Nonlinear/non-Gaussian sequential estimation applied to neural networks: Algorithms

Branimir Todorović[1], Claudio Moraga[2], Miomir Stanković[1], Branko Kovačević[3]

Abstract – **Sequential Monte Carlo methods known as Particle Filters are applied in estimation of the recurrent radial basis function network parameters and hidden variables. The considered algorithms demonstrate smaller estimation error and improve robustness in convergence (regarding the initial parameter values), compared to the well known extended Kalman filter.**
Keywords - **Sequential Monte Carlo, neural networks, estimation, importance sampling, resampling**

I. INTRODUCTION

In many real world applications of neural networks (NN) data arrive sequentially in time, exhibiting non-linear, non-Gaussian and non-stationary behavior and the task is to estimate (train) neural network and inference sequentially as well. The Bayesian approach to estimation arises as natural solution to this problem. This approach is based on the state space formulation of the NN dynamics and sequential propagation of the NN parameter probability density function (pdf). Since the pdf embodies all available statistical information, it represents the complete solution of the estimation problem, from which the optimal (with respect to any criterion) estimate of the parameters, and measure of the estimation accuracy can be obtained. However, the approach offers only a conceptual solution since the analytic form of the required pdf in general cannot be obtained. Special cases include the well known Kalman filter, which assumes a linear model and Gaussian nature of the noise. In order to deal with the non-linearity and non-Gaussianity of the data, various approximations have been considered. The extended Kalman filter linearizes the model (neural network in our case) around the last estimate and makes a Gaussian approximation of the parameter pdf. Using these approximations the EKF have been applied to the feed forward [2,7] and recurrent NN training [8,10].

Estimators based on Sequential Monte Carlo (SMC) methods, known as a Particle Filters, attempt to overcome constraints of the linear Gaussian approximation. The central idea is to represent the required pdf by a set of random samples with associated weights, propagate them sequentially according to the assumed state space model, and calculate estimates based on these samples and weights. Some interesting strategies for training of feed forward NN using SMC methods are discussed in [3].

[1]Faculty of Occupational Safety, University of Niš, 18000 Niš, Yugoslavia, E-mail: bssmtod@EUnet.yu

[2]Department of Artificial Intelligence, Polytechnical University of Madrid, Spain, and Department of Computer Science, University of Dortmund, Germany, E-mail: moraga@cs.uni-dortmund.de

[3]Faculty of Electrical Engineering, University of Belgrade, 11000 Belgrade, Yugoslavia, E-mail: kovacevic_b@kiklop.etf.bg.ac.yu

In this paper we consider particle filters as parameter estimators for recurrent radial basis function (RRBF) networks, and compare their performance to the performance of the extended Kalman filter.

II. STATE SPACE MODEL OF RRBF NETWORK

Without loss of generality we shall consider the RRBF network with one output neuron. The overal response of the RRBF network is given by:

$$f(\mathbf{s}_{k-1},\mathbf{u}_{k-1},\mathbf{w}) = a_0 + \sum_{i=1}^{n_H} a_i \phi_i(\mathbf{s}_{k-1},\mathbf{u}_{k-1},\mathbf{w}) \ . \quad (1)$$

We have used $\mathbf{w}$ to denote the n_w dimensional vector of unknown parameters (bias a_0, weights a_i, centers m_{il}, $m_{ij\tau}$ and widths σ_{il}, $\sigma_{ij\tau}$), and n_H is the number of hidden neurons. The output of the *i*-th hidden neuron is given by:

$$\phi_i(\mathbf{s}_{k-1},\mathbf{u}_{k-1},\mathbf{w}) = \exp\left\{-\sum_{l=1}^{\Delta_s}\left(\frac{s_{k-l}-m_{il}}{\sigma_{il}}\right)^2 - \sum_{j=1}^{n_u}\sum_{\tau=1}^{\Delta_u}\left(\frac{u_{j,k-\tau}-m_{ij\tau}}{\sigma_{ij\tau}}\right)^2\right\} \quad (2)$$

where $\mathbf{s}_{k-1} = [s_{k-1}\cdots s_{k-\Delta_s}]^{\mathrm{T}}_{1\times\Delta_s}$ is the vector of previous network outputs and $\mathbf{u}_{k-1} = [u^{\mathrm{T}}_{k-1}\cdots u^{\mathrm{T}}_{k-\Delta_u}]^{\mathrm{T}}_{1\times\Delta_s}$ is the vector of previous inputs.

The RRBF network dynamics is represented by following Markov, non-linear State Space Model (SSM):

$$x_k = \Phi(x_{k-1},\mathbf{u}_{k-1}) + d_k \ , \quad d_k \sim N(0,Q_k) \quad (3a)$$

$$y_k = Hx_k + v_k \ , \quad v_k \sim N(0,R_k) \quad (3b)$$

where $x_k = \begin{bmatrix} s_k \\ s_{k-1} \\ \vdots \\ s_{k-\Delta_s+1} \\ \mathbf{w}_k \end{bmatrix}$, $\Phi(x_{k-1},\mathbf{u}_{k-1}) = \begin{bmatrix} f(\mathbf{s}_{k-1},\mathbf{u}_{k-1},\mathbf{w}_{k-1}) \\ s_{k-1} \\ \vdots \\ s_{k-\Delta_s+1} \\ \mathbf{w}_{k-1} \end{bmatrix}$, $d_k = \begin{bmatrix} d_{k,s} \\ 0 \\ \vdots \\ 0 \\ d_{k,w} \end{bmatrix}$,

$Q_k = \begin{bmatrix} Q_{k,s} & 0 \\ 0 & Q_{k,w} \end{bmatrix}_{(\Delta_s+n_w)\times(\Delta_s+n_w)}$ and $H = [1 \ 0 \ \cdots \ 0]_{1\times(\Delta_s+n_w)}$.

The process noise d_k and observation noise v_k are assumed to be mutually independent, white, and Gaussian with covariances Q_k and R_k respectively. The state vector x_k is obtained by concatenating hidden variables (in our case they are previous outputs of the RRBF network $\mathbf{s}_k$) and adjustable parameters $\mathbf{w}_k$.

III. BAYESIAN FILTERING

Assuming that observations arrive sequentially in time $y_{0:k} = \{y_k, k = 0,1,...\}$, the task is to estimate sequentially the unknown state process $x_{0:k} = \{x_k, k = 0,1,...\}$. In case of NN

training, of primary concern is estimation of the filtering pdf $p(x_k/y_{0:k})$. If the initial pdf, $p(x_0/y_0)=p(x_0)$ is available (y_0 being the set of no measurements), the pdf $p(x_k/y_{0:k})$ may be obtained recursively in two stages: prediction and update.

The prediction stage uses the system model (1) to predict the state pdf forward from one measurement time to the next. Suppose that the required pdf $p(x_{k-1}/y_{0:k-1})$ at time step $k-1$ is available, than the prior pdf of the state at time *k* is:

$$p(x_k/y_{0:k-1})=\int p(x_k/x_{k-1})p(x_{k-1}/y_{0:k-1})dx_{k-1} \qquad (4)$$

In (3) use has been made of the fact that (1) describes a Markov process: $p(x_k/x_{k-1},y_{0:k-1})=p(x_k/x_{k-1})$. The probabilistic model of the state evolution $p(x_k/x_{k-1})$ is defined by the system equation and the known pdf of the process noise.

In **the update stage** the latest measurement y_k is used to modify the predicted pdf $p(x_k/y_{0:k-1})$ via the Bayes' rule:

$$p(x_k/y_{0:k})=\frac{p(y_k/x_k)}{\int p(y_k/x_k)p(x_k/y_{0:k-1})dx_{k-1}}p(x_k/y_{0:k-1}) \qquad (5)$$

The likelihood $p(y_k/x_k)$ is defined by the measurement model (2) and the known statistics of measurement noise. The resulting density is the posterior pdf $p(x_k/y_{0:k})$.

For the state space model (3) of the RRBF network, the extended Kalman filter produces an analytic solution of (5) by linearizing the dynamic equation (3a):

$$x(k)=\Phi_k(\hat{x}_{k-1},\mathbf{u}_{k-1})+F_k(x_k-\hat{x}_{k-1})+d_k, \qquad (6)$$

where $F_k=[\nabla_x \Phi_k(\hat{x}_{k-1},\mathbf{u}_{k-1})^{\mathrm{T}}]^{\mathrm{T}}$ and $\hat{x}_{k-1}$ is the estimate of x_{k-1} given observations Y_{k-1}. The relevant pdfs are approximated by Gaussians. Based on these approximations the extended Kalman filter equations are obtained:

State prediction:	$\hat{x}_k^- = \Phi_k(\hat{x}_{k-1},\mathbf{u}_{k-1})$	(7a)
Prediction error cov.:	$P_k^- = F_k P_k F_k^{\mathrm{T}} + Q_k$	(7b)
State estimation:	$\hat{x}_k = \hat{x}_k^- + K_k e_k$	(7c)
Estimation error cov.:	$P_k = (I-K_kH)\cdot P_k^-$	(7d)

where $e_k = y_k - H\cdot\hat{x}_k^-$ denotes innovation, $S_k = HP_k^-H^T + R_k$ is the innovation covariance, and $K_k = P_k^-H^T S_k^{-1}$ is the Kalman gain.

IV. SEQUENTIAL IMPORTANCE SAMPLING

The density $p(x_{0:k}/y_{0:k})$ constitutes the complete solution to the sequential estimation problem, from which we can obtain the marginal filtering pdf $p(x_k/y_{0:k})$ and the expectation:

$$\mathrm{E}_{p(x_{0:k}/y_{1:k})}[f_k(x_{0:k})]=\int f_k(x_{0:k})p(x_{0:k}/y_{0:k})dx_{0:k} \qquad (8)$$

for any $p(x_{0:k}/y_{0:k})$ - integrable $f_k: R^{(k+1)\cdot n_x}\to R$.

Assuming that we are able to simulate N i.i.d. random samples $\{x_{0:k}^{(i)}; i=1,2,...N\}$ according to $p(x_{0:k}/y_{0:k})$, the following estimate of the expectation (8) can be obtained:

$$\overline{\mathrm{E}_{p(x_{0:k}/y_{0:k})}[f_k(x_{0:k})]}=\frac{1}{N}\sum_{i=1}^{N}f_k(x_{0:k}^{(i)}) \qquad (9)$$

However, the pdf $p(x_{0:k}/y_{0:k})$ is unknown and one can not sample from it. Instead we can sample from the known, easy to sample importance function $\pi(x_{0:k}/y_{0:k})$, and use the following substitutions in equation (8):

$$\begin{aligned}\mathrm{E}_{p(\cdot/y_{1:k})}[f_k(x_{0:k})]&=\int f_k(x_{0:k})\frac{p(x_{0:k}/y_{1:k})}{\pi(x_{0:k}/y_{1:k})}\pi(x_{0:k}/y_{1:k})dx_{0:k}\\&=\frac{1}{p(y_{0:k})}\int f_k(x_{0:k})\omega_k(x_{0:k})\pi(x_{0:k}/y_{1:k})dx_{0:k}\end{aligned} \qquad (10)$$

where $\omega_k(x_{0:k})=p(y_{0:k}/x_{0:k})\,p(x_{0:k})/\pi(x_{0:k}/y_{0:k})$ are unnormalized importance weights. The normalizing constant $p(y_{0:k})$ can be rewritten in the following form:

$$p(y_{0:k})=\int \omega_k(x_{0:k})\pi(y_{0:k}/x_{0:k})\,dx_{0:k} \qquad (11)$$

By substituting (11) in (10), we obtain:

$$\mathrm{E}_{p(\cdot/y_{1:k})}[f_k(x_{0:k})]=\frac{\mathrm{E}_{\pi(\cdot/y_{1:k})}[f_k(x_{0:k})\omega_k(x_{0:k})]}{\mathrm{E}_{\pi(\cdot/y_{1:k})}[\omega_k(x_{0:k})]} \qquad (12)$$

and the estimate of $\mathrm{E}_{p(\cdot/y_{1:k})}[f_k(x_{0:k})]$ is given by:

$$\overline{\mathrm{E}_{p(\cdot/y_{1:k})}[f_k(x_{0:k})]}=\sum_{i=1}^{N}f_k(x_{0:k}^{(i)})\tilde{\omega}_k^{(i)} \qquad (13)$$

where $\tilde{\omega}_k^{(i)}$ are the normalized importance weights:

$$\tilde{\omega}_k^{(i)}=\omega_k^{(i)}(\textstyle\sum_{j=1}^{N}\omega_k^{(i)})^{-1} \qquad (14)$$

Sequential Monte Carlo estimation of the posterior pdf $p(x_{0:k}/y_{0:k})$, without modifying subsequently the past simulated trajectories $\{x_{0:k}^{(i)}; i=1,...,N\}$ is possible if the importance function has the following form:

$$\pi(x_{0:k}/y_{0:k})=\pi(x_{0:k-1}/y_{0:k-1})\pi(x_k/x_{0:k-1},y_{0:k}) \qquad (15)$$

By using (15), and taking into account that the state corresponds to a Markov process: $p(x_{0:k})=p(x_0)\prod_{j=1}^{k}p(x_j/x_{j-1})$, and the observations are conditionally independent given the states $p(y_{0:k}/x_{0:k})=\prod_{j=0}^{k}p(y_j/x_j)$, we obtain the recursive formula for computing the unnormalized importance weights:

$$\omega_k(x_{0:k})=\omega_{k-1}(x_{0:k-1})\frac{p(y_k/x_k)p(x_k/x_{k-1})}{\pi(x_k/x_{0:k},y_{1:k})} \qquad (16)$$

If $\pi(x_k/x_{0:k-1},y_{1:k})=\pi(x_k/x_{k-1},y_k)$ and if only a filtering pdf $p(x_k/y_{1:k})$ is required, the path $x_{0:k-1}^{(i)}$ can be discarded, as well as the history of observations $y_{0:k-1}$. Thus, the unnormalized importance weights are given by:

$$\omega_k^{(i)}\propto\omega_{k-1}^{(i)}\frac{p(y_k/x_k^{(i)})p(x_k^{(i)}/x_{k-1}^{(i)})}{\pi(x_k^{(i)}/x_{k-1}^{(i)},y_k)} \qquad (17)$$

Using the sampled particles $x_k^{(i)}\sim\pi(x_k/x_{k-1}^{(i)},y_k)$ and corresponding normalized weights $\tilde{\omega}_k^{(i)}=\omega_k^{(i)}(\sum_{j=1}^{N}\omega_k^{(i)})^{-1}$,

$i = 1,...N$, the posterior filtering density is approximated by:

$$\hat{p}(x_k / y_{0:k}) = \sum_{i=1}^{N} \tilde{\omega}_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{18}$$

Finally, the Sequential Importance Sampling (SIS) step can be summarized in the following algorithm.

Algorithm $[\{x_k^{(i)}, \omega_k^{(i)}\}_{i=1}^N] = \mathbf{SIS}[\{x_{k-1}^{(i)}, \omega_{k-1}^{(i)}\}_{i=1}^N, y_k]$

%Sample

$x_k^{(i)} \sim \pi(x_k / x_{k-1}^{(i)}, y_k),\ i = 1,...,N$

%Evaluate importance weights

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \frac{p(y_k / x_k^{(i)}) p(x_k^{(i)} / x_{k-1}^{(i)})}{\pi(x_k^{(i)} / x_{k-1}^{(i)}, y_k)},\quad i = 1,...,N$$

%Normalize the importance weights

$\tilde{\omega}_k^{(i)} = \omega_k^{(i)} / \sum_{j=1}^{N} \omega_k^{(i)},\ i = 1,...,N$

end

V. Sampling Importance Resampling

Degeneracy of the SIS algorithm. A very important drawback of sequential update (17) is that the variance of the importance weights increases over time [3]. In practice this means that one of the importance weights will tend to one while others will tend to zero. Thus, the effective particle size reduces from N to almost 1. In order to deal with this problem, strategies like proper selection of importance function and resampling have to be considered.

Selection of the importance function. Conventional particle filters use the transition prior as the importance density $\pi(x_k / x_{k-1}^{(i)}, y_k) = p(x_k / x_{k-1}^{(i)})$, which yields importance weights $\omega_k^{(i)} = \omega_{k-1}^{(i)} p(y_k / x_k^{(i)})$. This method is easy to implement but inefficient because the state space is explored without knowledge of the observations. However, Doucet [4] proved that the importance function $p(x_k / x_{k-1}^{(i)}, y_k)$, introduced in [11] minimizes the variance of the importance weights. For this distribution, unnormalized importance weights (17) are given by $\omega_k^{(i)} = \omega_{k-1}^{(i)} p(y_k / x_{k-1}^{(i)})$. The SSM of the RRBF network dynamics, defined by (3) belongs to an important class for which the optimal importance function and $p(y_k / x_{k-1}^{(i)})$ can be analytically obtained. For the importance function we obtain:

$$\begin{aligned} &p(x_k / x_{k-1}, y_k) = N(\bar{x}_k, \Sigma_k), \\ &\Sigma_k = (Q_k^{-1} + H^{\mathrm{T}} R_k^{-1} H)^{-1} \\ &\bar{x}_k = \Sigma_k (Q_k^{-1} f(x_{k-1}) + H^{\mathrm{T}} R_k^{-1} y_k) \end{aligned} \tag{19}$$

The density $p(y_k / x_{k-1})$ is obtained as:

$$\begin{aligned} p(y_k / x_{k-1}) \propto \exp(-0.5 \cdot (y_k - H\Phi_k(x_{k-1}))^{\mathrm{T}} \cdot \\ (HQ_k H^{\mathrm{T}} + R_k)^{-1} \cdot (y_k - H\Phi_k(x_{k-1})) \end{aligned} \tag{20}$$

Resampling. The basic idea of resampling methods is to eliminate particles with small importance weights and multiply particles with large importance weights in order to limit the degeneracy of the sequential importance sampling algorithm. A new set of equally weighted samples $\{x_k^{(i)}, N^{-1}\}_{i=1}^N$ is obtained by resampling (with replacement) N times from an approximate discrete representation of $p(x_k / y_{0:k})$ given by (20), so that $\Pr(x_k^{(i)*} = x_k^{(j)}) = \tilde{\omega}_k^{(j)}$.

Some authors advocate the idea that resampling should be used only if the effective particle size N_{eff} is below a fixed threshold [5]. Effective particle size N_{eff} can be estimated as follows [6]:

$$\hat{N}_{eff} = (\textstyle\sum_{i=1}^{N} (\tilde{\omega}_k^{(i)})^2)^{-1} \tag{21}$$

We have implemented and used in our experiments the stratified/systematic resampling described in [1], summarized in the following algorithm.

Algorithm $\{x_k^{(i)*}, \omega_k^{(i)*}\}_{i=1}^N] = \mathbf{RESAMPLE\_ss}[\{x_k^{(i)}, \omega_k^{(i)}\}_{i=1}^N]$

$u \sim U[0, 1/N]$; $CUM_\omega = 0$; $j = 1$; $i = 0$; $k = 0$;

$ind = 1,...,N$;

while $(u < 1)$;

if $u < CUM_\omega$

$u = u + 1/N$; $n_{Copies}^{(l)} = n_{Copies}^{(l)} + 1$

else

$u_1 \sim U[0, N - j]$; $i = j + [u1]$;

$CUM_\omega = CUM_\omega + \omega_k^{(i)}$; $l = ind(i)$;

$ind(i) = ind(j)$; $\omega_k^{(i)} = \omega_k^{(j)}$; $j = j + 1$;

end %if

end %while

$i = 1$;

for $l = 1 : N$

if $n_{Copies}^{(i)} > 0$

$x_k^{(i:i+n_{Copies}^{(i)}+1)*} = x_k^{(l)}$; $i = i + n_{Copies}^{(l)}$;

end %if

end %for

$\omega_k^{(i)*} = 1/N$, $i = 1,...,N$

end

VI. Examples

We consider the identification of the dynamic system, described by following SSM:

$$x_k = \frac{x_{k-1} x_{k-2} x_{k-3} u_{k-2} (x_{k-3} - 1) + u_{k-1}}{1 + x_{k-2}^2 + x_{k-3}^2} + d_k \tag{22a}$$

$$y_k = x_k + v_k \tag{22b}$$

Example 1. The observations y_k are obtained for input:

$$u(k) = \begin{cases} 0.5(\cos(2\pi k/250) + \sin(2\pi k/25)), & 0 < k \le 350 \\ \cos(2\pi k/250), & 350 < \mathrm{k} \le 500 \\ 0.8\cos(2\pi k/250) + 0.2\sin(2\pi k/25), & 500 < k \le 800 \end{cases} \tag{23}$$

Process noise d_k was drawn from $N(0, 1.\mathrm{e}\text{-}4)$ and observation noise v_k was drawn from $N(0,0.1)$. Parameters of the RRBF network with 4 hidden neurons were estimated using the following filters.

EKF – Extended Kalman filter with noise covariance matrices: $Q_{k,s}(1,1) = 1.e-4$, $Q_{k,s}(i,j) = 0$, $\forall i, j \neq 1$. $Q_{k,w}$ is a diagonal matrix with entries equal to 1.e-4; $R = 0.1$.

PF(prior) – Particle filter with transition prior as the impor-

tance function. $Q_{k,s}$ is the same as for the EKF. $Q_{k,w}$ is a diagonal matrix with entries equal to 5.e-3; $R = 0.1$.

PF(optimal) – Particle filter with optimal importance function (19). Noise covariances are the same as for the PF(prior).

We have conducted 100 simulations for different initial parameter values and noise sequences. In order to observe the RRBF network training, for each simulation the root mean squared error using the difference between the true state $x(k)$ and output of the RRBF network (predicted state) was calculated. The resampling was performed in every time step.

For this example particle filters converged in every simulation with mean RMSE presented in Table I, while EKF converged only in 19 simulations out of 100. The mean of RMSE calculated for these simulations was 2.31e-1.

TABLE I
MEAN AND VARIANCE OF RMSE ($d_k \sim N(0,1.\text{e-}4)$, $v_k \sim N(0,0.1)$

	EKF	PF (prior)	PF (optim.)
mean(RMSE)	–	1.89e-1	1.86e-1
var(RMSE)	–	3.34e-4	2.86e-4

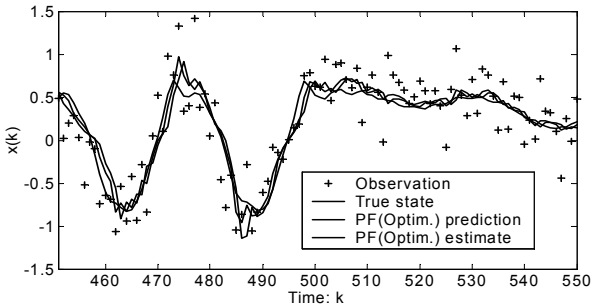

Fig. 1. State prediction (RRBF network output) and estimation using PF (optimal) ($d_k \sim N(0,1.e-4)$, $v_k \sim N(0,0.1)$

Example 2. As in the first example the observations y_k are obtained for the input defined by (23). Process noise d_k was drawn from $N(0,0.01)$ and observation noise v_k was drawn from $N(0,1.\text{e-}4)$. Parameters of the RRBF network with 4 hidden neurons were estimated using following filters:

EKF – Noise covariance matrices: $Q_{k,s}(1,1) = 0.01$, $Q_{k,s}(i,j) = 0$, $\forall i, j \neq 1$. $Q_{k,w}$ is a diagonal matrix with entries equal to 0.01; $R = 1.e-4$.

PF(prior) – $Q_{k,s}$, $Q_{k,w}$ are the same as for the EKF; $R = 0.1$.

PF(optimal) – $Q_{k,s}$, $Q_{k,w}$ and R are the same as for the EKF.

For this example, the EKF converged in 66 simulations out of 100. Mean and variance of the RMSE for these simulations are given in Table II.

TABLE II
MEAN AND VARIANCE OF RMSE ($d_k \sim N(0,0.01)$, $v_k \sim N(0,1.\text{e-}4)$

	EKF	PF (prior)	PF (optimal)
mean(RMSE)	2.56e-1	1.94e-1	1.68e-1
var(RMSE)	5.24e-3	1.29e-3	1.37e-4
Time(sec)	5.68	44.44	116.82

As for the PF(prior), it could not converge when R was set to the variance of the observation noise 1.*e*-4. However, when R was empirically increased to $R = 0.01$, the PF(prior) converged in every one of 100 simulations and the mean of the RMSE was slightly worse than for the PF(optimal) (Table II).

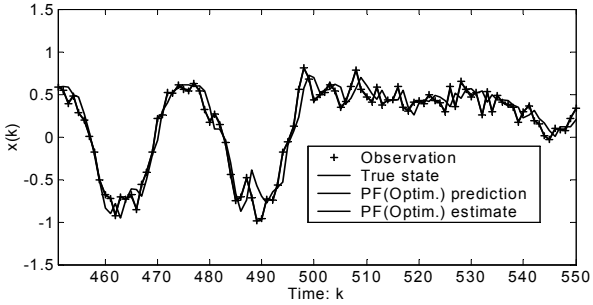

Fig. 2. State prediction (RRBF network output) and estimation using PF (optimal) ($d_k \sim N(0,0.01)$, $v_k \sim N(0,1.\text{e-}4)$

VII. CONCLUDING REMARKS

We have applied the sequential Monte Carlo methods in the form of particle filters in RRBF network parameter and state estimation The considered algorithms demonstrate smaller estimation error and improve robustness in convergence (regarding the initial parameter values), compared to the well known extended Kalman filter.

REFERENCES

[1] Carpenter, J., Clifford, P. and Fearnhead, P.: "An improved particle filter for nonlinear problems," Technical Report, Department of Statistics, Oxford University, England 1997

[2] de Freitas, J. F. G., Niranjan, M. and Gee, A.H.: "Hierarchical Bayesian-Kalman models for regularization and ARD in sequential learning," Technical Report CUED/F-INFENG/TR 307, Cambridge University., 1997.

[3] de Freitas, J. F. G., Niranjan, M. Gee, A.H., Doucet, A.: "Sequential Monte Carlo methods for optimization of neural network models," Technical Report CUED/F-INFENG/TR 328, Cambridge University., 1998.

[4] Doucet, A.: "On sequential simulation-based methods for Bayesian filtering, "Technical Report CUED/F-INFENG/TR 310, Cambridge University, 1998.

[5] Liu, J. S. and Chen, R.: "Sequential Monte Carlo methods for dynamic systems", *Jour. of Amer. Stat. Assoc.* Vol. 93, pp 1031-1041, 1998

[6] Kong, A., Liu, J. S. and Wong, W. H.: "Sequential imputations and Bayesian missing data problems," *Jour. of Amer. Stat. Assoc.*Vol. 89 (425), pp 278-288, 1994.

[7] Todorović, B., Stanković, M, Todorović-Zarkula, S.: Structurally adaptive RBF network in non-stationary time series prediction, In *Proc. IEEE AS-SPCC*, Lake Louise, Alberta, Canada, Oct. 1-4 (2000) pp. 224-229

[8] Todorović, B., Stanković, M., Moraga, C.: "Extended Kalman Filter trained Recurrent Radial Basis Function Network in Nonlinear System Identification," *Proc. of ICANN 2002*, Spain, LNCS 2415, pp 819-824, Springer, August 2002

[9] Todorović, B., Moraga C., Stanković, M., Kovačević, B.: "Nonlinear/non-Gaussian sequential estimation applied to neural networks: Theory" *Proc. of ICEST 2002*, Niš, October 2002.

[10] Williams, R.J.: "Some observations on the use of the extended Kalman filter as a recurrent network learning algorithm," Technical Report NU_CCS_92-1. Boston: Northeastern University, College of Computer Science, 1992

[11] Zaritskii, V. S., Svetnik, V. B., Shimelevich, L.I.: "Monte Carlo Technique in Problems of Optimal Data Processing," Auto. Reo. Cont., vol. 12, 1975, pp. 95-103.