6.2 Supervised Training of Recurrent Neural Networks

Recurrent neural networks (RNNs) are well known for their use in chaotic dynamic reconstruction, among other applications. Chaotic dynamics are commonly observed in a wide variety phenomena from molecular vibrations to satellite motions. In most cases, the underlying governing equations of chaotic dynamics are difficult to obtain. In such cases, they may be replaced with RNNs. In this experiment, Bayesian filter-trained RNNs were considered. The well-known chaotic Mackey-Glass system was used to generate both the training and test data. The EKF, the CDKF, and the CKF were employed for the purpose of supervised training of RNNs. Particle filters were not considered for the following reason: The supervised training of RNNs involves a large number of weights to be estimated. Hence, an enormous amount of particles is required to completely capture this huge state-space volume as outlined in Chapter 2. Simply put, particle filters are computationally quite demanding for this application.

Chaotic Mackey-Glass Attractor. The Mackey-Glass equation is often used to model the production of white-blood cells in Leukemia patients, and given by the delay differential equation [75]:

$$\frac{du_t}{dt} = 0.1u_t + \frac{0.2u_{t-\Delta}}{1+u_{t-\Delta}^{10}}, \tag{6.1}$$

where the delay $\Delta = 30$. To sample the time-series at discrete time steps, (6.1) was numerically integrated using the forth-order Runge-Kutta method with a sampling period of $T = 6$ s, and initial condition $u_t = 0.9$, for $0 \leq t \leq \Delta$. Given a chaotic

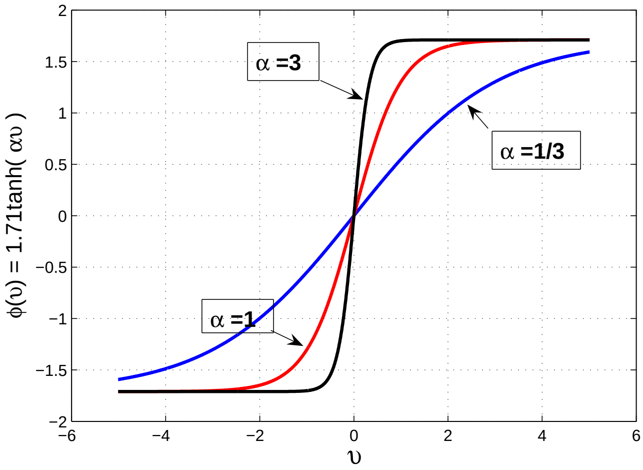


Figure 6.3: Effect of α on the shape of the activation function $\varphi(\upsilon) = 1.71\tanh(\alpha\upsilon)$.

system, it is known that the next data sample $u_{k+\tau}$ can be predicted from a properly chosen time sequence $\mathbf{u}_k = [u_k \ \ u_{k-\tau} \ldots u_{k-[d_E-2]\tau} \ \ u_{k-[d_E-1]\tau}]$, where d_E and τ are called the embedding dimension and the embedding delay, respectively. For the chaotic Mackey-Glass system, d_E and τ were chosen to be seven and one, respectively.

RNN Architecture. Bayesian filter-trained RNNs were used to predict the chaotic Mackey-Glass time-series data. The structure of a RNN was chosen to have seven inputs representing an embedding of the observed time-series, one output, and one self-recurrent hidden layer with five neurons. Hence, the RNN has a total of 71 connecting weights (bias included). The linear activation function was used by the output neuron, whereas all the hidden neurons used a hyperbolic tangent function of the form

$$\varphi(\upsilon) \ = \ 1.71\tanh(\alpha\upsilon),$$

where α was assumed to take values ranging from 1/3 to 3. As shown in Figure 6.3, the hyperbolic tangent function is 'mildly' nonlinear (that is, close to a linear function) around its origin when $\alpha = 1/3$. Its nonlinearity increases with α, and behaves closely similar to a switch when $\alpha = 3$.

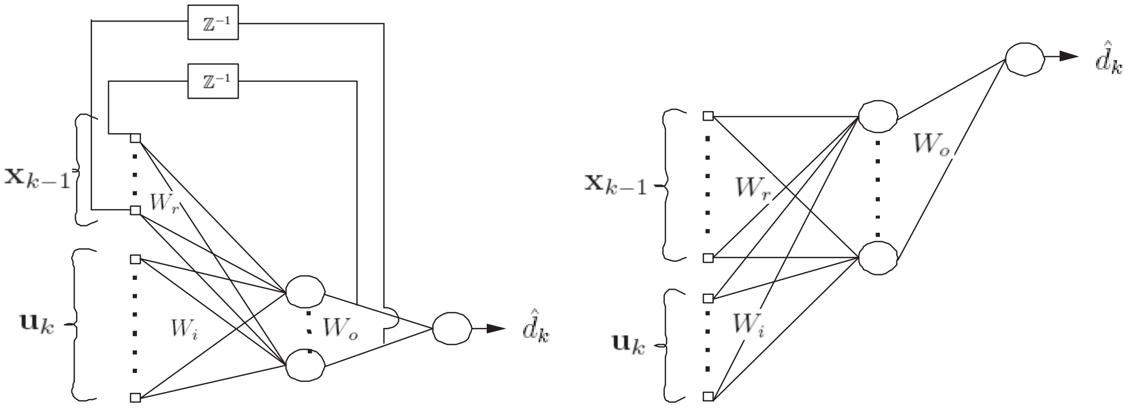
State-Space Model. To estimate the weight parameters using a Bayesian filter, they are typically assumed to be Gaussian random variables. Specifically, the weight variables are assumed to follow the first-order noisy autoregressive model, and the state-space model can therefore be written as

$$\begin{aligned} \mathbf{w}_k &= \mathbf{w}_{k-1} + \mathbf{q}_{k-1} \\ d_k &= W_o \varphi\big(W_r \mathbf{x}_{k-1} + W_i \mathbf{u}_k\big) + r_k, \end{aligned}$$

where

- The process noise $\mathbf{q}_k$ is assumed to be zero-mean Gaussian with covariance Q_{k-1}
- The measurement noise r_k is assumed to be zero-mean Gaussian with variance R_k
- The internal state of the RNN or the output of the hidden layer at time $(k-1)$ is denoted by $\mathbf{x}_{k-1}$ (Figure 6.4)
- The desired output d_k acts as the measurement
- W_i, W_r and W_o are input, recurrent and output weight matrices of appropriate dimensions; the weight vector $\mathbf{w}_k$ is obtained by grouping elements from W_i, W_r and W_o in 'some' orderly fashion

Data. A chaotic time sequence of length 1000 was generated, the first half of which was used for training and the rest for testing. To train the RNN using the



(a)

(b)

Figure 6.4: Schematic diagrams (a). Original RNN (b). Unfolded RNN of unity truncation depth.

CKF, 10 epochs/run were made. Each epoch was obtained from a 107 time-step long subsequence, starting from a randomly selected point. That is, each epoch consisted of 100 examples, all of which were gleaned by sliding a window of length eight over the subsequence. The weights were initialized to be zero-mean Gaussian with a diagonal covariance of $0.5I_w$; Q_{k-1} was made to decay such that $Q_{k-1} = (\frac{1}{\lambda} - 1)P_{k-1|k-1}$, where $\lambda \in (0, 1)$ is the "forgetting factor" as defined in the recursive least-squares algorithm [44]; this approximately assigns exponentially decaying weights to past measurements; λ was fixed at 0.9995, and R_k at 5×10^{-3} across the entire epoch; the state of the RNN at $t = 0$, $\mathbf{x}_0$, was assumed to be zero.

Unlike the CKF, which relies on integration, the EKF and the CDKF use gradient information, which in turn necessitate the use of the truncated backpropagation through time method. To unfold the recurrent loop of the neural network, a truncation depth of unity was found to be sufficient in this experiment (see Figure 6.4).

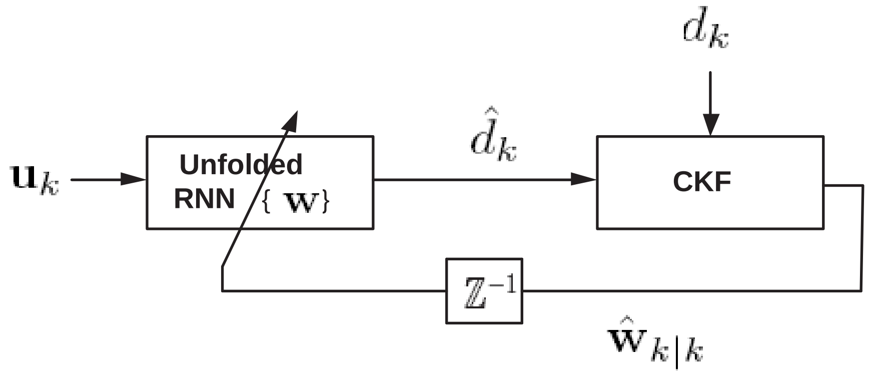


Figure 6.5: CKF-based supervised training of RNN.

Figure 6.5 illustrates how the CKF sequentially updates the weights from the input-output pair during a training phase.

Performance Metric. During the test phase, RNNs were initialized with a 20 time-step long test sequence and allowed to run autonomously using their own output for the next 100 steps. To fairly compare the performance of various filter-trained RNNs, 50 independent training runs were made for each value of α. As a performance metric, the ensemble-averaged cumulative absolute error, which is defined by

$$e_k = \frac{1}{50}\sum_{r=1}^{50}\sum_{i=1}^{k}|d_i^{(r)} - \hat{d}_i^{(r)}|; \quad k = 1, 2, \dots 100,$$

was used.

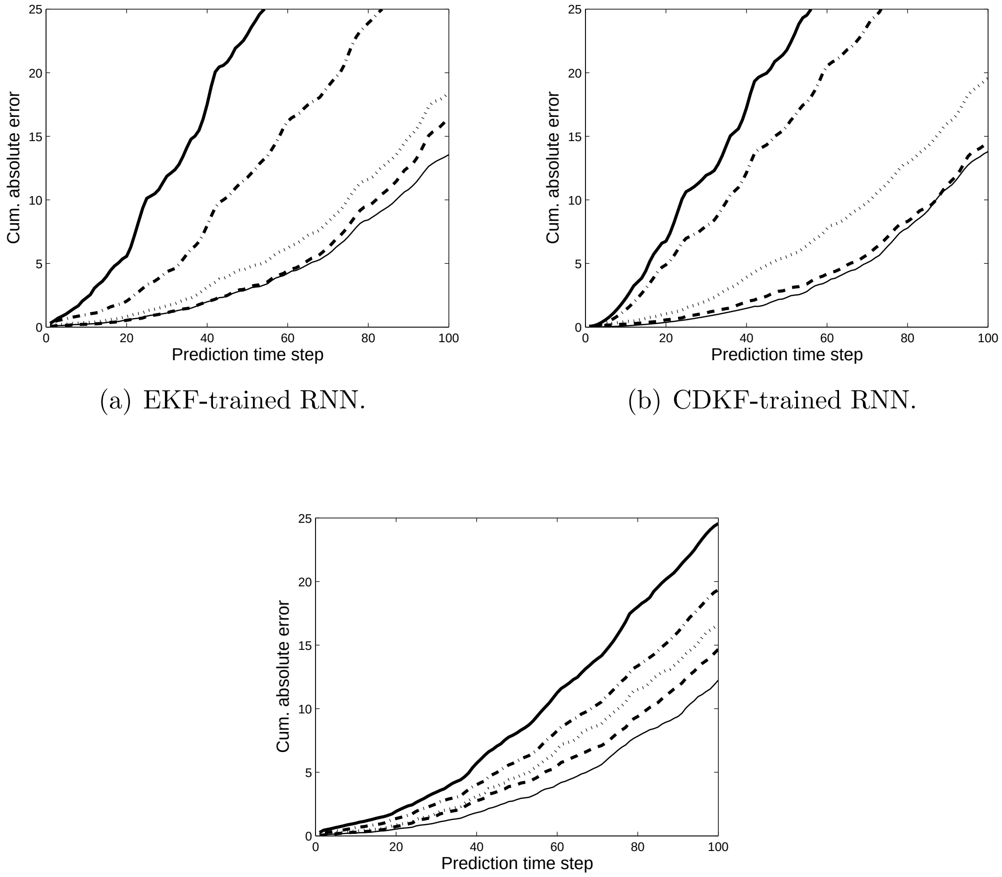
Observations. The long-term accumulative prediction error is expected to increase exponentially with time for the following two reasons:

- Chaotic systems are highly sensitive even to a slight perturbation in their present state, popularly referred to as the butterfly effect [90].

- The prediction error is amplified at each time step due to the closed loop structure.

From Figures 6.6(a) and 6.6(b), it is observed that the RNNs trained with the EKF and the CDKF break down at $\alpha = 2$ and beyond. The CKF-trained RNN performs reasonably well even when $\alpha = 3$, for which the hyperbolic tangent function is 'severely' nonlinear (Figure 6.6(c)). The reason is that the CKF tends to find a better local minimum of the cost function in the weight space than the EKF or the CDKF.

To visualize whether the CKF-trained RNN has captured the true dynamics of the chaotic time series, the phase plot– a three-dimensional diagram with its axes denoting the RNN outputs $\hat{d}_{k+2}, \hat{d}_{k+1}$, and $\hat{d}_k$– was constructed. The desired result is that the RNN closely approximate the true dynamics of the Mackey-Glass system. Figures 6.7(a), 6.7(b) and 6.7(c) show the phase plots of the true dynamics, and the reconstructed dynamics when $\alpha = 1/3$ and $\alpha = 3$, respectively. When $\alpha = 1/3$ the reconstructed phase plot closely resembles the true phase plot as desired; whereas it is not exactly the case when $\alpha = 3$.

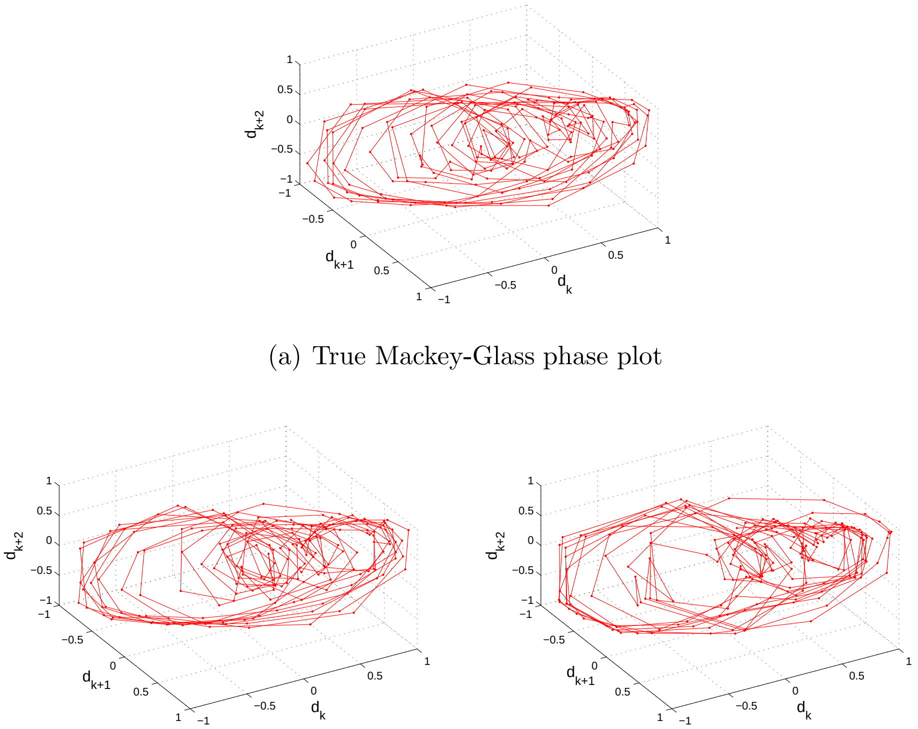


(a) EKF-trained RNN.

(b) CDKF-trained RNN.

(c) CKF-trained RNN.

Figure 6.6: Effect of nonlinearity on the autonomous-prediction performance. Nonlinearity is controlled by the parameter α, and the prediction performance is measured by the ensemble-averaged cumulative absolute error criterion ($\alpha = 1/3$ (solid-thin), 2/3 (dashed), 1 (dotted), 2 (dash-dot), and 3 (solid-thick))



(a) True Mackey-Glass phase plot

(b) Reconstructed plot when $\alpha = \frac{1}{3}$

(c) Reconstructed plot when $\alpha = 3$

Figure 6.7: Comparison of two different reconstructed phase plots with the true plot

6.3 Model-Based Signal Processing

In the second experiment, the empirical model of the chaotic Mackey-Glass system was built from the clean input-output data. In contrast, provided the noisy measurements of a dynamic system, the objective of this third experiment was to build a nonlinear empirical model of the dynamic system from noisy measurements for the following purposes:

- To denoise a given test signal (signal enhancement)
- To statistically decide whether the denoised test signal belongs to the empirical model (signal detection)

In this experiment, the idea of directly training RNNs in the supervised mode must be abandoned because the desired (teacher) output is noisy. A similar situation arises in many important real-life applications such as speech signal enhancement, image processing, decoding of symbols transmitted through a noisy wireless channel, and fault diagnosis. To achieve the above objectives, a systematic filtering setup is important.

Cooperative Filtering for Signal Enhancement

The objective of cooperative filtering is to construct an empirical model using (pseudo-) clean data extracted from the noisy data. To accomplish this objective, two distinct estimators, namely, the signal estimator and the weight (parameter) estimator, are coupled to operate in a cooperative manner (see Figure 6.8). At each time instant, the weight parameters of the RNN are estimated from the latest

Figure 6.8: Cooperative filtering illustrating interactions between the signal estimator (SE) and the weight estimator (WE); the labels TU and MU denote 'Time Update' and 'Measurement Update', respectively)

signal estimate, and the signal itself is estimated from the latest weight estimate, appropriately.

Data. To generate a noisy time series, the chaotic Mackey-Glass time series was considered again, but it was corrupted by additive white Gaussian noise this time. The signal-to-noise ratio was fixed at 10 dB. As in the second experiment, the architecture of a RNN was chosen to be 7-5R-1, and a hyperbolic tangent function of the form $\varphi(v) = \tanh(v)$ was used.

State-Space Models. The dynamic state-space model for the signal estimator can be written as

$$\begin{aligned} \mathbf{u}_k &= \mathbf{f}(\mathbf{u}_{k-1}, \hat{\mathbf{w}}_{k-1|k-1}, \mathbf{x}_{k-2}) + [1 \ \ 0 \ldots 0]v_{k-1} && (6.2) \\ z_k &= [1 \ \ 0 \ldots 0]\mathbf{u}_k + e_k, && (6.3) \end{aligned}$$

where

- $\mathbf{u}_k = [u_k \;\; u_{k-1} \ldots u_{k-6}]$ denotes the data window to be estimated
- The state transition function

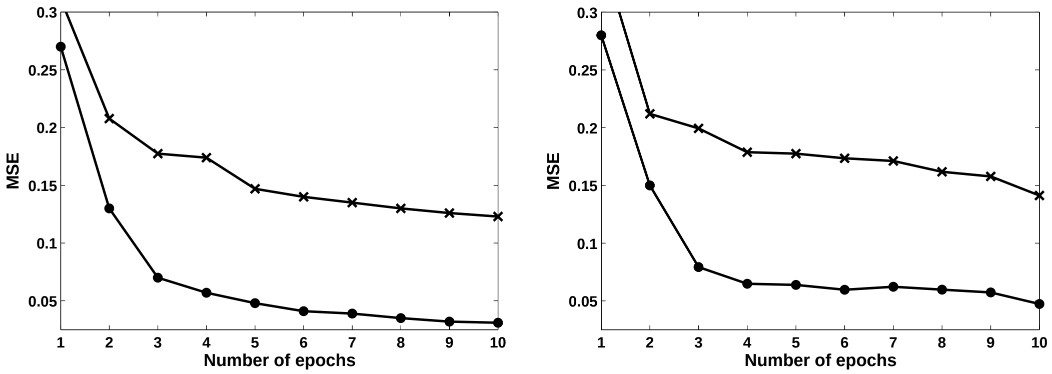
$$\mathbf{f}(.,.,.) \;\; = \;\; \begin{pmatrix} \hat{W}_{o,k-1|k-1}\varphi\big(\hat{W}_{r,k-1|k-1}\mathbf{x}_{k-2} + \hat{W}_{i,k-1|k-1}\mathbf{u}_{k-1}\big) \\ \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \begin{pmatrix} u_{k-1} \\ \vdots \\ u_{k-6} \end{pmatrix} \end{pmatrix}$$

- The measurement noise e_k was assumed to be $e_k \sim \mathcal{N}(0, \sigma_e^2)$, where the variance σ_e^2 was computed from the prescribed value of the signal-to-noise ratio
- The process noise v_{k-1} was assumed to be $v_{k-1} \sim \mathcal{N}(0, \sigma_v^2)$, where the variance σ_v^2 was fixed to be 10% of σ_e^2; the final result was not sensitive to this choice of percentage as long as it was below 100%
- The initial signal estimate was assumed to be zero with unity covariance.

To set the stage for the state-space model of the weight estimator, (6.3) is rewritten in terms of $\mathbf{w}$ as follows:

$$\begin{aligned} z_k = \mathbf{u}_k[1] + e_k \;\; &= \;\; W_o\varphi\big(W_r\mathbf{x}_{k-2} + W_i\mathbf{u}_{k-1}\big) + v_{k-1} + e_k \\ &\approx \;\; W_o\varphi\big(W_r\mathbf{x}_{k-2} + W_i\hat{\mathbf{u}}_{k-1|k-1}\big) + r_k, \end{aligned}$$

where the measurement noise $r_k \sim \mathcal{N}(0, \sigma_e^2 + \sigma_v^2)$. The state-space model of the weight



(a) Training Phase

(b) Test Phase

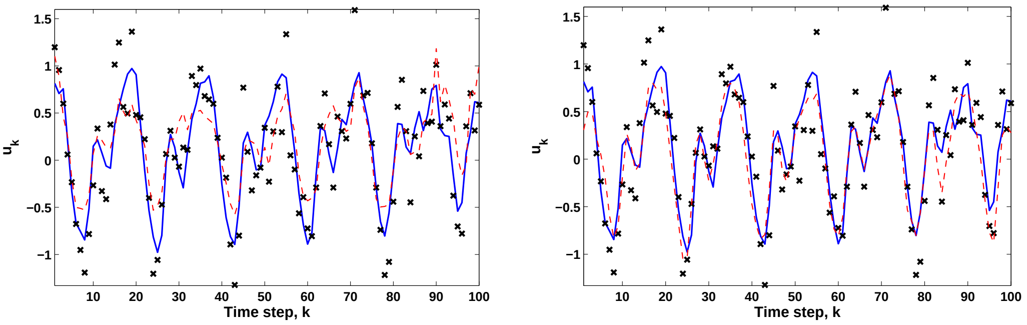
Figure 6.9: Ensemble-averaged (over 50 runs) Mean-Squared Error (MSE) Vs. number of epochs (x- EKF, filled circle- CKF).

estimator is therefore given by

$$\begin{aligned} \mathbf{w}_k &= \mathbf{w}_{k-1} + \mathbf{q}_{k-1} \\ z_k &= W_o \varphi\big(W_r \mathbf{x}_{k-2} + W_i \hat{\mathbf{u}}_{k-1|k-1}\big) + r_k. \end{aligned}$$

As shown in Figure 6.8, the cooperative filtering system functions only with inputs in a manner similar to unsupervised training.

To fairly compare the performance of the CKF-trained RNN against the EKF and the CDKF-trained RNNs, 50 independent training and test runs were made, each of which consisted of 10 epochs. Each training epoch consisted of a subsequence of 100 examples. During the test phase (that is, at the end of each training epoch here), the trained RNN was presented with a test sequence of length 100. Thus, the ensemble-averaged (over 50 runs) MSE was computed in the course of training and test phases (see Figures 6.9(a) and 6.9(b)). In the course of the test phase, the



(a) EKF

(b) CKF

Figure 6.10: Representative test signal before and after cleaning (x- noisy signal (or measurements), dotted thin- signal after cleaning, thick- original clean signal)

weight estimator remained turned off. As shown in Figures 6.9(a) and 6.9(b), the CKF improves performance by a discernable margin in both the training and test phases.

Figures 6.10(a) and 6.10(b) show the representative cleaned test signals obtained from the EKF and the CKF, respectively, at the end of the tenth epoch. The CKF significantly improves the quality of the signal as compared to the EKF.

Signal Detection

Motivated by the problem of detecting targets buried in sea clutter [122, 43], the third experiment was further augmented to deal with a signal detection scenario. To systematically perform signal detection, the consistency check– making a statistical decision whether the test signal is consistent with the trained model– based on the normalized innovations squared (NIS) statistic of signal estimators was introduced.

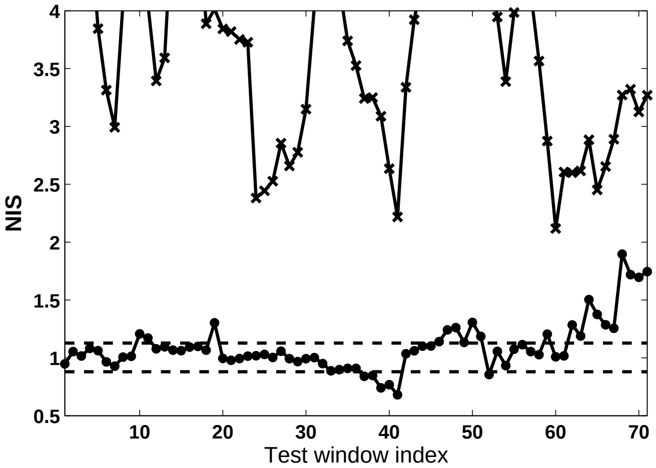


Figure 6.11: Normalized innovations squared (NIS) statistic Vs. test window index (x- EKF, filled circle- CKF, dotted thick- 95% confidence intervals).

Under the hypothesis that the test signal is consistent, the NIS statistic, defined by

$$\epsilon_k = [\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}]^T P_{zz,k|k-1}^{-1} [\mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}],$$

is a realization of the chi-squared distribution with n_z degrees of freedom, where n_z is the dimension of the measurement vector [9].

In this experiment, the NIS statistic of the test data was computed as follows: The test data of length 100 was divided into a number of overlapping data windows of length $K = 10$. Two adjacent windows were separated by one time step. Thus, we were able to obtain 71 data windows. The ensemble-averaged (over $N = 50$ runs) NIS statistic for all these windows were then computed. For example, the NIS statistic of the first window was computed as

$$\bar{\epsilon}(1) = \frac{1}{NK} \sum_{n=1}^{N=50} \sum_{k=1}^{K=10} \epsilon_k(n).$$

To accept the hypothesis for the consistency at 95% confidence level, the confidence interval was computed from [72]:

$$I \approx [\frac{1}{2NK}(\sqrt{(2NK-1)}-1.96)^2, \frac{1}{2NK}(\sqrt{(2NK-1)}+1.96)^2].$$

In this experiment, the confidence interval is shown by the dotted lines in Figure 6.11. The desired result is that the NIS statistic lie inside those confidence intervals more than 95% of the time. As can be seen from Figure 6.11, the CKF provides a reliable detection result. The CKF result indicates that the test signal belongs to the trained model with 95% confidence approximately.

Summary

In this chapter, the formulation of the square-root cubature Kalman filter (SCKF) is successfully validated through three different filtering problems. In all these problems, the SCKF significantly outperforms other presently known nonlinear filters.