Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic Systems

Sunil Elanayar V.T. and Yung C. Shin

Abstract— **This paper presents a means to approximate the dynamic and static equations of stochastic nonlinear systems and to estimate state variables based on Radial Basis Function Neural Network (RBFNN). After a nonparametric approximate model of the system is constructed from *a priori* experiments or simulations, a suboptimal filter is designed based on the upper bound error in approximating the original unknown plant with nonlinear state and output equations. The procedures for both training and state estimation are described along with discussions on approximation error. Nonlinear systems with linear output equations are considered as a special case of the general formulation. Finally, applications of the proposed RBFNN to the state estimation of highly nonlinear systems are presented to demonstrate the performance and effectiveness of the method.**

I. Introduction

THE PROBLEM of finding an exact or approximate model for dynamical systems occurs frequently in engineering applications. One way of achieving this goal is through system identification. In general, identification of nonlinear dynamic systems is approached via parametric or nonparametric methods. Parametric methods of identification assume problem dependent nonlinear functions and then identify the parameters appearing in these functions based on input-output data. These models known as prediction error models [1] employ least squares or maximum-likelihood means to estimate the parameters. Nonparametric Kernel based methods [2] do not assume knowledge of the system structure. They are, however, unsuitable for control applications. In all of these methods, the objective is to determine a model that best describes the input-output behavior of the data. Many physical systems, however, exhibit distinct plant and sensor characteristics that have to be separately modelled. In many cases when it is difficult to obtain a model structure for a system, or the available model is highly nonlinear so that estimating its states is very difficult, generic tools are desired that can approximately describe experimental data. Artificial neural networks present one possibility for achieving this. Narendra and Parthasarthy [3] laid the foundation for system identification and control using the globally approximating characteristics of neural networks. They emphasized the need to view static and dynamic representations in a unified fashion.

This paper presents a method to approximately identify discrete dynamic systems on the basis of multiple experiments using the RBFNN. More than one data set is used in order to ensure adequate learning by the neural network over the domain of interest. A modified recursive least squares training algorithm is employed to obtain the weights of the neural networks. Once trained, the neural network can be used on-line for estimating the states of the unknown nonlinear system. This requires the construction of a state estimator (filter) based on the neural network.

The problem of state estimation of a stochastic system based on the measurement data has an exact solution only when the plant and measurement equations are linear. Then the *a posteriori* density is Gaussian and the conditional mean and covariance are described by the well known Kalman filter equations. In contrast, most practical solutions to the nonlinear filtering problem are approximate in nature. Nonlinear filtering has been the subject of intensive research over the past three decades and several schemes to this end have been proposed and developed such as extended Kalman filters, statistically linearized filters, various methods based on global and local linearization [4]. Extended Kalman filter methods [5] rely on linearized state and output equations (along a nominal trajectory) to estimate the state. They, therefore, require exact models of underlying nonlinear systems that are in reality difficult to obtain in many cases and can lead to divergence when modeling error exists [6]. Statistically linearized filters were used for developing suitable approximate solutions to problems of estimation and control. These yielded better results than the Taylor series based extended Kalman filter [7].

Direct application of these methods to the neural network models, however, would not yield acceptable results or could even lead to divergence due to the inaccuracy of the RBFNN in representing the true nonlinear plant. In order to overcome this difficulty, an upper bound estimate for the covariance matrix is used to obtain minimum-variance state estimates in the new approach. The upper bound covariance computation explicitly includes approximation errors resulting from replacing the original systems with a RBFNN. Based on the upper bound error, a suboptimal filter is designed and used to estimate the state variables of the system. Derivation of the filter gain is an extension of the approach used by Gusak [8]. It is shown

Manuscript received January 16, 1992; revised October 8, 1992.

The authors are with the School of Mechanical Engineering, Purdue University, West Lafayette, IN 47907 USA.

IEEE Log Number 9205933.

through simulations that the constructed filter provides good state estimates.

The remainder of this paper is organized in the following way. First, the underlying problem is defined in a rigorous manner. In the following section, the architecture of the RBFNN is defined and then the training algorithm for the RBFNN is described. Section IV describes the procedure for the state estimation of nonlinear systems by using the trained RBFNN. Finally, Section V demonstrates application of the proposed RBFNN to the state estimation of selected highly nonlinear models.

II. Statement of the Problem

Consider a nonlinear discrete state-space model with unknown structural functions. Let the plant dynamics be as follows

$$x_{k+1} = f(x_k, u_k) + w_k \tag{1}$$

and the measurement equations given by

$$y_k = h(x_k) + v_k \text{ for } k = 0, 1, 2, \cdots. \tag{2}$$

The state vector $x_k = (x_{1k}, \cdots, x_{nk})$ is n-dimensional and the output vector denoted by y_k is m-dimensional. For simplicity, inputu_k is assumed to be a scalar. However, the results obtained are equally valid for multi-input systems. State and measurement noise vectors w_k and v_k are assumed to be independent, Gaussian white processes with zero mean such that $E[w_k w_j^T] = W\delta_{kj}$ and $E[v_k v_j^T] = V\delta_{kj}$. In addition, the initial first and second moments are given by $E[x_0] = \mu_0, E[(x_0 - \mu_0)(x_0 - \mu_0)^T] = \rho_0$.

The objective is to obtain an approximate model for the plant dynamic and static equations (1) and (2), which is referred to as the training stage. Then the trained model is to be used for estimating the states of the original unknown plant. In order to further develop the approach, the following assumptions are required:

1) The states and input remain bounded in an open domain of $\mathbb{R}^{n+1}$ denoted by D, i.e.,

$$(x_k, u_k) \in D \quad \forall k$$

Note that this assumption is more constraining than bounded input-bounded output (BIBO) stability.

2) The functions $f(.,.)$ and $h(.)$ are unknown vector functions that are assumed to satisfy global continuity conditions on D.

$$\begin{aligned} \mathcal{F}: \{f : \|f(x + \delta_x, .) - f(x, .)\| \le K_f \|\delta_x\|\} \\ \{f : \|f(., u + \delta_u) - f(., u)\| \le K_u \|\delta_u\|\} \\ \{h : \|h(x + \delta) - h(x)\| \le K_h \|\delta\|\} \end{aligned} \tag{3}$$

where $x, x + \delta_x, u, u + \delta_u$ belong to the domain of definition of the functions.

3) During training, all the state variables are assumed to be measurable with additive noise. This would correspond to the stage when experiments are performed on the system or simulations are carried out to obtain an approximate model of the system.

4) The covariance matrices of the process and measurement noises are assumed known.

5) The input is assumed to be exactly measurable at all times.

Certain additional assumptions are required in achieving proper convergence of the weights of the RBFNN. This aspect will be addressed in the next section.

III. Non-Parametric Approximation of the Model

For the nonlinear dynamical systems represented by (1) and (2), the goal is to obtain a general non- parametric approximation over the domain D. This approximation is to be global over the region D. As mentioned above, it is assumed that explicit expressions for the state transition functions and output equations are unavailable. The static and dynamic equations are replaced by generic approximating functions. This section develops a constructive method for achieving this based on multiple experiments. Let us first consider the problem of approximating $f(.,.)$. It is known that for nonlinear dynamic systems, the determination of a sequence $\{u_k\}$ for the plant to have a desired trajectory is a difficult task even when the function $f(.,.)$ is known and all states are accessible [3]. Hence, to achieve global approximations over D, multiple experiments (with possible variations in the input sequences) are assumed to be available. Another reason for the use of multiple experiments is to account for variations in initial conditions.

The vector functions $\hat{f}(.,.)$ and $\hat{h}(.)$ are used in the sequel to denote the approximations. In addition, define $X_k^i = [x_k^i, u_k^i]$ as the $n+1$ vector of states appended with the input for the ith experiment. Then, the problem of approximately identifying $f(.,.)$ can be stated in terms of minimizing a least squares criterion. Given N sequential observations from each of M different experiments $\{E_i : i = 1, \cdots, M\}$, and associated observations $\{z_k^1, z_k^2, \cdots, z_k^M\}, z_k^i = x_k^i + \zeta_k^i$, it is desired to obtain an approximated model, such that

$$J = \sum_{i=1}^{M} \sum_{k=2}^{N} [z_k^i - \hat{f}(x_{k-1}^i, u_{k-1})]^T [z_k^i - \hat{f}(x_{k-1}^i, u_{k-1})] \tag{4}$$

is minimized. The vector function $\hat{f}(.,.)$ is required to be globally approximating over region D. Additive noise in observing x_k^i is interpreted as the error in approximating $f(.,.)$. In this paper, the vector functions are chosen to be represented in terms of radial basis functions.

Radial basis functions have been used as a technique for multivariate scattered data interpolation and recently attracted attention in the neural network community [9], [10], [11]. Similar to backpropagation neural networks, radial basis functions possess the properties of approximating nonlinear functions of several variables. Unlike the former, radial basis function expansions have a "linear in the parameters" representation. As a result, convergence properties of the parameters can be guaranteed. Although the original applications of radial basis functions was in interpolation, a definition in the context of approximation is introduced here [12]:

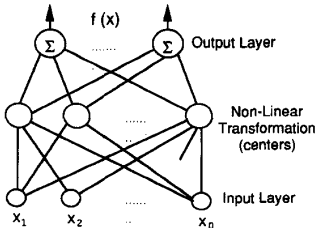


Fig. 1. Schematic diagram of radical basis function expansion.

Definition: Given a continuous function $F : \mathbb{R}^+ \rightarrow \mathbb{R}$, *and points* $\{X_j^c : j = 1, \cdots, p\}, X_j^c \in \mathbb{R}^+$, *which become dense in the open region D of* $\mathbb{R}^+$, *there exists a sequence of functions*

$$t_p(X) = \sum_{j=1}^{p} \lambda_j^p \Phi(\|X - X_j^c\|) + \lambda_0^T X$$

and some bounded open domain on which

$$|t_p(X) - F(X)| \rightarrow 0$$

as $p \rightarrow \infty$, *where* $|.|$ *is the Euclidean norm. The functions* $\Phi(.)$ *are termed radial basis functions, and the prototypical points* X_i^c *are referred to as the centers of the basis functions. The argument of the basis functions is a scalar given by* $\triangleq \|X_j - X_j^c\|$.

The radial basis function approach constructs an approximation based on the locations of the data points. In typical applications, they are used to approximate a continuous function $F(.)$ based on scattered sampling pairs $(X_j, F(X_j))$, where $X_j \in D$. A linear term with the parameters λ_0 has been added based on simulation experiences of the authors. Note that, addition of the linear term does not affect the convergence proofs given by [12]. Different types of functions $\Phi(.)$ satisfy the requirements of this definition. Some choices found in the literature include thin plate splines $(\Phi(r) = r^2 \log r)$ by Dunchon [13], Hardy multiquadratics $(\Phi(r) = \sqrt{c + r^2})$ by Hardy [14], $\Phi(r) = r^l$ (l an odd integer) [15], $\Phi(r) = \exp(-r^2)$ by Schagen [16], etc. Regardless of the choice of the radial basis function, the approximated model assumes a linear form in terms of parameters λ_j^p's, if it is assumed that the centers of the basis functions are fixed beforehand. Based on these basis (activation) functions and parameters (weights), the paradigm "Radial Basis Function Neural Network (RBFNN)" can be constructed to provide a mapping in domain D as shown in Fig. 1. In this paradigm, neurons are represented by radial basis functions with the centers X_i^c, which are interconnected by weights. With a prespecified choice of centers, the structure resembles a single layered neural network, where each node of the hidden layer performs a nonlinear transformation specified by the basis function. There are $p + 1$ nodes in the hidden layer, p of which are assigned a center and have an activation function given by $\Phi(.)$. One node of unity activation is used to denote the linear term. One of the inherent properties of radial basis functions in interpolation is that the parameters are uniquely defined if the coefficient matrix

$$A_{ij} = \Phi(\|X_i - X_j\|) \tag{5}$$

is nonsingular.

In the present work, we have chosen the Hardy multiquadratics (MQ) $\Phi(r) = \sqrt{c + r^2}, c > 0$ due to the additional flexibility provided by a choice of the constant c. In usual practice, c is chosen to be of the same order as the spacing between the independent data points. The MQ basis functions can be considered as solutions of Laplace equations. Therefore, when scattered data are fitted by a linear combination of these functions, the resulting equations are amenable to iterative solution techniques if certain finite difference approximations to the iterated Laplacian are applied [17]. One remarkable property of the MQ method is that if data points are sufficiently scattered, the coefficient matrix A in (5) is always nonsingular for all positive integers p and n [18], thereby guaranteeing uniqueness of solution. Additional useful properties of radial basis functions in general are that no assumptions are made regarding spacing of the independent variable data. It is possible to construct sufficiently smooth approximations by additional polynomial terms, and hence enhance noise rejection capabilities. Further details and convergence proofs are available in the literature ([12], [18], and [19]). Based on the above discussion, it is assumed that a true set of parameters, and a given number of basis functions ***(p)*** exist that will completely approximate $f(.,.)$ and $h(.)$ in the domain D.

A. Dynamic System Modeling by Radial Basis Function Neural Network

During training, the dynamic portion of the nonlinear state space model given in (1) and (2) can now be represented using the RBFNN as, including a noise term (for approximation error),

$$\begin{aligned} x_{k+1}^i &= [\Lambda \ \Lambda_0] \begin{bmatrix} \Psi(X_k^i) \\ X_k^i \end{bmatrix} + w_k^i \\ z_k^i &= x_k^i + \zeta_k^i \end{aligned} \tag{6}$$

where the superscript i denotes the experiment index. It is assumed that the initial conditions x_0^i belong to the domain D. Note that X_k contains the state variables and input, and $\Psi(X_k^i) = [\Phi_1(X_k^i), \cdots, \Phi_p(X_k^i)]^T$ contains the basis functions corresponding to p centers. Each row of the matrices Λ and Λ_0 correspond to an element of the vector function $\hat{f}(.)$. Fig. 2 shows a schematic diagram for approximating the complete dynamic stochastic system given by (1) and (2). If the vector $\theta_j^T, j = 1, \cdots, n$, is used to denote the jth row of the matrix $[\Lambda \ \Lambda_0]$, (6) can be rewritten as:

$$x_{k+1}^i = \begin{bmatrix} \Psi_{ik}^T & 0 & \cdots & 0 \\ 0 & \Psi_{ik}^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \Psi_{ik}^T \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} + w_k^i \tag{7}$$

for $i = 1, \cdots, M$. The notation $\Psi_{ik} \triangleq [\Psi^T(X_k^i) \ X_K^{iT}]^T$ is used in the above. Further simplification of the notation using $\Theta = [\theta_1 \theta_2 \cdots \theta_n]^T$, and combining all M experiments, a system with Mn state variables and np parameters

$$\eta_{k+1} = \xi_k(\eta_k, u_k)\Theta + w_k \tag{8}$$

with measurements

$$\gamma_k = \eta_k + \zeta_k \tag{9}$$

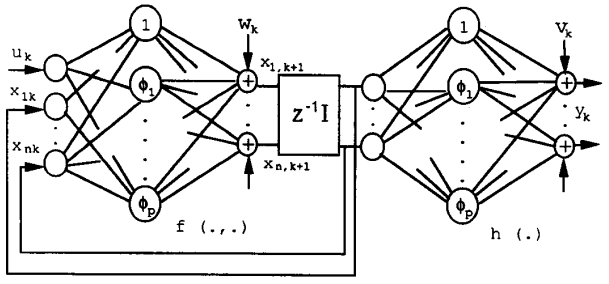


Fig. 2. Schematic of the RBFNN for dynamic nonlinear systems.

is obtained, where $\eta_k^T = [x_{1k}^1, x_{1k}^2, \cdots, x_{1k}^M, x_{2k}^1, \cdots, x_{nk}^M]$. In the following, knowledge of the statistical properties of ζ_k is not assumed. The matrix

$$\xi_k(\eta_k, u_k) = \begin{bmatrix} \begin{pmatrix} \Psi_{1k}^T \\ \vdots \\ \Psi_{MK}^T \end{pmatrix} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \begin{pmatrix} \Psi_{1k}^T \\ \vdots \\ \Psi_{MK}^T \end{pmatrix} \end{bmatrix} \tag{10}$$

is of dimension $Mn \times np$.

B. Training of the RBFNN

Once the structural functions of the plant (1) have been replaced with an equivalent system of radial basis functions, the training algorithm for the RBFNN is formulated as one of parameter identification. Identification of systems in the form of (8) and (9) is well known in control theory. Note that, as a result of combining the M experiments, the objective function in (4) can be written as

$$J = \sum_{k=1}^{N} [\gamma_k - \xi_{k-1}(\eta_{k-1}, u_{k-1})\Theta]^T [\gamma_k - \xi_{k-1}(\eta_{k-1}, u_{k-1})\Theta] \tag{11}$$

Equating the gradient with respect to Θ to zero gives the least squares estimation rule

$$\hat{\Theta}_N = \left[\sum_{k=2}^{N} \xi_{k-1}^T \xi_{k-1}\right]^{-1} \left[\sum_{k=2}^{N} \xi_{k-1}^T \gamma_k\right] \tag{12}$$

However, the strict validity of (12) for parameter identification of nonlinear systems cannot be uniformly guaranteed. Here, additive noise terms in both state and output measurements are considered. In order to improve convergence of the parameter estimation, the algorithm outlined by Knapp and Pal [20] is adopted here:

1) Since η_k is not exactly measurable, replace the matrix $\xi_{k-1}(\eta_{k-1}, u_{k-1})$ with $\hat{\xi}_{k-1}(\eta_{k-1}, u_{k-1})$.
2) An instrumental variable step of replacing ξ_{k-1}^T by ξ_{k-2}^T ensures consistent estimates of the parameter Θ. As a result, the estimation rule (12) can be rewritten as

$$\hat{\Theta}_N = \left[\sum_{k=3}^{N} \hat{\xi}_{k-2}^T \xi_{k-1}\right]^{-1} \left[\sum_{k=3}^{N} \hat{\xi}_{k-2}^T \gamma_k\right] \tag{13}$$

It is known [20] that the convergence of the least squares algorithm ($\hat{\Theta}_N \to \Theta$ in probability) depends on the satisfaction of two additional conditions, namely,

$$\frac{1}{N} \sum_{k=3}^{N} \hat{\xi}_{k-2}^T \xi_{k-1}^T \to \sum \tag{14}$$

and

$$\frac{1}{N} \sum_{k=3}^{N} \hat{\xi}_{k-2} \left[\gamma_k - \hat{\xi}_{k-1}\Theta\right] \to 0. \tag{15}$$

These conditions are difficult to verify for a general matrix ξ_k, but they will be assumed to be true. The validity of these assumptions will be further investigated through simulation analysis presented in a later section. The recursive version of algorithm (13) is written as follows:

$$\hat{\Theta}_{N+1} = \hat{\Theta}_N + R_N \hat{\xi}_{N-1}^T \left[I + \hat{\xi}_N R_N \hat{\xi}_{N-1}^T\right]^{-1} \times \left[\gamma_{N+1} - \hat{\xi}_N \hat{\Theta}_N\right] \tag{16}$$

where

$$R_{N+1} = R_N - R_N \hat{\xi}_{N-1}^T \left[I + \hat{\xi}_N R_N \hat{\xi}_{N-1}^T\right]^{-1} \hat{\xi}_N R_N \tag{17}$$

Faster convergence of the training algorithm is obtained by choosing $R_0 = \sigma I$, with a sufficiently large value of σ. Approximation of the static output equations can be carried out in an analogous fashion.

C. Approximation Error Estimation

In order to use the RBFNN for state estimation, the maximum error in approximation has to be estimated. Define the maximum error in approximating $f(.,.)$ as

$$e_f = \|f(X) - \hat{f}(X)\|_\infty, \quad X \in D$$

where $\|e_1, \cdots, e_n\|_\infty = Max\{|e_1|, \cdots, |e_n|\}$. When the process and measurement noises are small, an estimate of e_f can be obtained from the M experiments using

$$e_f = Max\|\gamma_k - \hat{\xi}_{k-1}(\gamma_{k-1}, u_{k-1})\hat{\Theta}\| \quad k = 1, 2, \cdots \tag{18}$$

Similarly, the constant e_h is defined as the approximation error for the output equation. To proceed, let system in (6) be rewritten as

$$x_{k+1} = f'(x_k, u_k) + Fx_k + bu_k + w_k \tag{19}$$

where matrix F and vector b are obtained from the matrix Λ_0. The approximated output equations can also be rewritten as

$$y_k = h'(x_k) + Hx_k + \nu_k \tag{20}$$

Two Lipschitz constants a and d are defined as

$$\|f'(x_k, u_k) - f'(x_k + \delta_k, u_k)\|_\infty \leq a\|\delta_k\|_\infty \tag{21}$$

and

$$\|h'(x_k + \delta_k) - h'(x_k)\|_\infty \leq d\|\delta_k\|_\infty \tag{22}$$

These constants a and d can also be estimated using the training set as was done with the constants e_f and e_h.

The training algorithm for the RBFNN is summarized as follows:

1) Choose p vectors from among the M training experiments in a random fashion. For the dynamic portion, vectors of dimension $n+1$ are chosen at random times k.
2) Based on the M training experiments, and, using (16) and (17), the network parameters for the dynamic and static equations are identified.
3) Upper bound errors of approximation are estimated from the training data as given by (18). Lipschitz constants a and d are estimated using (21) and (22).

In the next section, the approximated system given by (19) and (20) is used to design a minimum variance filter for on-line estimation of state variables.

IV. State Estimation Using the RBFNN

To estimate the states of a system represented by the RBFNN, a nonlinear filter is to be designed. Performance of the filtering system depends on the accuracy of the model used. If modeling inaccuracies are not included in the covariance calculations, the calculated covariance matrix becomes unrealistically small, so that undue confidence is placed in the estimates [6]. The estimator ignores subsequent measurements and this results in large estimation errors. This "divergence" problem can be minimized by accounting for modelling errors.

In order to circumvent this problem, we will use information about the maximum approximation errors in the covariance calculations. An upper bound covariance matrix is obtained, such that minimizing the estimated variance results in minimizing the true variance. For convenience, the approximated system using the RBFNN is rewritten as

$$x_{k+1} = f'(x_k, u_k) + Fx_k + bu_k + w_k \tag{23}$$

$$y_k = h'(x_k) + Hx_k + \nu_k \tag{24}$$

Consider a state estimator of the following type

$$\hat{x}(k+1) = f'(\hat{x}_k, u_k) + F\hat{x}_k + bu_k + K_k[y_k - h'(\hat{x}_k) - H\hat{x}_k] \tag{25}$$

Define the estimation error as $\tilde{x}_k \triangleq x_k - \hat{x}_k$ and the error in approximating the system dynamic and static equations by $\tilde{f} \triangleq f(x_k, u_k) - f'(x_k, u_k) - Fx_k - bu_k$ and $\tilde{h} \triangleq h(x_k) - h'(x_k) - Hx_k$. The true covariance matrix of the system is denoted by $\rho_k \triangleq E[\tilde{x}_k \tilde{x}_k^T]$. Based on the derivation given in the Appendix, a recursive upper bound for the covariance matrix is given by the following equation

$$\begin{aligned}\hat{P}_{k+1} = {} & l_1(F - K_kH)\hat{P}_k(F - K_kH)^T + l_2 I + l_3 Tr(\hat{P}_k)I \\ & + l_4 K_k K_k^T + l_5 Tr(\hat{P}_k)K_k K_k^T + W + K_k V K_k^T\end{aligned} \tag{26}$$

with

$$\hat{P}_0 = \rho_0.$$

The constants $l_1 - l_5$ are as defined in the Appendix, and ρ_0 is the known initial covariance matrix. To obtain the minimum variance gain matrix K_k^* we set the first variation of $\hat{P}_{k+1}$ with respect to K_k to 0 [21]. Thus

$$-l_1(F - K_kH)\hat{P}_k H^T \delta K_k^T + K_k[l_4 I + l_5 Tr(\hat{P}_k)I + V]\delta K_k^T = 0.$$

Since δK_k is arbitrary, the optimal gain is given by

$$K_k^* + F\hat{P}_k H^T \left[\left(\frac{l_4}{l_1} + \frac{l_5}{l_1} Tr(\hat{P}_k)\right) I + \frac{1}{l_1} V + H\hat{P}_k H^T\right]^{-1} \tag{27}$$

The following theorem based on the results in [8] establishes the required properties for the assumed covariance equations. In the following, the matrix inequality $Q_1 \geq Q_2$ is used to mean that $Q_1 - Q_2$ is positive semi-definite (PSD).

Theorem 1: Given the existence of an approximated dynamic system in (23) and (24) that satisfies approximation error conditions of (18), and Lipschitzean continuity conditions in (21) and (22), then:

1) $\hat{P}_k \geq \rho_k$;
2) $\hat{P}_{k+1}(K_k) \geq \hat{P}_{k+1}(K_k^*) \geq \rho_{k+1}$, *where the optimal gain is given by (27).*

Proof: To prove (1), consider the inequality obtained from (26) and (36):

$$\begin{aligned}\hat{P}_{k+1} - \rho_{k+1} \geq {} & l_1(F - K_kH)(\hat{P}_k - \rho_k)(F - K_kH)^T \\ & + l_3 Tr(\hat{P}_k - \rho_k)I + l_5 Tr(\hat{P}_k - \rho_k)K_k K_k^T\end{aligned} \tag{28}$$

Since $\hat{P}_0 = \rho_0$, and the right hand side of the above inequality is PSD, (1) follows by induction. Note that, the right hand side of the above inequality is PSD for any gain matrix. Part (2) of the theorem follows from the above argument and derivation of the optimal gain by minimizing $\hat{P}_{k+1}$.

The above derivation of a state estimator for the RBFNN applies to very general nonlinear stochastic systems. For a wide class of problems encountered in practice, the output equations occur in a linear fashion. Furthermore, uncertainties in the output equation are usually less severe than in the dynamic model equations [22]. For such applications a simplified filter can be derived as was done in the more general case. Consider an output equation that is linear and known accurately

$$y_k = Hx_k + \nu_k \tag{29}$$

In this case, the estimator is specified by the equations

$$K_k^* = (2+e_f)F\hat{P}_k H^T\left[(2+e_f)H\hat{P}_k H^T + V\right]^{-1} \quad (30)$$

and

$$\begin{aligned}\hat{P}_{k+1} =&(2+e_f)(F-K_kH)\hat{P}_k(F-K_kH)^T \\ &+a(1+a+e_f)Tr(\hat{P}_k)I+ne_f(1+e_f+a)I \\ &+W+K_kVK_k^T. \end{aligned} \quad (31)$$

In this section a state estimator for the RBFNN based on the upper bound errors in approximation has been derived. Additional terms in the gain equation result from taking into account these modelling errors. The next section presents a few example applications of the RBFNN to approximate the static and dynamic equations and to subsequent state estimation.

V. Example Applications of the RBFNN State Estimator

In this section, application of the method to three practical examples will be demonstrated. It is not possible to compare the method with any existing methods because of the nature of assumptions made in deriving the estimator. In each of the following examples, the number of basis in the RBFNN was chosen to be 75. Training was carried out until the mean squared errors were below a prespecified value.

Example 1: This example considers the estimation of the altitude and velocity of a vertically falling body with an unknown ballistic coefficient [22]. Radar measurements are corrupted with additive Gaussian white noise. The nonlinear model and output equations are as shown below:

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} - Tx_{2k} \\ x_{2k} - 3Tx_{2k}^2 e^{-0.05x_{1k}} \end{bmatrix} \quad (32)$$

$$y_k = \sqrt{10000 + x_{1k}^2} + \nu_k$$

The states x_1 and x_2 denote altitude (Kft) and velocity (Kft/sec), respectively. The sampling interval was chosen as $T = 0.125$ sec and variance of the measurement noise is $V = 0.01(Kft)^2$. Note that the model is inputless and contains no process noise. Five experiments with simulated data were used in performing the training of the dynamic and static equations. These experiments used initial conditions randomly obtained from the intervals $x_{10} \in (297, 303)$, and $x_{20} \in (14, 26)$. A value of five was used for the constant appearing in the Hardy multiquadrics. The estimated values of the constants a, d, e_f, e_h were 2.006, 7.54, 3.31, and 3.21, respectively. After training, the RBFNN states were estimated using the initial conditions and initial covariance matrix

$$\rho_0 = \hat{P}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}$$

Fig. 3 portrays the estimates of x_1 and x_2 for a single run. It demonstrates the accuracy of the state estimation scheme with the RBFNN. It is to be noted that although stability of the matrix equation for the upper bound covariance matrix in (26) cannot be ensured, the estimator gain matrix was observed to converge to a constant matrix in just a few iterations. To prevent the covariance matrix from becoming unbounded, suitable upper bounds were placed on its elements. Fig. 4 shows the results of state estimation when the initial conditions of the estimator ($\hat{x}_0$) were chosen as (0,0), while the true initial conditions of the system (x_0) remained unchanged at the values given above.

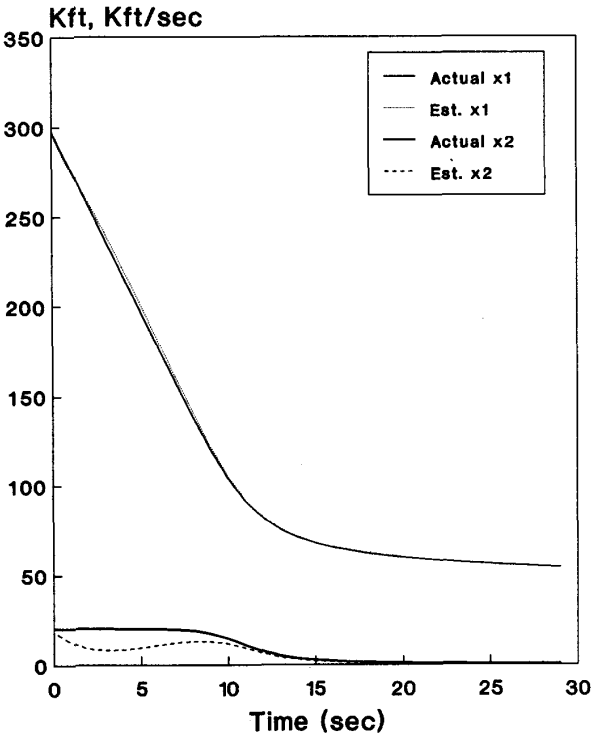


Fig. 3. Estimation with initial condition (300,20).

Next, a Monte Carlo analysis using 50 runs [each with different initial conditions uniformly distributed in the range (297–303, 14–26)] was performed. For each case the error vector $(x_k - \hat{x}_k)$ was generated at all sampling times, and the absolute values of the average errors at these times were obtained. Fig. 5 shows a plot of the absolute average error in estimating both the state variables. The plot shows similar but improved results over those obtained by using a second order filter from a known model [22]. It is worth pointing out that addition of pseudonoise w_k in estimation would help in further tuning the state estimator.

For the comparison of the proposed method with an existing scheme, the extended Kalman filter technique was used for state estimation with the simulated data. Without the knowledge on the exact model, the filter diverges as shown in Fig. 6. The estimation was limited to the first five seconds due to the rapid divergence of the estimated state variable.

Example 2: The second example chosen is that of a non-isothermal, non-adiabatic stirred reaction with an irreversible

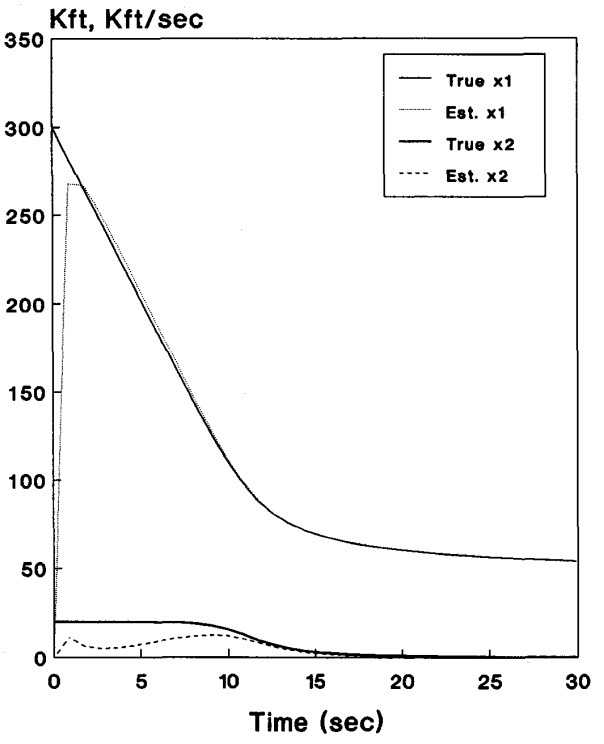


Fig. 4. Estimation with initial condition (0,0).

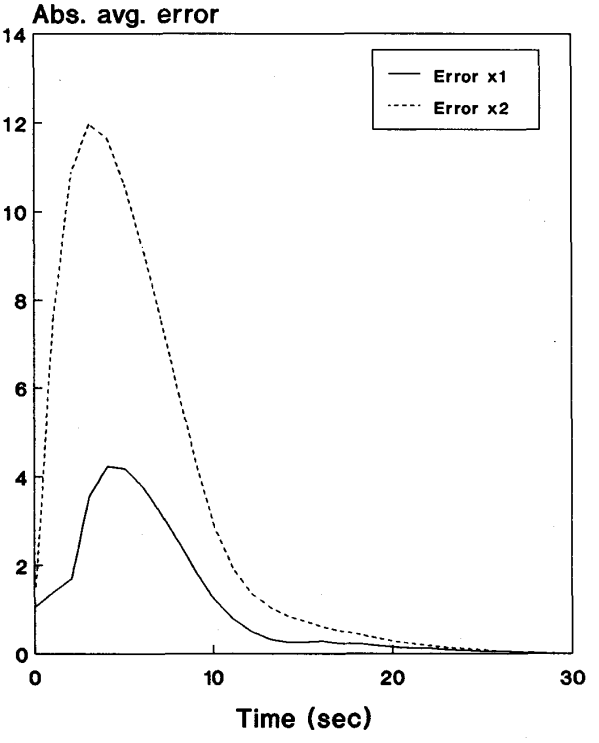


Fig. 5. Absolute average estimation errors.

first order reaction [23]. In this case, the nature of experiments that can be performed on the system is assumed to be of a

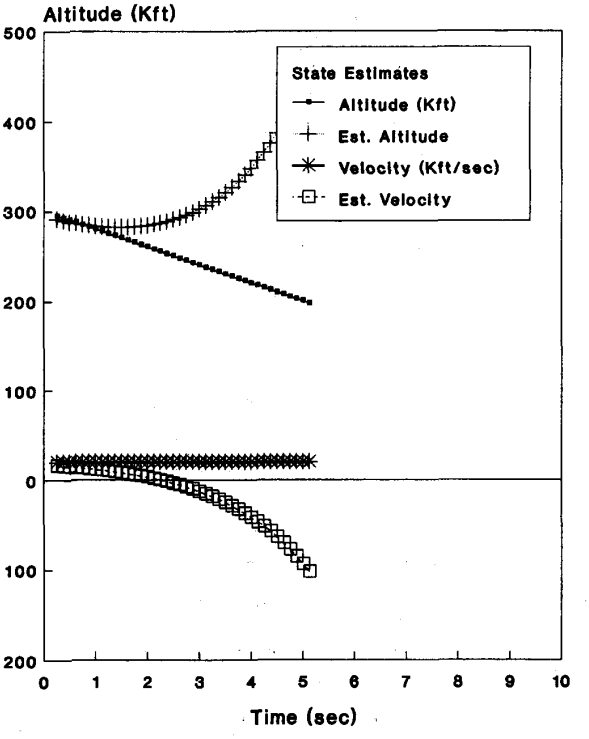


Fig. 6. State estimation by an extended Kalman filter.

restricted type. The second order system is given by:

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} + T\{-x_{1k} + 0.05(1 - x_{1k})e^{x_{2k}}\} \\ x_{2k} + T\{-x_{2k} + 0.4(1 - x_{1k})e^{x_{2k}} - ux_{2k}\} \end{bmatrix} \tag{33}$$

with an assumed measurement equation given by $y_k = [0\ 1]x_k + \nu_k$, such that variance of ν_k, $V = 0.001$. Here, x_1 is the conversion, and x_2 is the reactor temperature. The sampling interval was chosen to be 0.005 sec. It was assumed that only constant inputs could be used in identifying the physical process. In this example, five experiments were conducted with identical initial conditions of $x_0 = (1, 1)$, but with inputs given by $u_k = u_0 + \beta_k$. Nominal inputs of 1.0, 1.5, 2.0, 2.5, and 3.0 were used for u_0, where β_k is white Gaussian noise of variance 0.001. The inputs were chosen so as to span the region (1.0,3.0). In this example, the RBFNN training was performed to approximate the function over a region in (x_1, x_2, u) space. The value of the multiquadric constant c was chosen as 0.005. During state estimation, the output equation was assumed to be known and hence (30) and (31) were used. The constants a and e_f were estimated as 2.766 and 0.8967, respectively. Fig. 7 depicts the results of the state estimation when an input sequence of the form $u_k = 2.25 + \beta_k$ was used. Initial conditions of (1,1) and $\rho_0 = 0.1I$ were used. The results show that state estimation is accurate even for input sequences not used during the training stage.

Example 3 The final example is that of an inverted pendulum driven by an armature controlled DC motor governed

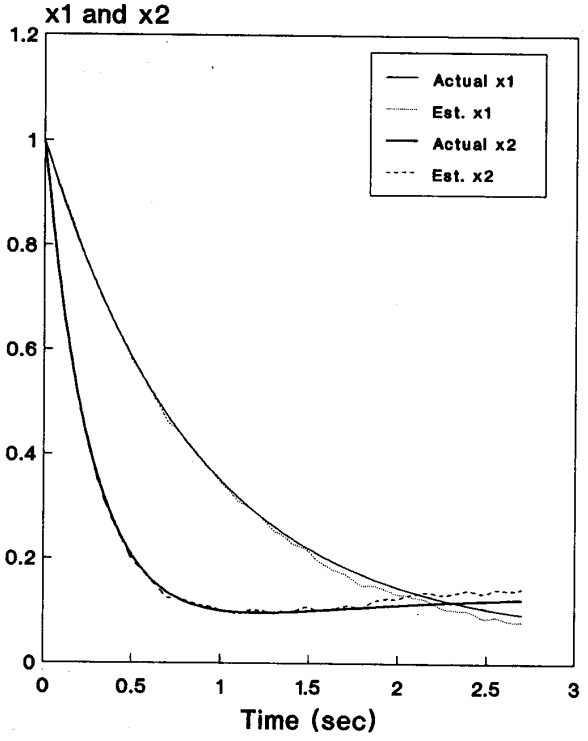


Fig. 7. State estimates x_1 and x_2.

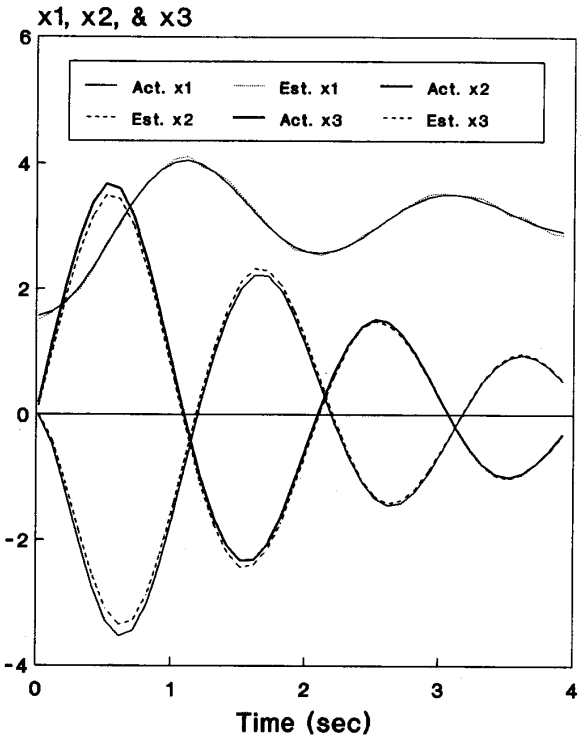


Fig. 8. Estimation with initial condition (1.57,0,0).

by the equations (Walcott *et al.* [24]):

$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \\ x_{3,k+1} \end{bmatrix} = \begin{bmatrix} x_{1k} + Tx_{2k} \\ x_{2k} + T(9.8 \sin x_{1k} + x_{3k}) \\ x_{3k} - 10T(x_{2k} + x_{3k}) \end{bmatrix} \tag{34}$$

where $T = 0.01$ sec, and output $y_k = x_{1k} + \nu_k$. The state variables x_1, x_2, and x_3 are in this case angular position, velocity, and motor current, respectively. Strength of the measurement noise was assumed to be $V = 0.001$. To train the RBFNN, five experiments were used with initial conditions such that $x_{10} \in ((\pi - 1)/2, (\pi + 1)/2)$, and $x_{20}, x_{30} = 0$. During state estimation, numeric values of $a = 38.6, e_f = 0.3106, c = 0.01$ were used. Fig. 8 shows the results of state estimation using the following initial values.

$$x_0 = \hat{x}_0 = \begin{pmatrix} \frac{\pi}{2} \\ 0 \\ 0 \end{pmatrix},$$

$$\rho_0 = \hat{P}_0 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$

Estimation of the states is seen to be highly accurate for all three states. As a second variation, the initial conditions of the estimator states were assumed to be (0,0,0), while the system remained unchanged with the same initial conditions as above. Fig. 9 shows that the state estimation performs well even under these conditions. The initial covariance matrix for this case was the same as above, but with $\hat{P}_0(1,1) = 1$.

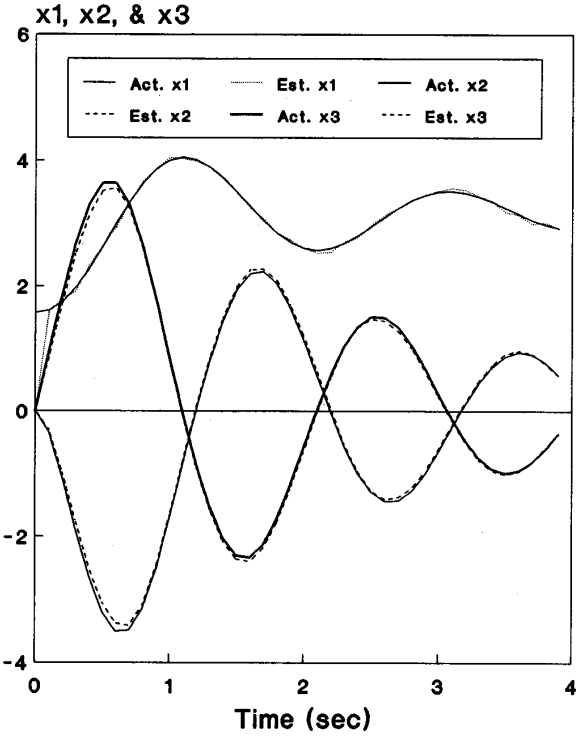


Fig. 9. Estimation with initial condition (0,0,0).

VI. Conclusions

This paper presented an RBFNN approach to approximately represent general nonlinear stochastic systems in prespecified domains of the state and input space. Multiple experiments are used to train the network. A parameter identification approach to training the network has been presented. Since the parameters appear linearly in the RBFNN, least squares estimation is possible. The trained network can then be used to predict states for different input sequences and initial conditions than those used during training.

A state estimator has been designed for use with the RBFNN. The gain matrix has been derived on the basis of an upper bound covariance matrix. Explicit inclusion of the approximation error into the estimation algorithm helps in minimizing filter divergence. Three applications of the method to estimate the states of nonlinear systems have been presented. The accuracy of state estimation using the RBFNN has been shown to be very good even when there are uncertainties in the knowledge of initial conditions.

Appendix

A. Derivation of the Upper Bound Covariance Matrix

Consider the state estimator applied to the approximated system,

$$\hat{x}(k+1) = f'(\hat{x_k}, u_k) + F\hat{x_k} + bu_k + K_k[y_k - h'(\hat{x_k}) - H\hat{x_k}] \tag{35}$$

Recall that the approximation error in approximating the system was specified by $\tilde{f} \triangleq f(x_k, u_k) - f'(x_k, u_k) - Fx_k - bu_k$ and $\tilde{h} \triangleq h(x_k) - h'(x_k) - Hx_k$. Furthermore, let us define conditions on the predetermined approximation errors and the Lipschitzean continuity of the approximated functions.

$$\|\tilde{f}\|_\infty \le e_f; \|\tilde{h}\|_\infty \le e_h$$
$$\|f'(x_k, u_k) - f'(\hat{x_k}, u_k)\|_\infty \triangleq \|\bar{f}\|_\infty \le a\|\tilde{x}_k\|_\infty$$
$$\|h'(x_k) - h'(\hat{x}_k)\|_\infty \triangleq \|\bar{h}\|_\infty \le d\|\tilde{x}_k\|_\infty$$

Then,

$$\begin{aligned}\tilde{x}_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\ &= \tilde{f} + F\tilde{x}_k + \bar{f} + w_k - K_k[\tilde{h} + H\tilde{x}_k + \bar{h} + v_k] \\ &= \tilde{f} + \bar{f} - K_k\bar{h} - K_k\tilde{h} + (F - K_kH)\tilde{x}_k + w_k - K_kv_k\end{aligned}$$

The true covariance matrix is given by

$$\begin{aligned}&E[\tilde{x}_{k+1}\tilde{x}_{k+1}^T] \triangleq \rho_{k+1} \\ &= E[\tilde{f}\tilde{f}^T + K_k\tilde{h}\tilde{h}^TK_k^T] + E[\tilde{f}\bar{f}^T + \bar{f}\tilde{f}^T] \\ &\quad - E[\tilde{f}\tilde{h}^TK_k^T + K_k\tilde{h}\tilde{f}^T] - E[\tilde{f}\bar{h}^TK_k^T + K_k\bar{h}\tilde{f}^T] \\ &\quad + E[\tilde{f}\tilde{x}_k^T(F - K_kH)^T + (F - K_kH)\tilde{x}_k\tilde{f}^T] \\ &\quad + E[\bar{f}\bar{f}^T + K_k\bar{h}\bar{h}^TK_k^T] - E[K_k\tilde{h}\bar{f}^T + \bar{f}\tilde{h}^TK_k^T] \\ &\quad + E[K_k\tilde{h}\bar{h}^TK_k^T + K_k\bar{h}\tilde{h}^TK_k^T] - E[\bar{f}\bar{h}^TK_k^T + K_k\bar{h}\bar{f}^T] \\ &\quad + E[\bar{f}\tilde{x}_k^T(F - K_kH)^T + (F - K_kH)\tilde{x}_k\bar{f}^T] \\ &\quad - E[K_k\tilde{h}\tilde{x}_k^T(F - K_kH)^T + (F - K_kH)\tilde{x}_k\tilde{h}^TK_k^T] \\ &\quad - E[K_k\bar{h}\tilde{x}_k^T(F - K_kH)^T + (F - K_kH)\tilde{x}_k\bar{h}^TK_k^T] \\ &\quad + (F - K_kH)\rho_k(F - K_kH)^T + W + K_kVK_k^T\end{aligned}$$

Now, let us construct an upper bound to the true covariance matrix on a term by term basis [8]. In order to do so we will make use of the matrix inequalities presented in Theorem 2. The symbol $Tr(.)$ is used to denote the trace of a matrix.

Theorem 2: Let S and T be square matrices of appropriate dimensions. Define n-vectors e_1, e_3, and m-vectors e_2, e_4 such that

$$\|e_1\|_\infty \le a'; \ \|e_2\|_\infty \le b'; \ \|e_3\|_\infty \le c'\|\tilde{x}_k\|_\infty;$$
$$\|e_4\|_\infty \le d'\|\tilde{x}_k\|_\infty;$$

then the following inequalities hold true:

1) $E[e_1e_1^T] \le na'^2I; E[e_2e_2^T] \le mb'^2I$
2) $E[e_1e_3^T + e_3e_1^T] \le a'c'(n + Tr(\rho_k))I$
3) $E[Se_2e_1^T + e_1e_2^TS^T] \le a'b'(nI + mSS^T)$
4) $E[e_1\tilde{x}_k^TS^T + S\tilde{x}_ke_1^T] \le a'(nI + S\rho_kS^T)$
5) $E[Se_1e_3^T + e_3e_1^TS^T] \le c'a'[Tr(\rho_k)I + nSS^T]$
6) $E[Se_1\tilde{x}_k^TT^T + T\tilde{x}_ke_1^TS^T] \le a'(T\rho_kT^T + nSS^T)$
7) $E[e_3e_3^T + Se_4e_4^TS^T] \le c'^2Tr(\rho_k)I + d'^2Tr(\rho_k)SS^T$
8) $E[e_3e_4^TS^T + Se_4e_3^T] \le c'd'Tr(\rho_k)(I + SS^T)$
9) $E[e_3\tilde{x}_k^TS^T + S\tilde{x}_ke_3^T] \le S\rho_kS^T + c'Tr(\rho_k)I$
10) $E[Se_3\tilde{x}_k^TT^T + T\tilde{x}_ke_3^TS^T] \le c'[T\rho_kT^T + Tr(\rho_k)SS^T]$

Proof: The theorem can be proved on the basis of Hirsch's theorem [25] and elementary matrix manipulations. Hirsch's theorem states that for any square complex matrix $A = (a_{ij})$ of dimension n, the eigenvalues are such that $|\lambda_k| \le n \ Max_{ij}|a_{ij}| \ \forall k = 1, \cdots, n$. Since $E[e_1e_1^T] \le \lambda_{\max}\{E[e_1e_1^T]\}I$, and Hirsch's theorem implies that $\lambda_{\max}\{E[e_1e_1^T]\} \le na'^2$, from which (1) follows $E[e_1e_1^T] \le na'^2I$.

It can be seen also that $E[e_3e_3^T] \le E[e_3^Te_3I] \le c'^2\|\tilde{x}\|_\infty^2I = c'^2Tr(\rho_k)I$. Assertions (2) through (10) can be proved by simple matrix manipulations. As an example, let us consider the proof of (6). Since, $(Se_1 - a'T\tilde{x}_k)(Se_1 - a'T\tilde{x}_k)^T \ge 0$, it follows that

$$a'E[(Se_1\tilde{x}_k^TT^T + T\tilde{x}_ke_1^TS^T)] \le a'^2(nSS^T + T\rho_kT^T)$$

Therefore, $E[Se_1\tilde{x}_k^TT^T + T\tilde{x}_ke_1^TS^T] \le a'(T\rho_kT^T + nSS^T)$.

Based on the above theorem, the upper bound covariance matrix can be given by the following recursive inequality.

$$\begin{aligned}\rho_{k+1} \le\ & l_1(F - K_kH)\rho_k(F - K_kH)^T + l_2I + l_3Tr(\rho_k)I \\ & + l_4K_kK_k^T + l_5Tr(\rho_k)K_kK_k^T + W + K_kVK_k\end{aligned} \tag{36}$$

where

$$\begin{aligned}l_1 &= 1 + a + d + 2e_f + e_h \\ l_2 &= ne_f(2 + a + d + e_f + e_h) \\ l_3 &= a(a + d + e_f + e_h) \\ l_4 &= me_h(1 + a + d + e_f + e_h) \\ l_5 &= (a + d + ad + d^2 + e_fd + e_hd).\end{aligned}$$

Thus, the upper bound covariance matrix can be written as

$$\begin{aligned}\hat{P}_{k+1} =\ & l_1(F - K_kH)\hat{P}_k(F - K_kH)^T + l_2I + l_3Tr(\hat{P}_k)I \\ & + l_4K_kK_k^T + l_5Tr(\hat{P}_k)K_kK_k^T + W + K_kVK_k^T \\ \hat{P}_0 =\ & \rho_0\end{aligned}$$

REFERENCES

[1] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification.* Cambridge, MA: MIT Press, 1983.

[2] A. A. Georgiev, "Nonparametric system identification by kernel methods," *IEEE Trans. Automat. Contr.*, vol. 29, pp. 356–358, 1984.

[3] K. S. Narendra and K. Parthasarthy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, 1990.

[4] E. A. Misawa and J. K. Hedrick, "Nonlinear observers: A state-of-the-art survey," *ASME DSC*, vol. 10, pp. 59–69, 1988.

[5] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* New York: Academic Press, 1970.

[6] R. J. Fitzgerald, "Divergence of the Kalman filter," *IEEE Trans. Automat. Contr.*, vol. 16, pp. 736–747, 1971.

[7] A. K. Mahalanabis and M. Farooq, "A second-order method for state estimation of non-linear dynamical systems," *Int. J. Contr.*, vol. 14, pp. 631–639, 1971.

[8] P. P. Gusak, "Upper bound for the RMS filtering performance criterion in quasilinear models with incomplete information," *Autom. and Remote Contr.*, vol. 42, pp. 70–76, 1981.

[9] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science*, vol. 24, pp. 978–982, 1990.

[10] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Sys*, vol. 2, pp. 321–355, 1988.

[11] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[12] I. R. H. Jackson, "Convergence properties of radial basis functions," *Constr. Approx.*, vol. 4, pp. 243–264, 1988.

[13] J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory of Functions of Several Variables* (Lecture Notes in Math 571). New York: Springer, 1977, pp. 85–100.

[14] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *J. Geophys. Res.*, vol. 76, pp. 1905–1915, 1971.

[15] M. J. D. Powell, *Approximation Theory and Methods.* Cambridge: Cambridge University Press, 1981.

[16] I. P. Schagen, "Sequential exploration of unknown multi-dimensional functions as an aid to optimization," *IMA J. Numerical Anal.*, vol. 4, pp. 337–347, 1984.

[17] R. Franke, "Scattered data interpolation: Tests of some methods," *Math Comput.*, vol. 38, pp. 181–200, 1982.

[18] C. A. Micchelli, "Interpolation of scattered data: Distance matrices and conditionally positive definite functions," *Constr. Approx.*, vol. 2, pp. 11–22, 1986.

[19] M. J. D. Powell, "Radial basis functions for multivariable approximations: A review," in *Proc. IMA Conf. on Algorithms for the Approx. of Functions and Data.* Oxford: Oxford University Press, pp. 143–167, 1985.

[20] C. H. Knapp and P. K. Pal, "Parameter identification in a class of nonlinear systems," *IEEE Trans. Automat. Contr.*, vol. 28, pp. 497–503, 1983.

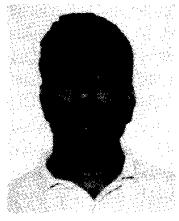
[21] M. Aoki, *Optimization of Stochastic Systems: Topics in Discrete-Time Dynamics.* San Diego, CA: Academic Press, 1989.

[22] P. S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 2. New York: Academic Press, 1982.

[23] A. A. Patwardhan, J. B. Rawlings, and T. F. Edgar, "Model predictive control of nonlinear processes in the presence of constraints," in *IFAC Nonlinear Contr. Sys. Des. Symp.*, Capri, Italy, 1989, pp. 345–349.

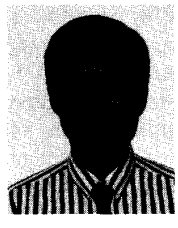
[24] B. L. Walcott, M. J. Corless, and S. H. Zak, "Comparative study of non-linear state-observation techniques," *Int. J. Contr.*, vol. 45, no. 6, pp. 2109–2132, 1987.

[25] M. Marcus and H. Ming, *A Survey of Matrix Theory and Matrix Inequalities.* Boston, MA: Allyn and Bacon, 1964.



Sunil Elanayar V. T. received the B.Tech. degree in 1986 and the M.S. degree in 1988 from the Indian Institute of Technology, Delhi, India, and the University of Alabama, Tuscaloosa, U.S.A., respectively. From 1988 to 1990 he was a graduate student at the Pennsylvania State University, University Park. He received the Ph.D. degree from Purdue University, West Lafayette, IN, U.S.A. in 1993 and is currently a postdoctoral student in the school of Mechanical Engineering at Purdue University.

His research interests include neural networks and applications of AI in process monitoring. He is a member of the SME (Society of Manufacturing Engineers) and Alpha Pi Mu.



Yung C. Shin received the B.S. degree from the Seoul National University in 1976, the M.S. degree from the Korean Advanced Institute of Science in 1978, and the Ph.D. degree from the University of Wisconsin in Madison in 1984, all in mechanical engineering.

From 1984 to 1988, he was employed at the General Motors Technical Center in Warren, Michigan as a senior project engineer. Between 1988 and 1990 he was an assistant professor at Pennsylvania State University. In 1990 he joined the faculty of Purdue University in mechanical engineering, where he is currently an associate professor. His current research interests include intelligent control, dynamic system modeling and identification, in-process sensing, neural networks, and adaptive control with manufacturing processes and systems.

He has organized and coorganized the Artificial Neural Networks in Engineering Conference twice, in 1991 and 1992, a symposium on Neural Networks in Manufacturing and Robotics in 1992, and a symposium on Intelligent Design and Manufacturing in 1993. He also was a coeditor of a special issue on neural networks for the *Journal of Intelligent Manufacturing*, and *Intelligent Engineering Systems through Artificial Neural Networks*, volumes 1 and 2.