**Proceedings of the 17th World Congress**
**The International Federation of Automatic Control**
**Seoul, Korea, July 6-11, 2008**

# A Neuro-based EM-Particle Smoothing Algorithm For Identification of Nonlinear State Space Models

**Ali A.Gorji** * **Mohammad B.Menhaj** **

* *Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran, (e-mail:gorji1983@aut.ac.ir)*
** *Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran, (e-mail:tmenhaj@ieee.org)*

**Abstract:** The expectation maximization(EM) algorithm and particle filtering have been greatly used in many estimation problems. In this paper, we propose a combination of the EM algorithm and particle smoothing for identification of nonlinear state space models using artificial neural networks. After representing a radial basis function(RBF) neural network as a parametric structure for describing the state transition and output equations of a state space model, the EM algorithm is applied for updating parameters and estimating states of the nonlinear system. Moreover, the particle smoothing algorithm is used at the E phase for state estimation. Simulation studies show the fast convergence rate and satisfactory accuracy of the proposed method in identification of nonlinear plants whose state transition function, output structure or both are unknown.

Keywords: Nonlinear State Space Models, radial basis functions, the EM algorithm, particle smoothing, joint parameter and state estimation, radial basis function networks

## 1. INTRODUCTION

The expectation maximization(EM) algorithm initially proposed by [1], [2] has been greatly used in many parameter and state estimation problems [3], [4]. Indeed, the considerable flexibility of the EM algorithm in joint parameter and state estimation has made them popular to many nonlinear system identification problems.
The combination of the EM algorithm and various state estimation methods for identification of nonlinear systems has been the topic of the literature [7]. However, the significant weakness of the literature is an assumption on the overall structure of the nonlinear system supposed to be completely or, in some cases, partially known. In fact, no discussion has been made on generalization of the EM algorithm to a more complex case in which the overall system is completely unknown. This problem can arise in many real applications such as blind signal processing whose best example is blind equalization of nonlinear channels [8] or many common models whose dynamic is unknown [4].
Using parametric structures for modeling nonlinear systems when the system is completely unknown was first proposed by [9]. They proposed using a linear dynamical system instead of the state transition and output models and, then, applied the EM algorithm to the joint parameter and state estimation of the proposed structure. Next, they extended their method to a nonlinear case when a radial basis function(RBF) network was used as a parametric structure [4]. The aforementioned strategy was also successfully applied to identification of nonlinear systems and time series forecasting in the presence of missing data. The most significant flaw of the mentioned work was using the extended Kalman filter/smoother(EKF/EKS)[10] method at the state estimation phase leading to poor results in the presence of severe nonlinearities. Moreover, deriving learning rules when the EKS is used in conjunction with the EM algorithm is very hard and, therefore, applying new learning rules, specifically adjusting the parameters of the basis functions, may be impossible.
Recently, the sequential Monte Carlo framework known as particle filtering has been greatly used in many estimation, tracking and control issues [11], [12], [13]. Besides the much more superior quality of this method in dealing with nonlinear systems compared with the traditional Kalman based approaches, the particle filter/smoother can be much more easily joint with the EM algorithm than the EKS method. The latter advantage has caused that the combination of the particle smoothing algorithm and the EM to be applied to many state and parameter estimation issues [7], [14] although applying the proposed method to blind case where a parametric structure has been used to model the state transition or output structure is yet an open problem.
In this paper, we show how to apply the particle smoothing and EM algorithm, named as the smoothed EM-particle, to identification of general nonlinear state space models (NSSM) described by an intelligent structure where we consider the class of discrete time state space models. After a brief review of NSSMs and parameter and state estimation problems in section 2, the EM algorithm is presented in section 3. In this section, a very brief discussion is made on particle filtering/smoothing and its application in the state estimation phase of the EM algorithm. Section 4

deals with our proposed method for adjusting the parameters of the RBF network used as the parametric structure in the paper. Simulation studies justifying the superiority of the algorithm are presented in section 5. Finally, section 6 concludes the paper.

## 2. NONLINEAR STATE SPACE MODELS

A general representation of an NSSM can be written as:

$$\begin{aligned} \mathbf{x}_t &= f_{\theta^x}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{v}_t \\ \mathbf{y}_t &= g_{\theta^y}(\mathbf{x}_t) + \mathbf{w}_t \end{aligned} \tag{1}$$

where $\mathbf{x}_t$ is the state vector at time t, $\mathbf{y}_t$ are the output observations, $\mathbf{u}_t$ is the external input vector, $\mathbf{v}_t$ and $\mathbf{w}_t$ are the white noises with zero mean and covariance matrixes Q and R, respectively. Furthermore, f and g are the nonlinear parametric equations with parameters $\theta^x$ and $\theta^y$, respectively, meeting the local Lipschitz condition [15]. In addition, without any loss of generality, we have eliminated the external input $\mathbf{u}_t$ from the output function. This issue is usually justified in control applications where the external input is considered in the state transition model. However, in many other applications, inputs appear in the output model. For example, in the observer trajectory planning topic [16], the state transition model is independent from external inputs. The identification stages can be summarized as the following ones:

(1) **State Estimation**
Because the state vector is represented by a stochastic variable, the final aim of the state estimation is estimating the posterior density function of states given outputs and parameters, $p(\mathbf{x}_t|\mathbf{d}_{1:t}, \theta_t)$, where $\mathbf{d}_{1:t} = \{\mathbf{y}_{1:t}, \mathbf{u}_{1:t}\}$.

(2) **Parameter Estimation**
Generally, the parameter estimation stage can be defined as either estimating the posterior density function of parameters given states and measurements when dual filters are used or optimizing a cost function such as the log likelihood function. To tackle the dual filtering method, the following simple time evolution model is usually considered for parameters of the system:

$$\theta_t = \theta_{t-1} + \mathbf{n}_t \tag{2}$$

Here $\mathbf{n}_t$ is an artificial gaussian noise with zero mean and an arbitrary covariance matrix. The augmented state vector can be constructed as $X = \{\mathbf{x}_t, \theta_t\}$. There are various ideas for estimating the above augmented state vector such as using the dual Kalman or particle filtering [7] and Rao-Blackwellized particle filtering [17]. However, some weaknesses of the aforementioned strategies such as sensitivity to the initial value of parameters or the computational cost [13] has persuaded researchers to prefer the point estimation approaches for estimating parameters of the model. The EM algorithm discussed later is a very popular type of point estimation approaches.

With regard to the above discussion, the objective of the system identification problem can be summarized as estimating the model states ($\mathbf{x}_t$) and a set of parameters $(\theta_x, \theta_y)$ given measurements $\{\mathbf{u}_{1:t}, \mathbf{y}_{1:t+1}\}$.

## 3. THE EM ALGORITHM FOR JOINT PARAMETER AND STATE ESTIMATION

The EM algorithm was first proposed by Dempster [1] as an extension of the work conducted by Baum and his colleagues [2]. The EM algorithm can be considered as a type of maximum likelihood (ML) estimator. In other words, the final goal of the EM algorithm is to maximize the log likelihood function, $ln(p(y|\theta))$. To maximize the log likelihood function, or equivalently minimize $-ln(p(y|\theta))$, the function should be integrated over estimated states as follows:

$$ln(p(y|\theta)) = ln(\int_x p(y, x|\theta)dx) \tag{3}$$

Now, consider an arbitrary distribution function, q(x), for states. By using the jensen's inequality principle [5] and the arbitrary distribution of states, the above equation can be written as follows:

$$\begin{aligned} ln(p(y|\theta)) &= \int ln\big(p(y, x|\theta)q(x)\big) \\ &\quad - \int ln(q(x))q(x) = F(q, \theta) \end{aligned} \tag{4}$$

where for the sake of brevity the phrase $dx$ has been eliminated. The maximization of the log likelihood function is equivalent to maximizing the lower bound, $F(q, \theta)$. The EM algorithm uses two separate stages to maximize the lower bound of the log likelihood function. These two steps can be summarized as follows:

- **E Step**
At the E step of the EM algorithm, the distribution function of states are updated to maximize $F(q, \theta)$. It is very straightforward to show that maximization of F is equivalent to estimate the posterior distribution of hidden states given measurements as follows [4]:
$$q^t = p(\mathbf{x}_t|\mathbf{y}_{1:t}, \theta_{t-1}) = argmax_q\big(F(q, \theta_{t-1})\big) \tag{5}$$
Indeed, the E step is not anything but a common state estimation framework.
- **M Step**
Similarly, the M step of the EM algorithm can be considered as the maximization of $F(q, \theta)$ with regard to parameters when states estimated at the E step are used.In other words, the M step is dealt with by maximizing the following function:
$$\theta_t = argmax_\theta < p(y|x, \theta) >_{q^t} \tag{6}$$
where $< p(y|x, \theta) >_{q^t}$ is the complete likelihood function whose expected value is computed over the distribution of hidden states.

By repeating the E and M step of the EM algorithm at consecutive time steps, one can easily show that the likelihood increases at each time step [5]. This important characteristic entails the stability of the EM algorithm.

### 3.1 E Step: State Estimation Algorithms

Although Kalman based smoothers such as Extended Kalman Smoother (EKS) and Unscented Kalman Smoother (UKS) [18] provide a simple framework to estimate hidden states in an off-line mode, the lack of suitable accuracy in dealing with nonlinear systems is taken into account as one of their important problems. Moreover, implementation of the M step of the EM algorithm, specifically computing the expected value of the likelihood function, may be very hard or even intractable when hidden states' density function is described by a Gaussian one. To remedy this problem, sampling approaches have been proposed in the literature among which the particle smoothing algorithm plays an important role in many estimation problems. The great power of the particle smoothing in modeling and estimation of the nonlinear state space systems and its flexibility in joining with the EM algorithm compared with the EKS or UKS method have increased its popularity. In the following, first, the particle filter algorithm is presented and, then, some recent ideas for implementation of the particle smoothing algorithm are introduced.

### 3.2 The Particle Filter for State Estimation

Consider the problem of online state estimation as computing the posterior probability density function $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. To provide a recursive formulation for computing the above density function, the following stages should be taken:

- Prediction stage: this step is proceeded as computing the following distribution functions:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int_{\mathbf{x}_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \quad (7)$$

- Update stage: this step can be described as follows:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \quad (8)$$

The particle filter estimates the probability distribution function $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ by sampling from a specific distribution function as follows:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \Sigma_{i=1}^{N} \tilde{w}_t^i \delta(\mathbf{x}_t - \mathbf{x}_t^i) \quad (9)$$

where i=1,2,...,N is the sample number, $\tilde{w}_t$ is the normalized importance weight and $\delta$ is the delta dirac function. In the above equation, the state vector $\mathbf{x}_t^i$ is sampled from the proposal density function $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t})$. By substituting the above equation in (7) and, then, simplifying (8) and the fact that states are drawn from the proposal function q, the recursive equation for the prediction and update steps can be written as follows:

$$w_t^i = w_{t-1}^i p(\mathbf{y}_t|\mathbf{x}_t^i) \frac{p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i, \mathbf{y}_{1:t})} \quad (10)$$

where $\mathbf{x}_t^i$ is the $i^{th}$ sample of $\mathbf{x}_t$. The main failure of the above procedure known as the sequential importance sampling (SIR) algorithm in the literature is the degeneracy problem. That is, after a few iterations one of the normalized importance ratios tends to 1, while the remaining ratios tend to zero. This problem causes the variance of the importance weights to increase stochastically over time [5]. To avoid the degeneracy of the SIS algorithm, a selection (resampling) stage may be used to eliminate samples with low importance weights and multiply samples with high importance weights. Indeed, the resampling stage conducts the following mapping:

$$\{\mathbf{x}_t^i, w_t^i\} \rightarrow \{\mathbf{x}_t^{m(i)}, \frac{1}{N}\} \quad (11)$$

In [5], a comprehensive discussion has been made on how to implement the residual resampling method known as the most efficient resampling approach in the literature. Using the residual resampling joint with the SIS algorithm can be presented as the sequential importance resampling (SIR) algorithm used for state estimation in many NSSMs.
**Remark 1:** The easiest choice for the proposal density function is the dynamical probability function as follows:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}^i, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}^i) \quad (12)$$

In [19], some other candidates for the proposal function have been presented but we use the simple dynamical one in this paper.

### 3.3 Particle Smoothing

The main goal of smoothing algorithms is estimating the smoothed distribution function $p(\mathbf{x}_t|\mathbf{y}_{1:T})$ where T is the number of available distributions. Many ideas have been presented in the literature for estimating the above-mentioned distribution function [20]. Three main algorithms are known as Forward-Backward Algorithm, Two filter smoothing method and maximum a posteriori approach. The first two methods are not discussed here. Indeed, the first method is very time consuming while its accuracy completely depends on the forward pass conducting a particle filter procedure. The second method cannot be applied to our framework because sampling from a proposal function at the backward stage is very hard or even impossible when a neural network structure is used as the output function [14]. The maximum a posteriori method is more reliable than the forward-backward approach. Moreover, it can be easily implemented for our application. In the following, a very brief review is made on the aforementioned algorithm and further details can be obtained from the given references.

- **Maximum a Posteriori Particle Smoothing:**
  In this approach, after implementing the forward pass and computing particles, smoothed states are estimated by maximizing the posterior density function as follows:

$$\hat{\mathbf{x}}_{1:T} = argmax_{\mathbf{x}_{1:T} \in \mathbf{x}_{k=1:T}^{i=1:N}} p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T}) \quad (13)$$

  The above maximization can be accomplished by using the Viterbi algorithm implemented on a set of discretized state space induced by the particle filter approximation. Indeed, the above algorithm consists of two separate stages. The first stage is a simple filtering method while the other one is an optimization conducted over a set of nominal trajectories which

are samples states. Therefore, the idea behind this approach is very easy and its implementation is also straightforward. More details of this algorithm can be found in [21].

## 4. M STEP: THE GRADIENT BASED ALGORITHM FOR PARAMETER ESTIMATION

Consider (1). In this section, we show how to adjust parameters of the dynamical part and a completely similar procedure can be also proceeded for the output equation. To adjust the parameters of the model recursively, the likelihood function should be first determined. To do so, the likelihood function can be written as follows:

$$p(\mathbf{x}_t|\theta^{\mathbf{x}}, Q) = \frac{1}{(2\pi)^{\frac{n_x}{2}}|Q_t|^{\frac{1}{2}}} \times exp[-\frac{1}{2}(\mathbf{x}_t - \tilde{\mathbf{x}}_t)^T Q_t^{-1}(\mathbf{x}_t - \tilde{\mathbf{x}}_t)] \quad (14)$$

where $n_x$ is the dimension of states. Also, $\mathbf{x}_t$ is the state vector estimated at the E step of the EM algorithm. However, $\tilde{\mathbf{x}}_t$ refers to the predicted values of the state vector computed with regard to states at the previous time. Given the above equation, the negative log likelihood function for the dynamical model is written as:

$$L(\theta^x, Q) = \frac{1}{2}(\mathbf{x}_t - \tilde{\mathbf{x}}_t)^T Q_t^{-1}(\mathbf{x}_t - \tilde{\mathbf{x}}_t) + \frac{1}{2}ln(|Q_t|) \quad (15)$$

The method used for maximizing/minimizing the aforementioned function depends on the parametric structure considered for the state transition or output model. The RBF network has been preferred to many other intelligent structures because of linear relationship between the network's parameters and output of the RBF network. This characteristic facilitates dealing with the maximization procedure by implementation of the least square method. To do so, the RBF network is first presented as follows:

$$\mathbf{x}_t = \Sigma_{k=1}^{k_x}\mathbf{h}_k\rho_k(\phi, \mathbf{x}_{t-1}) + A\mathbf{x}_{t-1} + B\mathbf{u}_t + \mathbf{b} + \mathbf{v}_t \quad (16)$$

where $k_x$ is the number of basis functions for the state transition model and $\rho(\phi)$ is a radial basis function with the parameter $\phi$. Moreover, the parameter of the model is defined as follows:

$$\theta^x = [\mathbf{h}_1\ \mathbf{h}_2\ \ldots\ \mathbf{h}_{k_x}\ A\ B\ \mathbf{b}] \quad (17)$$

Various types of functions have been proposed to represent the basis functions in (17) such as exponential, polynomial or linear structures [5]. Among them, exponential functions provide a powerful tool to tackle the nonlinearity embedded in many real systems. Briefly, an exponential basis function can be introduced as follows:

$$\rho(\phi) = exp[-\frac{1}{2}(\mathbf{x}_t - \mu)^T S^{-1}(\mathbf{x}_t - \mu)] \quad (18)$$

where $\mu$ and S represent the mean and covariance of each exponential basis function, respectively. Now, the overall form shown in (17) can be rewritten for the dynamical model as:

$$\mathbf{x}_t = \theta_{t-1}^x \Phi_{t-1}^x + \mathbf{v}_t \quad (19)$$

where $\Phi^x$ is known as the nonlinear regressor and determined as:

$$\Phi_{t-1}^x = \begin{bmatrix} exp(\mathbf{x}_{t-1}, \mu_x^1, S_x^1) \\ . \\ . \\ . \\ exp(\mathbf{x}_{t-1}, \mu_x^{k_x}, S_x^{k_x}) \\ \mathbf{x}_{t-1} \\ \mathbf{u}_t \\ 1 \end{bmatrix} \quad (20)$$

The M phase of the EM algorithm is equivalent to maximizing the expected value of the log-likelihood function over estimated states. In other words the cost function is defined as below:

$$J_x = \Sigma_{t=1}^T < L(\theta^x, Q) >_{p_t^x} \quad (21)$$

where $< L(\theta^x, Q) >_{p_t^x}$ is computed by using the following equation:

$$< L(\theta^x, Q) >_{p_t^x} = \int\int L(\theta^x, Q)p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:T})d\mathbf{x}_{t-1}d\mathbf{x}_t \quad (22)$$

Taking derivative from (21) and equating zero leads to the following learning rules for updating the parameters of the model and covariance matrixes:

$$\theta_{m+1}^x = \Big(\Sigma_{t=1}^T < \mathbf{x}_t(\phi_{t-1}^x)^T >_{p_t^x}\Big) \Big(\Sigma_{t=1}^T < (\phi_{t-1}^x)^T(\phi_{t-1}^x)^T >_{p_t^x}\Big)^{-1}$$
$$Q_{m+1} = \frac{1}{T}\Big(\Sigma_{t=1}^T < \mathbf{x}_t\mathbf{x}_t^T >_{p_t^x} - \theta_{m+1}^x\Sigma_{t=1}^T < \phi_{t-1}^x\mathbf{x}_t^T >_{p_t^x}\Big) \quad (23)$$

Similarly, a learning rule is derived for parameters of the output function only by substituting $\mathbf{x}_t$ for $\mathbf{y}_t$ and defining $J_y$ like $J_x$ shown by (21).
Now, the expected values in above equations should be computed. To do so, the joint probability density function of smoothed states can be written as:

$$p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:T}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{1:t})p(\mathbf{x}_t|\mathbf{y}_{1:T}) \quad (24)$$

By some algebraic operations, the above function can be simply written as follows:

$$p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:T}) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})p(\mathbf{x}_t|\mathbf{y}_{1:T})}{\int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1}} \quad (25)$$

By using the MAP smoothing method presented in the last section, the above equation can be approximated by the following expression:

$$p(\mathbf{x}_t, \mathbf{x}_{t-1}|\mathbf{y}_{1:T}) \sim$$
$$\Sigma_{i=1}^{N}\Big(\frac{w_{t-1}^{i}p(\mathbf{x}_t^M|\mathbf{x}_{t-1}^{i})\delta(\mathbf{x}_{t-1}-\mathbf{x}_{t-1}^{i})\delta(\mathbf{x}_t-\mathbf{x}_t^M)}{\Sigma_{l=1}^{N}w_{t-1}^{l}p(\mathbf{x}_t^M|\mathbf{x}_{t-1}^{l})}\Big) \quad (26)$$

where $w_{t-1}^{i}$ denotes the $i^{th}$ normalized importance weight computed at the forward pass (filtering) and $\mathbf{x}_t^M$ is the smoothed state estimated by using the MAP particle smoothing approach. Now, expected values of (23) can be easily approximated only by substituting (26) for $p_t^x$ in (23) and converting nonlinear integrals to a set of simple summations. Note that for computing expressions dependent on $\mathbf{x}_t$ the MAP smoother gives us a unique approximation $\mathbf{x}_t^M$ and, therefore, $\mathbf{x}_t$ can be easily approximated by the mentioned expression. Therefore, learning rules of (23) can be easily implemented with regard to aforementioned approximations.

## 5. SIMULATION RESULTS

Assume the following nonlinear state space model:

$$x_t = .5x_{t-1} + \frac{25x_{t-1}}{1+x_{t-1}^2} + 8cos(1.2t) + v_t$$
$$y_t = .05x_t^3 + w_t \quad (27)$$

Here $v_t$ and $w_t$ are white noises with zero means and arbitrary covariance matrixes. Assume that the dynamic of the above model is unknown. To model the dynamic of the presented structure, an RBF network with 10 basis functions is used. The mean and covariance matrix of basis functions are chosen to cover the support of states. Training and test stages of the network are presented in the following subsections.

### *5.1 Training the Network*

First, synthetic data are generated according to the proposed model. A sequence of states and observations is generated with regard to the above-represented nonlinear model. To train the network, the EM algorithm joint with the MAP smoother presented in sections 3 and 4 are used. To show the effect of the sample size, the algorithm is implemented by various number of particles. To show the power of the algorithm in estimation of the parameters of the RBF network, Fig. 1 presents estimated values for one of the parameters of the RBF network. It is obvious that by increasing the number of particles the RBF network has been able to learn the dynamic of the model much better.

### *5.2 Testing the Network*

To evaluate the performance of the trained network, another data set is generated. To consider the effect of the initial value of states, we chose a random initial value with a normal distribution by zero mean and covariance 1. Now, we want to proceed the state estimation phase. To do so, the trained RBF network is substituted with the actual dynamical model. We use 500 particles for implementation of the estimation phase. Fig. 2 shows the result of state estimation using the generic particle filter where the RBF network trained in the last section by 1000 particles has been used. Moreover, Fig. 3 presents the estimation error

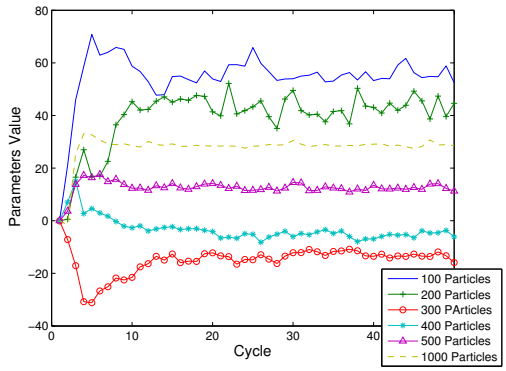


Fig. 1. Parameter Estimation Results for Different Number of Particles

with regard to various number of particles where the error is computed as follows:

$$e = \frac{1}{N}\Sigma_{t=1}^{N}\Big(x_t - \hat{x}_t\Big)^2 \quad (28)$$

Here $\hat{x}_t$ denotes the estimated state and $x_t$ is the actual one. Simulation results show that by increasing the number of particles the generalization ability of the network grows. Finally, the robustness of our trained network is tested by

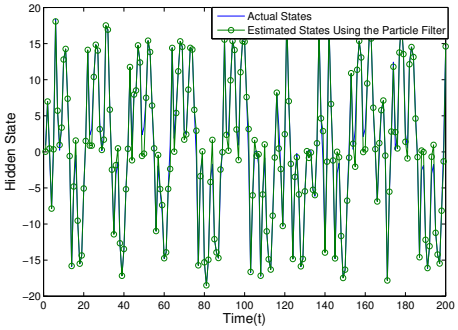


Fig. 2. Estimated States Using the Particle Filter and the Network Trained by 1000 Particles

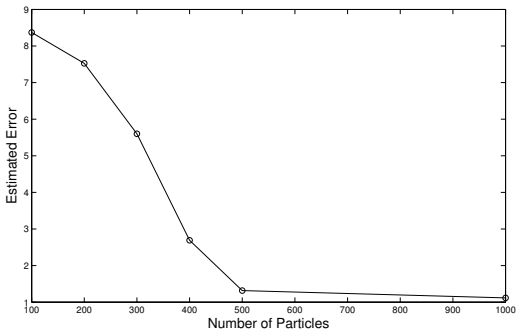


Fig. 3. Estimation Error for Networks Trained by Different Number of Particles

adding an artificial noise to coefficients of the actual model described by (27) where coefficients are defined as $\theta = [.5\ 25\ 8\ .05]$. After generating output observations, Fig. 4 shows the estimation error for the RBF network trained by 1000 particles with various values of the noise covariances. Simulation studies show that the trained network gives

satisfactory results for the variance lower than .02 but for other variances results are not reliable.

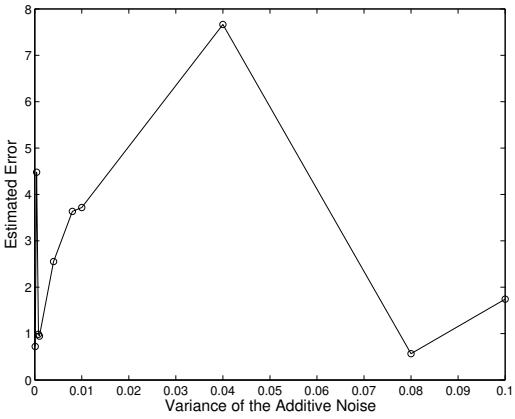


Fig. 4. Estimation Error for the Time Variant Model With Different Variances of the Additive Noise

## 6. CONCLUSION

In this paper, a new approach was presented for identification of nonlinear state space models using the RBF network. Because the identification issue can be formulated as a joint state and parameter estimation problem, the EM algorithm was discussed for estimating states and updating the parameters of the RBF network. Linear relationship between the output of the RBF network and its internal parameters caused the LS method to be chosen for parameter estimation. Furthermore, the MAP particle smoothing was used for state estimation at the E phase of the EM algorithm. Simulation studies showed the superiority of our proposed procedure for identification of nonlinear state space models when no information is available about the overall structure of the state transition or output model.

## REFERENCES

[1] L. E. Baum, J. E. Eagon, An Inequality with Applications to Statistical Estimation for Probalistic Functions of Markov Processes and to a Model for Ecology, Bulletin of American Mathematical Society, Vol. 73, pp. 360-363, 1967.

[2] A.P. Dempster, N. M. Laird, D. B. Rubin, Maximum Likelihood From Incomplete Data Via the EM Algorithm, J. Royal Statistical Society Series B, 39: 1-38, 1977.

[3] Thomas B. Schon, Adrian Wills, and Brett Ninness, Maximum Likelihood Nonlinear System Estimation, in Proceedings of the 14th IFAC Symposium on System Identification, Newcastle, Australia, March 2006.

[4] S. Roweis, Z. Ghahramani, Learning Nonlinear Dynamical Systems using the EM Algorithm, Simon Haykin, Ed. Kalman Filtering and Neural Networks, pp. 175-220, Wiley, 2001.

[5] Joao F. G. de Freitas, Bayesian Methods for Neural Networks, PhD thesis, Trinity College, University of Cambridge, 1999.

[6] Z. Ghahramani, Learning Dynamic Bayesian Networks. In Giles, C. L. and Gori, M., editors, Adaptive Processing of Temporal Information, volume 1387 of Lecture Notes in Artificial Intelligence, Springer-Verlag.

[7] Thomas B.Schon, Estimation of Nonlinear Dynamical Systems Theory and Applications, PhD thesis, Electrical Engineering Department, Linkopings University, 2006.

[8] Petar M.Djuric, Jayesh H.Kotecha, J. Zhang, Y. Huang, T. Ghirmai, Monica F.Bugallo, and J. Miguez, Particle Filtering and Application in Wireless Communications, IEEE Signal Processing Magazine, September 2003.

[9] Z. Ghahramani, and Geoffrey E.Hinton, Parameter Estimation for Linear Dynamical Systems, Tech. Rep, CRG-TR-96-2, Computer Science Department, University of Toronto, February 1996.

[10] R. E. Kalman, and R. S. Bucy, New Results in Linear Filtering and Prediction, Trans. American Society of Mechanical Engineers, Series D, Journal of Basic Engineering, Vol. 83D, pp. 95-108, 1961.

[11] A. Doucet, N. de Freitas, and N. Gordon, Eds., Sequential Monte Carlo Methods in Practice: Springer-Verlag, 2001.

[12] Christophe Andrieu, Arnaud Doucet, Sumeetpal S. Singth, and Vladislav B. Tadic, Particle Methods for Change Detection, System Identification, and Control, Proceedings of the IEEE, Vol. 92, No. 3, March 2004.

[13] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund, Particle Filters for Positioning, Navigation, and Tracking, IEEE Transactions on Signal Processing, Vol. 50, No. 2, February 2005.

[14] Mark Briers, Arnaud Doucet, and Simon Maskell, Smoothing Algorithms for State-Space Models, Technical Report CUED/F-INFENG/TR.498, Cambridge University, Engineering Department, 2004.

[15] Jean-Jacques Slotine, and Weiping Li, Applied Nonlinear Control, Prentice-Hall, 1991.

[16] Sumeetpal S.Singh, Nikolaos Kantas, Ba-Ngu Vo, Arnaud Doucet, and Robin J.Evans, Simulation-based Optimal Sensor Scheduling With Application to Observer Trajectory Planning, Automatica 43, 817-830, 2007.

[17] Thomas Schon, Fredrik Gustafsson, and Per-Johan Nordlund, Marginalized Particle Filters for Mixed Linear/Nonlinear State Space Models, IEEE Transactions on Signal Processing, Vol. 53, No. 7, July 2005.

[18] Simo Sarkka, Unscented Rauch-Tung-Striebel Smoother, Accepted for Publication in IEEE Transactions on Automatic Control, 2007.

[19] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra, Sequential Monte Carlo Samplers, J. R. Statist. Soc. B, 68, Part 3, pp. 411-436, 2006.

[20] Mike Klaas, Mark Briers, Nando de Freitas, Arnaud Doucet, Simon Maskell, and Dustin Lang, Fast Particle Smoothing: If I Had a Million Particles, Proceedings of the 23rd International Conference on Machine Learning, Pittsburg, PA, 2006.

[21] Simon J. Godsill, Arnaud Doucet, and M. West, Maximum a Posteriori Sequence Estimation Using Monte Carlo Particle Fillters, Ann. Inst. Stat. Math., 53, 82-96, 2001.