda U} \tag{24}$$

where $\mathbf{V} \in \Re^{N \times p}$ is an orthonormal matrix consisting of p orthonormal eigenvectors $\mathbf{v}_j \in \Re^N$, $j = 1, 2, \cdots, p$

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p] \tag{25}$$

and

$$\mathbf{\Lambda} = diag\{\lambda_1, \lambda_2, \cdots, \lambda_p\} \in \Re^{p \times p} \tag{26}$$

where λ_j, $j = 1, 2, \cdots, p$ represent eigenvalues in the order of decreasing multitude, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$. $\mathbf{U} \in \Re^{p \times p}$ is an orthonormal matrix. The same notations $\mathbf{V}$, $\mathbf{\Lambda}$, $\mathbf{v}_j$'s and λ_j's as in Section 3.1 are used in association with the eigenvectors and eigenvalues for the sake of simplicity.

Note that these will be different because the SVD decomposition of the regression matrix **P**, and not the output vector **y**'s covariance matrix, is used in the non-linear PCA version. Due to this, an equivalence relationship in the non-linear version of Eq.(21) doesn't hold. This implies that the optimal subset of M eigenvectors used to construct the vector $\mathbf{y}_p$ would not necessarily be exactly the first M principal eigenvectors of **P**. A procedure of selecting an optimal subset of the most important basis using eigenvectors of **P** can however be established simply according to the magnitude of the projection of the output vector **y** towards each eigenvector. A subset of M eigenvectors of **P** with the largest magnitude of projections, representing the largest portion of system energy, will be selected. This extraction process is based on the fact that noise existing in the system output $y(t)$ should be much less predictable than the noise free part $y_p(t)$ and can be represented as the residual with the smallest portion of system energy.

The projection of the output vector **y** on each eigenvector can be computed by

$$c_j = <\mathbf{y}, \mathbf{v}_j>, \qquad j = 1, 2, \cdots, p \tag{27}$$

These eigenvectors are then ranked in the order of decreasing multitude of absolute value of c_j, yielding

$$\left.\begin{array}{rcl} k & = & \arg\{\max[\mid c_i \mid, \quad for \quad i = j, j+1, \cdots, p]\} \\ \tilde{c}_j & = & c_k \\ \tilde{\mathbf{v}}_j & = & \mathbf{v}_k \\ c_k & = & c_j \\ \mathbf{v}_k & = & \mathbf{v}_j \end{array}\right\} j = 1, 2, \cdots, M \tag{28}$$

Construct a vector $\mathbf{y}_p$ using the first M $\tilde{\mathbf{v}}_j's, j = 1, 2, \cdots, M$

$$\mathbf{y}_p = \sum_{j=1}^{M} \tilde{c}_j \tilde{\mathbf{v}}_j \tag{29}$$

In a similar way, $y_p(t)$ is retrieved using Eq.(22) as in Section 3.1. Denote the associated residual vector as

$$\tilde{\mathbf{e}} = \mathbf{y} - \mathbf{y}_p = [\tilde{e}(1), \tilde{e}(2), \cdots, \tilde{e}(N)]^T$$

As the orthonormal conditions $\tilde{\mathbf{v}}_i^T \tilde{\mathbf{v}}_j = 0, i \neq j$ and $\tilde{\mathbf{v}}_i^T \tilde{\mathbf{v}}_j = 0, i = j$ hold, hence

$$\mathbf{y}^T \mathbf{y} = \sum_{j=1}^{M} \tilde{c}_j^2 + \tilde{\mathbf{e}}^T \tilde{\mathbf{e}} \tag{30}$$

An appropriate size M can be determined by checking the residual sequence $\{\tilde{e}(t)\}$ using independent model validity tests (Billings and Zhu, 1996). From Eq.(28) and Eq.(30) $\tilde{c}_1^2 \geq \tilde{c}_2^2 \geq \cdots \geq \tilde{c}_M^2$, This implies that the non-linear PCA based prefiltering approach should also be optimal in the sense of retrieving information contained in the regression matrix using the smallest number of principal components.

Note that in system identification the structure determination is crucial and this implies that the selection of physical variables is important in modelling and control. This makes orthogonal least squares (OLS) algorithms superior to PCA and widely applied (Billings *et al*, 1989). The PCA transformation can obscure physical variable relationships, noting that $y_p(t)$ are transformations of the full basis function set of size p. However, this disadvantage doesn't exist in the present approach because the prefilter is only a preprocessing procedure used to obtain unbiased parameter estimates. As a preprocessing procedure the PCA based prefiltering is effective in that it is optimal in abstracting energy using the smallest number of variables with the additional advantage that non-linear model structure selection is avoided.

Another important point is that the prefiltered signal $y_p(t)$ is only used as an auxiliary variable in approximating $z(t)$, which forms the state vector $\mathbf{z}(t)$, in the identification stage so as to obtain good state space model parameter estimates but not as a real state vector estimator which will be calculated using the Kalman filter algorithm which has more desirable properties for on-line signal processing.

Substituting Eq.(7) into Eq.(15) to construct a one-step ahead predictor

$$\begin{aligned} \hat{y}(t|t-1) &= \sum_{i=1}^{n_y} \theta_i(\mathbf{x}(t)) y_p(t-i) + \sum_{i=n_y+1}^{n_y+n_u} \theta_i(\mathbf{x}(t)) u(t+n_y-i) \\ &= \mathbf{\Omega}^T \mathbf{\Phi}(t-1) \end{aligned} \tag{31}$$

where

$$\begin{aligned} \mathbf{\Omega} = & [\omega_{11}, \cdots, \omega_{1p}, \omega_{21}, \cdots, \omega_{2p}, \cdots, \omega_{(n_y)1}, \cdots, \omega_{(n_y)p}, \omega_{(n_y+1)1}, \\ & \cdots, \omega_{(n_y+1)p}, \cdots, \omega_{(n_y+2)p}, \cdots, \omega_{(n_y+n_u)1}, \cdots, \omega_{(n_y+n_u)p}]^T \end{aligned} \tag{32}$$

$$\begin{aligned} \mathbf{\Phi}(t-1) = & [N_1(\mathbf{x}(t)) y_p(t-1), \cdots, N_p(\mathbf{x}(t)) y_p(t-1), \cdots, N_1(\mathbf{x}(t)) y_p(t-n_y), \cdots, \\ & N_p(\mathbf{x}(t)) y_p(t-n_y), N_1(\mathbf{x}(t)) u(t-1), \cdots, N_p(\mathbf{x}(t)) u(t-1), \cdots, \\ & N_1(\mathbf{x}(t)) u(t-n_u), \cdots, N_p(\mathbf{x}(t)) u(t-n_u)]^T \end{aligned} \tag{33}$$

The normalized least mean squares can be used to yield an estimate of the free parameters $\mathbf{\Omega}$ at time t, denoted as $\hat{\mathbf{\Omega}}(t)$, according to the following recursion (Brown and Harris, 1994)

$$\hat{\mathbf{\Omega}}(t) = \hat{\mathbf{\Omega}}(t-1) + \eta \frac{\mathbf{\Phi}(t-1) e(t)}{\mathbf{\Phi}(t-1)^T \mathbf{\Phi}(t-1) + \eta_1} \tag{34}$$

$$e(t) = y(t) - \hat{y}(t|t-1) \tag{35}$$

$$\hat{y}(t|t-1) = \hat{\mathbf{\Omega}}^T(t-1) \mathbf{\Phi}(t-1) \tag{36}$$

where $\eta \in (0, 2)$ is the learning rate, η_1 is a positive arbitrarily small constant. The initial condition $\hat{\mathbf{\Omega}}(0)$ should be set beforehand. After the parameters are learned using the estimation data set, the state space model given by Eq.(9) and (10) can be constructed automatically. Using the Kalman filter algorithms, state vector estimation can be carried out in an on-line manner on the system as new observations appear only if the system is reasonably stationary. To accomodate a non-stationary system, parameter estimation for the state space model can be repeated using new data.

4 The algorithms

The identification algorithms using prefiltering for non-linear Kalman filtering with neurofuzzy state space model structure can be summarized as follows.

1. The regression matrix $\mathbf{P}$ is constructed according to Eq.(23) (Appendix A). Perform the SVD transformation defined by Eq.(24). Obtain the eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p$. Set M=1;

2. Using Eq.(27)-(29) to construct $\mathbf{y}_p$, check the residual vector $\{\tilde{e}(t)\}$ using model validity tests (Billings and Zhu, 1994). If the model validity tests show that the residual is decorrelated goto step 3, otherwise increase M by 1 and repeat step 2;

3. Retrieve noise free part $y_p(t), t = 1, 2 \cdots, N$ using Eq.(22). The noise free part $[y_p(t-1)$, $y_p(t-2)$, $\cdots$, $y_p(t-n_y)]$ are then substituted into Eq.(33) to construct the regressors $\mathbf{\Phi}(t-1), t = 2, \cdots, N$;

4. Estimate the free parameters $\mathbf{\Omega}(t)$ in the neurofuzzy network using NLMS algorithm Eq.(34)-Eq.(36).

5. Construct neurofuzzy state space model using Eq.(9) and Eq.(10).

6. Apply Kalman filter algorithms Eq.(43)-(45) (Appendix B) to Eq.(9) and Eq.(10) for on-line state vector estimation on a validation data set.

5 Numerical Examples

Example 1: Consider a non-linear system

$$\begin{aligned} z(t) &= \frac{2.5z(t-1)z(t-2)}{1+z^2(t-1)+z^2(t-2)} + 0.3\cos(0.5(z(t-1)+z(t-2))) + 0.5u(t-1) \\ y(t) &= z(t) + \xi(t) \end{aligned} \tag{37}$$

where the system input $u(t) = \sin(\pi * t/20)$. Two noise sequences $\xi(t) \sim N(0, \sigma_\xi^2)$, where the noise levels were set as $\sigma_\xi = 0.05$ and $\sigma_{\xi_2} = 0.08$ respectively, were added to the state variable generating two sequences of system output observation $y(t)$. A data sequence of 1000 samples, plotted in Fig.1, was generated for each noise level. For both cases, the first 500 samples $\{u(t), y(t)\}$ from $t = 1$ to $t = 500$ was used as an estimation set. The Kalman filter algorithm was applied to a validation data set of 500 samples consisting of $\{u(t), y(t)\}$ from $t = 501$ to $t = 1000$.

For the case of $\sigma_\xi = 0.05$, an order-1 ($m = 1$) B-spline neuro-fuzzy network was used to approximate the system. The input vector to the neurofuzzy network basis function was predetermined as $\mathbf{x}(t) = [y(t-1), y(t-2), u(t-1)]^T$, that is, $n_y = 2$, $n_u = 1$. The knot vectors for the input domain was defined as

$$\{-0.5, -0.4, -0.1, 0.65, 1.4, 1.7, 1.8\}$$

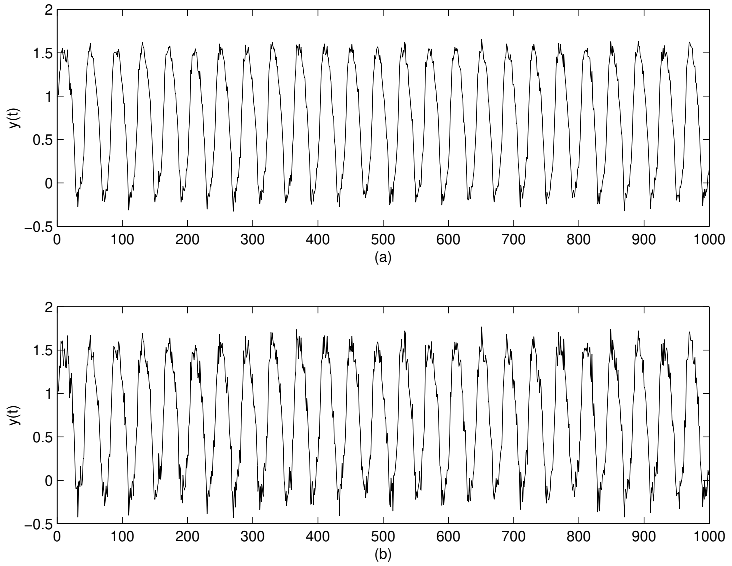

Figure 1: The output data $y(t)$ in example 1; (a) $\sigma_\xi = 0.05$ and (b) $\sigma_\xi = 0.08$

for system output $y(t)$, and

$$\{-1.2, -1.1, -0.9, 0, 0.9, 1.1, 1.2\}$$

for system input $u(t)$, resulting 5 one-dimensional basis functions for $y(t-1)$, $y(t-2)$ and $u(t-1)$ respectively. The 125 multidimensional basis functions were formed using a direct multiplication of one-dimensional basis functions (Appendix A). The regression matrix $\mathbf{P}$ was then constructed according to Eq.(23), and the SVD transformation defined by Eq.(24) was performed to $\mathbf{P}$. Eqn's.(27)-(29) was then applied to construct $\mathbf{y}_p$, and the process was terminated at $M = 30$ by checking the residual vector $\{\tilde{e}(t)\}$ using model validity tests (Billings and Zhu, 1994). The results plotted in Fig.2 indicate the residual is uncorrelated after the prefiltering. The performance of the prefilter plotted in Fig.3 indicates that the prefilter is effective in retrieving signal from noisy observations. The prefiltered signal $y_p(t)$ was then used to construct the regressors $\mathbf{\Phi}(t-1)$'s for training the state space model using NLMS algorithm Eqn's.(34)-(36). The learning rate selected was $\eta = 0.15$. The training process was carried out repeatedly using the estimation data. The convergence of the mean squared error of the network $\frac{1}{N}\sum_{t=1}^{N}[y(t) - \hat{z}(t)]^2$ was plotted in Fig.4 demonstrating the convergence of the network. The mean squared error was 0.0031 at the epoch 50.

The Kalman filter algorithm (Appendix B) was finally applied over the validation data set of 500 points from $t = 500$ to $t = 1000$. The initial conditions were predetermined as $\mathbf{z}(500) = [y(500), y(499)]^T$, $\Sigma(500) = \mathbf{I}$. The statistical information of the noise was set as $\mathbf{S} = 0$, $\mathbf{Q} = \sigma_\varepsilon^2 = 0.0001$ denoting a small number, and $\mathbf{R} = \sigma_\xi^2 = 0.0031$ using the MSE of estimation data set. Note that the matrix $\mathbf{A}(t)$ is determined using the operating point dependent neurofuzzy model at each time step. The performance of the neurofuzzy Kalman filter was plotted in Fig.5 demonstrating the excellent approximation performance of the neurofuzzy network to the real system over the

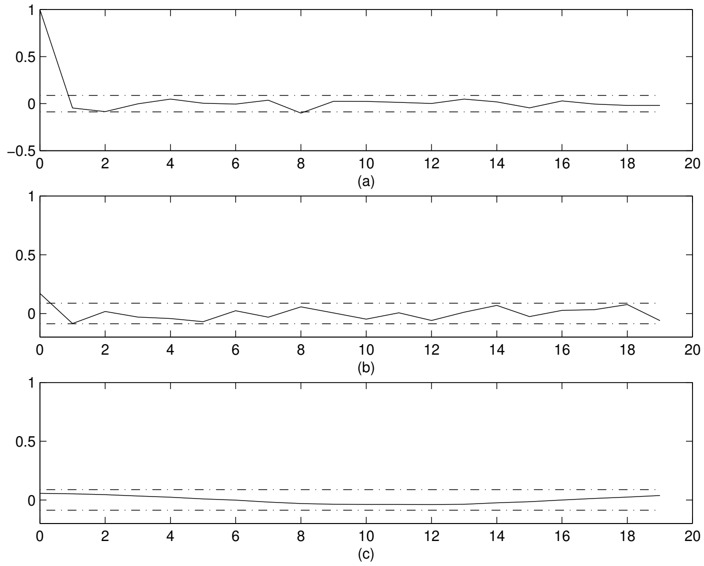

Figure 2: Model validity tests in example 1 ($\sigma_\xi = 0.05$); (a) Model validity test $\Phi_{\tilde{e}\tilde{e}}(\tau)$; (b) Model validity test $\Phi_{(y\tilde{e})\tilde{e}^2}(\tau)$ and (c) Model validity test $\Phi_{(y\tilde{e})u^2}(\tau)$

validation data set.

For the case of $\sigma_\xi = 0.08$, an order-1 ($m = 1$) B-spline neuro-fuzzy network was used to approximate the system. The input vector to the neurofuzzy network basis function was predetermined as $\mathbf{x}(t) = [y(t-1), y(t-2), u(t-1)]^T$, that is, $n_y = 2$, $n_u = 1$. The knot vectors for the input domain was defined as

$$\{-0.6, -0.5, -0.1, 0.65, 1.4, 1.8, 1.9\}$$

for system output $y(t)$, and

$$\{-1.2, -1.1, -0.9, 0, 0.9, 1.1, 1.2\}$$

for system input $u(t)$, resulting 5 one-dimensional basis functions for $y(t-1)$, $y(t-2)$ and $u(t-1)$ respectively. Note that the knot positions are slightly adjusted, compared to the case of $\sigma_\xi = 0.05$, as the output range is different due to the increase of observation noise level. The 125 multidimensional basis functions were formed using a direct multiplication of one-dimensional basis functions (Appendix A). The regression matrix $\mathbf{P}$ was then constructed according to Eq.(23), and the SVD transformation defined by Eq.(24) was performed to $\mathbf{P}$. Eqn's.(27)-(29) was then applied to construct $\mathbf{y}_p$, and the process was terminated at $M = 35$ by checking the residual vector $\{\tilde{e}(t)\}$ using model validity tests (Billings and Zhu, 1994). The results plotted in Fig.6 indicate the residual is uncorrelated after the prefiltering. The performance of the prefilter plotted in Fig.7 indicates that the prefilter is effective in retrieving signal from noisy observations. The prefiltered signal $y_p(t)$ was then used to construct the regressors $\mathbf{\Phi}(t-1)$'s for training the state space model using NLMS algorithm Eqn's.(34)-(36). The learning rate $\eta = 0.15$. The training process was carried out repeatedly using the estimation data. The convergence of the mean squared error

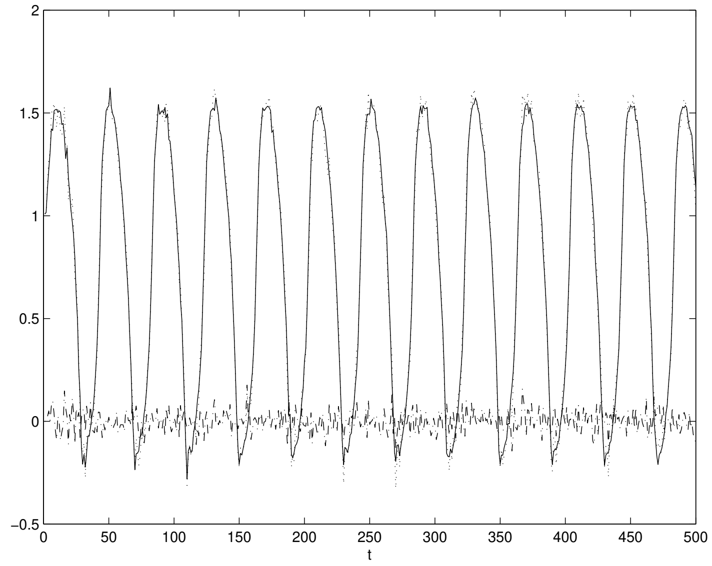

Figure 3: The performance of prefiltering over the estimation set in example 1 ($\sigma_\xi = 0.05$); (Solid Line: prefiltered signal $y_p(t)$; Dotted Line: system observations $y(t)$ and Dashdot Line: residual $\tilde{e}(t)$)

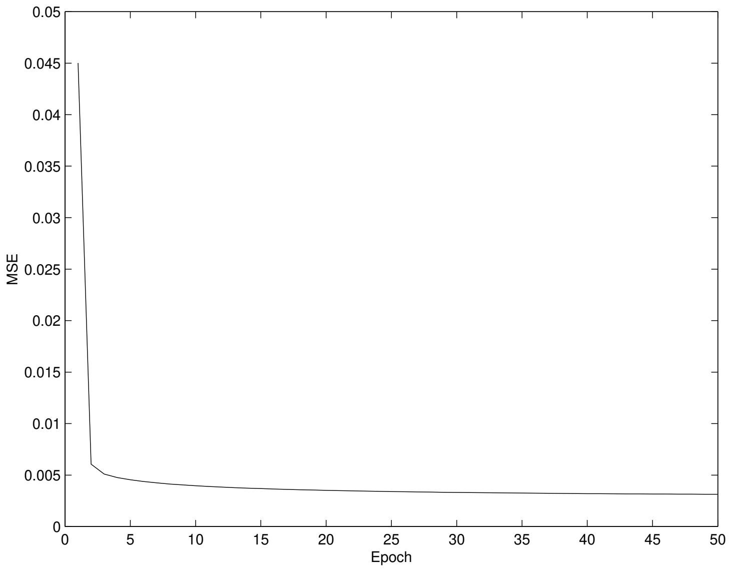

Figure 4: The MSE in the estimation set in example 1 ($\sigma_\xi = 0.05$)

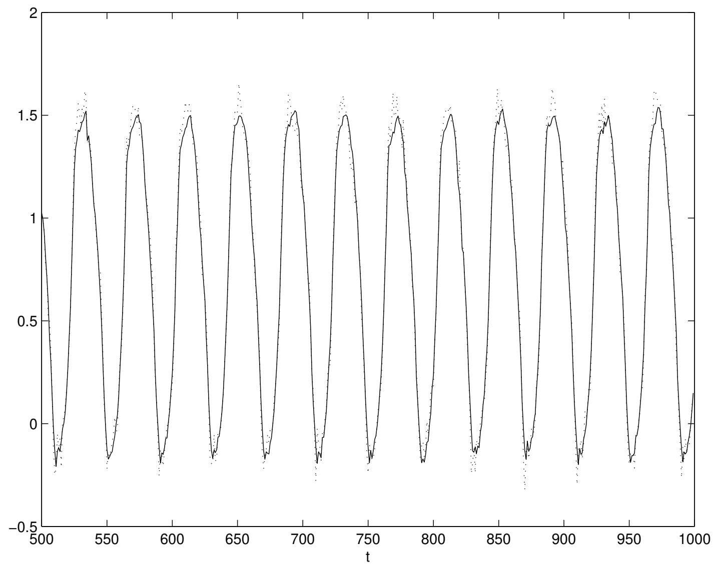

Figure 5: The state estimator on the validation set using Kalman filter algorithm in example 1 ($\sigma_\xi = 0.05$); (Solid Line: filtered state variable $z(t)$ and Dotted Line: system observations $y(t)$)

of the network $\frac{1}{N}\sum_{t=1}^{N}[y(t) - \hat{z}(t)]^2$ was plotted in Fig.8 demonstrating the convergence of the network. The mean squared error was 0.0076 at the epoch 50.

The Kalman filter algorithm (Appendix B) was finally applied over the validation data set of 500 points from $t = 500$ to $t = 1000$. The initial conditions were predetermined as $\mathbf{z}(500) = [y(500), y(499)]^T$, $\Sigma(500) = \mathbf{I}$. The statistical information of the noise was set as $\mathbf{S} = 0$, $\mathbf{Q} = \sigma_\varepsilon^2 = 0.0004$ denoting a small number, and $\mathbf{R} = \sigma_\xi^2 = 0.0076$ using the MSE of estimation data set. Note that the matrix $\mathbf{A}(t)$ is determined using the operating point dependent neurofuzzy model at each time step. The performance of the neurofuzzy Kalman filter was plotted in Fig.9 demonstrating the excellent approximation performance of the neurofuzzy network to the real system over a validation data set. By varying the noise level in the output in above example it has been shown the proposed method is robust to the effect of measurement noise.

Example 2: Consider a non-linear system

$$\begin{array}{rcl} z(t) & = & \frac{z(t-1)z(t-2)}{1+z(t-1)^2} - 0.2\cos(\pi u(t-1))z(t-2) + 1.2u(t-1) \\ y(t) & = & z(t) + \xi(t); \end{array} \tag{38}$$

where the system input $u(t) = \sin(\pi * t/50)$. A noise sequence $\xi(t) \sim N(0, 0.1^2)$ was added to the state variable generating system output observation $y(t)$. A data sequence of 1000 samples, plotted in Fig.10, was generated. The first 500 samples $\{u(t), y(t)\}$ from $t = 1$ to $t = 500$ was used as an estimation set. The Kalman filter algorithm was applied to a validation data set of 500 samples consisting of $\{u(t), y(t)\}$ from $t = 501$ to $t = 1000$. An order-1 ($m = 1$) B-spline neuro-fuzzy network was used to approximate the system. The input vector to the neurofuzzy network basis function was predetermined as $\mathbf{x}(t) = [y(t-1), y(t-2), u(t-1)]^T$, that is, $n_y = 2$, $n_u = 1$. The knot vectors for the input domain was defined as

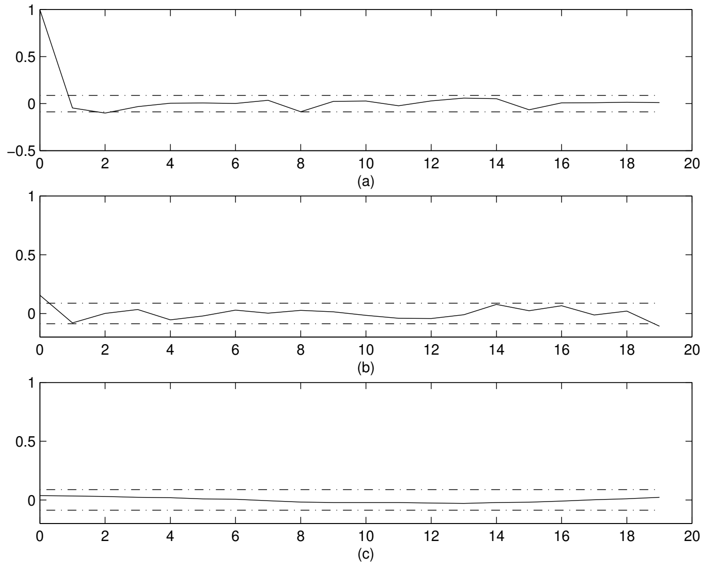

Figure 6: Model validity tests in example 1 ($\sigma_\xi = 0.08$); (a) Model validity test $\Phi_{\tilde{e}\tilde{e}}(\tau)$; (b) Model validity test $\Phi_{(y\tilde{e})\tilde{e}^2}(\tau)$ and (c) Model validity test $\Phi_{(y\tilde{e})u^2}(\tau)$

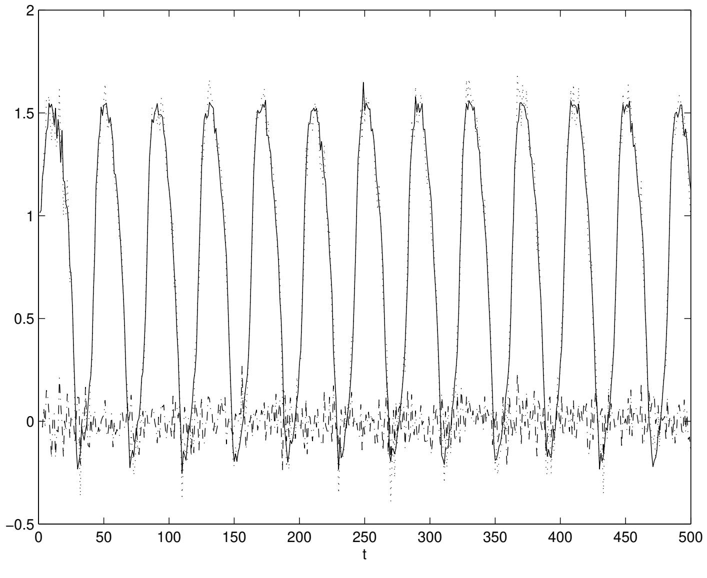

Figure 7: The performance of prefiltering over the estimation set in example 1 ($\sigma_\xi = 0.08$); (Solid Line: prefiltered signal $y_p(t)$; Dotted Line: system observations $y(t)$ and Dashdot Line: residual $\tilde{e}(t)$)

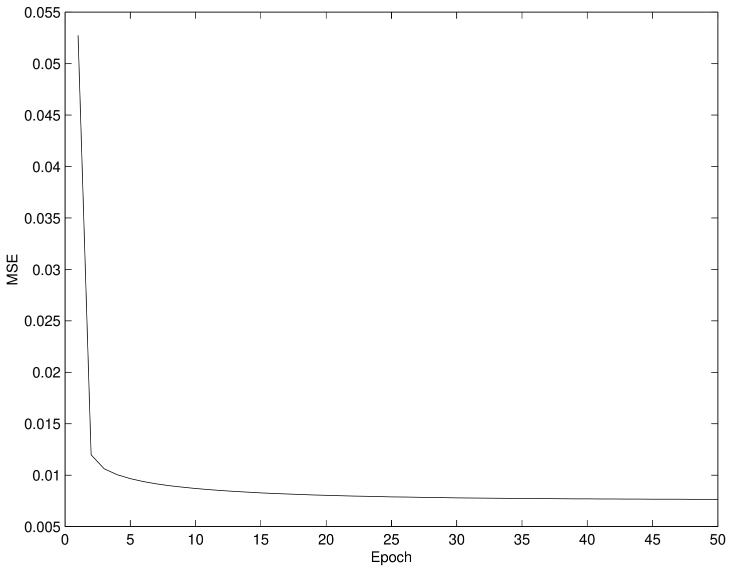

Figure 8: The MSE in the estimation set in example 1 ($\sigma_\xi = 0.08$)

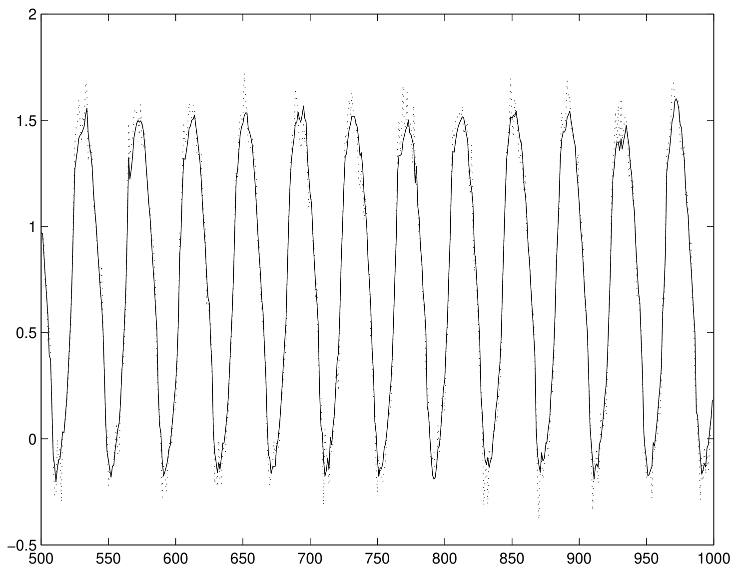

Figure 9: The state estimator on the validation set using Kalman filter algorithm in example 1 ($\sigma_\xi = 0.08$); (Solid Line: filtered state variable $z(t)$ and Dotted Line: system observations $y(t)$)

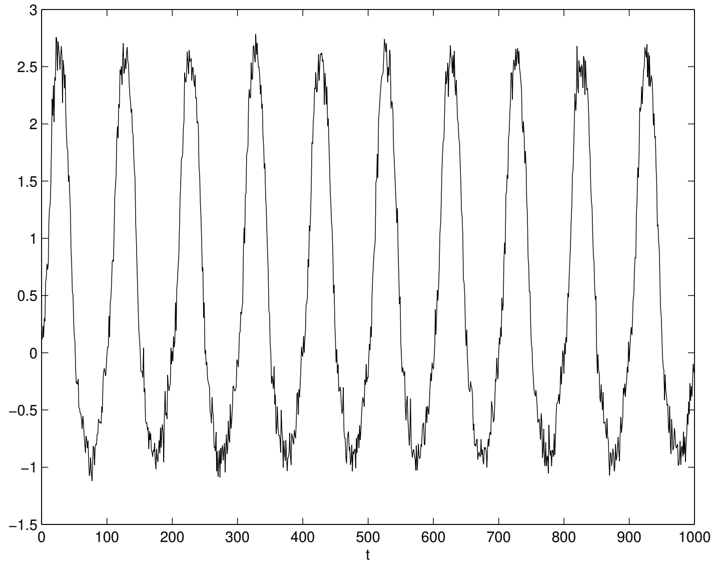

Figure 10: The output data $y(t)$ in example 2

$$\{-3, -2, -1, 0.5, 2, 3, 4\}$$

for system output $y(t)$, and

$$\{-1.2, -1.1, -0.8, 0, 0.8, 1.1, 1.2\}$$

for system input $u(t)$, resulting 5 one-dimensional basis functions for $y(t-1)$, $y(t-2)$ and $u(t-1)$ respectively. The 125 multidimensional basis functions were formed using a direct multiplication of one-dimensional basis functions (Appendix A). The regression matrix $\mathbf{P}$ was then constructed according to Eq.(23), and the SVD transformation defined by Eq.(24) was performed to $\mathbf{P}$. Eqn's.(27)-(29) was then applied to construct $\mathbf{y}_p$, and the process was terminated at $M = 25$ by checking the residual vector $\{\tilde{e}(t)\}$ using model validity tests. The results plotted in Fig.11 indicate the residual is uncorrelated after the prefiltering. The performance of the prefilter plotted in Fig.12 indicates that the prefilter is effective in retrieving signal from noisy observations. The prefiltered signal $y_p(t)$'s were then used to construct the regressors $\mathbf{\Phi}(t-1)$'s for training the state space model using NLMS algorithm Eq.(34)-Eq.(36). The learning rate $\eta = 0.15$. The training process was carried out repeatedly using the estimation data. The convergence of the mean squared error of the network $\frac{1}{N}\sum_{t=1}^{N}[y(t) - \hat{z}(t)]^2$ was plotted in Fig.13 demonstrating the convergence of the network The mean squared error was 0.011 at the epoch 20.

The Kalman filter algorithm (Appendix B) was finally applied over the validation data set of 500 points from $t = 500$ to $t = 1000$. The initial conditions were predetermined as $\mathbf{z}(500) = [y(500), y(499)]^T$, $\Sigma(500) = \mathbf{I}$. The statistical information of the noise was set as $\mathbf{S} = 0$, $\mathbf{Q} = \sigma_\varepsilon^2 = 0.0004$ as a small number, and $\mathbf{R} = \sigma_\xi^2 = 0.011$ using the MSE of estimation data set. Note that the matrix $\mathbf{A}(t)$ is determined using the operating point dependent neurofuzzy model at each time step. The performance of the neurofuzzy Kalman filter was plotted in Fig.14 once

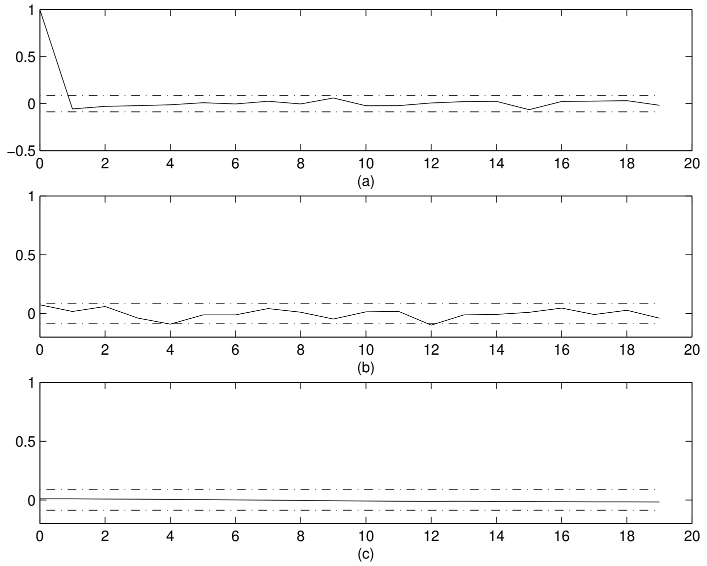

Figure 11: Model validity tests in example 2; (a) Model validity test $\Phi_{\tilde{e}\tilde{e}}(\tau)$; (b) Model validity test $\Phi_{(y\tilde{e})\tilde{e}^2}(\tau)$ and (c) Model validity test $\Phi_{(y\tilde{e})u^2}(\tau)$

again demonstrating the excellent approximation performance of the neurofuzzy network to the real system over a validation data set.

6 Conclusions

A new state estimator is introduced based on a neuro-fuzzy operating point dependent structure for state estimation using the Kalman filter algorithm. This paper examines a parameter bias problem in the identification of the state space model performed based on system input and observations. A simple effective prefiltering method has been introduced to achieve unbiased parameter estimates in the state space model which then be applied for state estimation using the classical Kalman filtering algorithm. The new prefiltering procedure has been introduced based on a non-linear principal component analysis (PCA) method using the B-spline neuro-fuzzy basis function set. A prefiltering procedure over the noisy observation data was performed followed by the identification of the system which is based on the preprocessed data instead of the noisy observations. The new non-linear prefiltering procedure utilized only noisy system observations without *prior* knowledge of the actual system structure. Model validity tests have been applied to check the effectiveness of the prefilter in retrieving signal. The Kalman filter algorithm is applied over a validation data set using the model having been determined using the prefiltering procedure. Numerical examples are included to demonstrate the effectiveness of the new approach. By invoking the certainty equivalence principle a stochastic state feedback controller can be designed on the resultant state model.

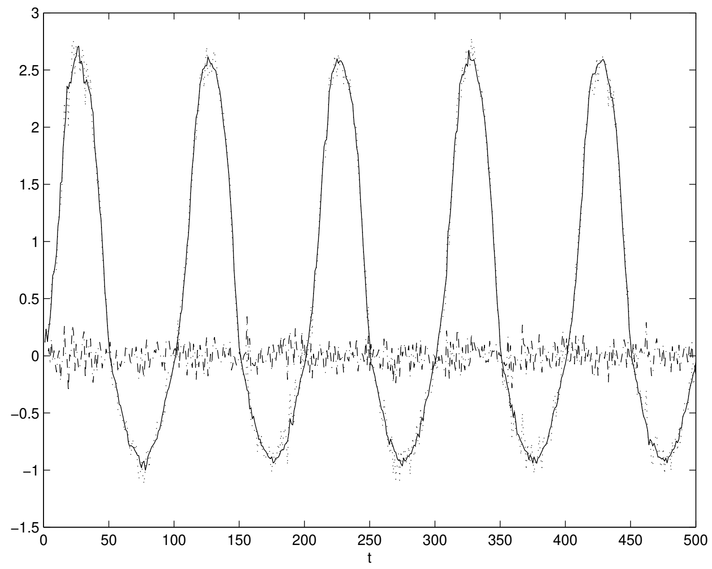

Figure 12: The performance of prefiltering over the estimation set in example 2; (Solid Line: prefiltered signal $y_p(t)$; Dotted Line: system observations $y(t)$ and Dashdot Line: residual $\tilde{e}(t)$)

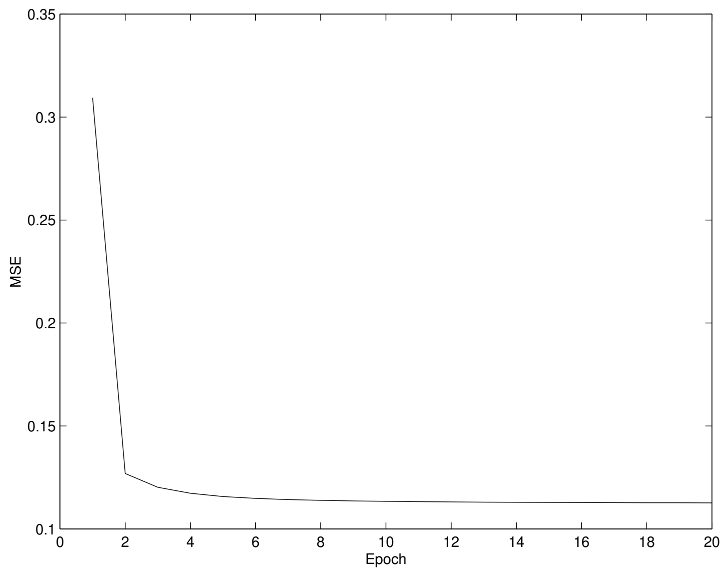

Figure 13: The MSE in the estimation set in example 2

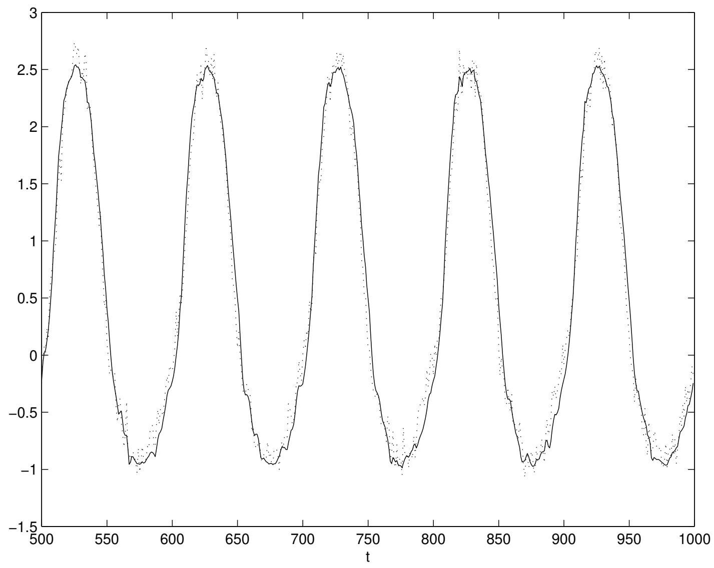

Figure 14: The state estimator on the validation set using Kalman filter algorithm in example 2; (Solid Line: filtered state variable $z(t)$ and Dotted Line: system observations $y(t)$)

7 Acknowledgement

The authors gratefully acknowledge the finacial support of EPSRC and RACAL.

Appendix A

A neurofuzzy network using B-spline function as the membership function. A general one-dimensional B-spline model is formed as a linear combination of L B-spline basis functions as

$$f(x) = \sum_{k=1}^{L} \omega_k B_{k,m}(x) \tag{39}$$

The coefficient ω_k's represent the set of adjustable parameters associated with the set of basis functions. The set of basis functions $B_{k,m}(x)$'s in the model is a polynomial of a given degree m and is uniquely defined by an ordered sequence of real values denoted as a knot vector $\tau = \{\tau_1, \tau_2, \cdots, \tau_{L+m+1}\}$. The knot sequence forms a partitioning of the input domain into $(L+m)$ disjoint intervals. The definition of basis functions set can be expressed as a recursion (Dierckx, 1995)

$$B_{k,m}(x) = \frac{x - \tau_k}{\tau_{k+m} - \tau_k} B_{k,m-1}(x) + \frac{\tau_{k+m+1} - x}{\tau_{k+m+1} - \tau_{k+1}} B_{k+1,m-1}(x) \tag{40}$$

with

$$B_{k,0} = \begin{cases} 1 & \tau_k \leq x < \tau_{k+1} \\ 0 & otherwise \end{cases}$$

The multidimensional B-spline basis functions are formed by a direct multiplication of one dimensional basis functions

$$N(\mathbf{x}) = \prod_{i=1}^{n} B_{k_i,m}(x_i) \tag{41}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T \in \Re^n$. $k_i = 1, 2, \cdots, L_i$, L_i is the number of B-spline basis functions defined in x_i, the ith component of $\mathbf{x}$. The multidimensional B-spline basis function models are constructed as a tensor product of one-dimensional models using $p = \prod_{i=1}^{n} L_i$ basis functions as

$$\begin{aligned} f(\mathbf{x}) &= \sum_{k=1}^{p} \omega_k N_k(\mathbf{x}) \\ &= \sum_{k=1}^{p} \omega_k \prod_{i=1}^{n} B_{k_i,m}(x_i) \end{aligned} \tag{42}$$

Appendix B

Consider the state space model of Eq.(9) and (10). Let $\hat{\mathbf{z}}(t+1)$ denote the conditional mean of $\mathbf{z}(t+1)$ given observations of $\{y(t)\}$ up to and including time t. Denote

$$E\left\{ \begin{bmatrix} \varepsilon(t) \\ \xi(t) \end{bmatrix} \left[\varepsilon(\tau)^T \xi(\tau)^T \right] \right\} = \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^T & \mathbf{R} \end{bmatrix} \delta(t - \tau)$$

where $\delta(\bullet)$ denotes *kronecker* delta. Then $\hat{\mathbf{z}}(t+1)$ satisfies the following recursion (the Kalman filter):

$$\hat{\mathbf{z}}(t+1) = \mathbf{A}(t)\hat{\mathbf{z}}(t) + \mathbf{K}(t)\left[y(t) - \mathbf{C}\hat{\mathbf{z}}(t)\right] + \mathbf{B}\tilde{u}(t) \tag{43}$$

$$\mathbf{K}(t) = \left[\mathbf{A}(t)\Sigma(t)\mathbf{C}^T + \mathbf{S}\right]\left[\mathbf{C}\Sigma(t)\mathbf{C}^T + \mathbf{R}\right]^{-1} \tag{44}$$

$$\Sigma(t+1) = \mathbf{A}(t)\Sigma(t)\mathbf{A}(t)^T + \Gamma\mathbf{Q}\Gamma^T - \mathbf{K}(t)\left[\mathbf{C}\Sigma(t)\mathbf{C}^T + \mathbf{R}\right]\mathbf{K}(t)^T \tag{45}$$

with the initial conditions given as

$$\begin{aligned} \mathbf{z}(t_0) &= \mathbf{z}_0 \\ \Sigma(t_0) &= \Sigma_0 \end{aligned} \tag{46}$$

where $\mathbf{K}(t)$ is the Kalman gain. $\Sigma(t)$ is the state error variance, that is,

$$\Sigma(t) = E\{[\hat{\mathbf{z}}(t) - \mathbf{z}(t)][\hat{\mathbf{z}}(t) - \mathbf{z}(t)]^T | y(t-1), y(t-2), \cdots\} \tag{47}$$

References

[1] Billings, S. A., Chen, S. and Korenberg, M. J. (1989). Identification of MIMO non-linear systems using a forward-regression orthogonal estimator. *Int. J. Control*, Vol. 49, No. 6, pp2157-2189.

[2] Billings, S. A. and Zhu, Q. M. (1994) Nonlinear model validation using correlation tests. *Int. J. of Control*, Vol. 60, pp1107-1120.

[3] Bossley, K. M. (1997). *Neurofuzzy Modelling Approaches in System Identification.* Ph.D Thesis, Dept of Electronics and Computer Science, University of Southampton.

[4] M. Brown and Harris, C. J. (1994). *Neurofuzzy Adaptive Modelling and Control.* Prentice Hall, Hemel Hempstead.

[5] C. K. Chui and Chen, G. (1987). *Kalman Filtering with Real-Time Applications.* Springer Series in Information Science, Berlin Heidelberg.

[6] Dierckx, P. (1995). *Curve and Surface Fitting with Splines.* Monographs on Numerical Analysis , Clarendon Press, Oxford.

[7] R. S. Doyle and Harris, C. J. (1996). Multi-sensor data fusion for helicopter guidance using neuro-fuzzy estimation algorithms. *The Royal Aeronautical Society Journal.* June/July, pp241-257.

[8] M. Feng and Harris, C. J. (1997). Adaptive neurofuzzy control for a class of state dependent nonlinear processes. To appear in *Int. J. of Systems Science.*

[9] J. F. Gibson, J. D. Farmer, M. Casdagli and Eubank, S. (1992). An analytic approach to practical state space reconstruction. *Physica D.* **57**, pp1-30.

[10] G. C. Goodwin and Sin, K. S. (1984). *Adaptive Filtering Prediction and Control.* Information and System Science Series, Prentice Hall, Englewood Cliffs, New Jersey.

[11] Kavli, T. (1993). ASMOD - an algorithm for adaptive spline modelling of obervation data. *Int. J. of Control.* Vol. 58, No. 4, pp947-967.

[12] Papoulis, A. (1965). *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, New York.

[13] Priestley, M. B. (1980). State-dependent models: A general approach to non-linear time series Analysis. *J. Time Series Analysis*, Vol. 1, No. 1, pp47-71.

[14] Webb, A. R. (1993). An approach to non-linear principal components analysis using radial basis functions. *Defence Research Agency Malvern Memorandum* 4739.