Invited Paper

Practical methods of tracking of non-stationary time series applied to real world data

I T Nabney, A McLachlan and D Lowe
Neural Computing Research Group
Aston University
Birmingham B4 7ET
UK
i.t.nabney@aston.ac.uk

ABSTRACT

In this paper, we discuss some practical implications for implementing adaptable network algorithms applied to non-stationary time series problems. Two real world data sets, containing electricity load demands and foreign exchange market prices, are used to test several different methods, ranging from linear models with fixed parameters, to non-linear models which adapt both parameters and model order on-line.

Training with the extended Kalman filter, we demonstrate that the dynamic model-order increment procedure of the resource allocating RBF network (RAN) is highly sensitive to the parameters of the novelty criterion. We investigate the use of system noise for increasing the plasticity of the Kalman filter training algorithm, and discuss the consequences for on-line model order selection.

The results of our experiments show that there are advantages to be gained in tracking real world non-stationary data through the use of more complex adaptive models.

Keywords: non-stationary time series, on-line learning, extended Kalman filter

1. INTRODUCTION

A basic assumption underlying most current neural network theory is that the data are generated by a stationary process, i.e. that the distribution from which the data are drawn is time invariant. (This is true even for 'dynamic' time series problems.[1]) The assumption of stationarity can be acceptable in many circumstances, particularly if the time evolution permits quasi–stationary 'windowing' of the data (in which case the network can be retrained once its performance degrades). There are many cases, however, where such an approach will not give an acceptable level of performance, and it is this which motivates the investigation of alternative techniques.

In this paper we are concerned with problem domains in which batch and recursive training processes ('recursive' implies we are allowed to cycle repeatedly through the training data) are not viable options. One reason is due to potentially very large data sets, but the main reason is because the characteristics of different portions of data evolve in time (such as periodicity, seasonality or multiple trending behaviours on different time scales).

Earlier work has used adaptive algorithms (such as the extended Kalman filter, which adjusts network weights on-line, and Resource-allocating Radial Basis Function (RBF) networks, which enlarge the network structure on-line) on non-stationary data. These, and related methods, have been tested on synthetic data. Here we present a comparison between these techniques and simpler models, both RBF and linear, on two non-stationary real world data sets.

 0-8194-2141-3/96/$6.00

2. SEQUENTIAL OPTIMISATION AND THE EXTENDED KALMAN FILTER

This paper is concerned with the sequential estimation of nonlinear stochastic systems, primarily by a probabilistic description. For example, the class of stochastic approximation methods would be appropriate. One of the authors[2] has previously considered a stochastic approximation method for the unsupervised updating of the positions and smoothing factors of the first layer of an RBF using effectively a sequential EM algorithm. However for non-stationary problems we have to be prepared to trade off 'plasticity' for 'convergence'. This is because one criterion for the stochastic approximation methods to converge asymptotically is that the effective 'learning rate' should decrease to zero. However this is in conflict with the requirement of tracking and adapting to novel data. We cannot have both asymptotic consistency and adaptability.

Our problem may be expressed in a Bayesian framework as follows: given an estimate of the network parameters (weights *and* centres[1]) $\hat{\boldsymbol{w}}_{t-1}$ at instant $t-1$, based on the entire data history Y_{t-1} up to this instant, we wish to obtain a new estimate $\hat{\boldsymbol{w}}_t$ based on the previous estimate and the new information $\boldsymbol{y}_t$. In the general case we are concerned with the *distribution* of possible weight values, i.e. the posterior

$$p(\boldsymbol{w}_t|Y_t) = \frac{p(\boldsymbol{y}_t|\boldsymbol{w}_t)p(\boldsymbol{w}_t|Y_{t-1})}{p(\boldsymbol{y}_t|Y_{t-1})} \tag{1}$$

where $p(\boldsymbol{w}_t|Y_{t-1})$ represents the prior of the parameters given the measurements Y_{t-1}, and the likelihood $p(\boldsymbol{y}_t|\boldsymbol{w}_t)$ is determined from the noise density. Once we have calculated the posterior density, any estimate of the parameter vector may be obtained, at least in principle. For example the conditional mean or the maximum *a posteriori* estimate, $\hat{\boldsymbol{w}}_t$, of the weights may be calculated from $p(\boldsymbol{w}_t|Y_t)$. It is not generally possible to obtain analytic solutions for the Bayesian recursive relation (1), or even extract computationally tractable algorithms except in special circumstances. For instance, under assumptions of a linear model

$$\boldsymbol{f}_t(\boldsymbol{w}_t) = C_t\boldsymbol{w}_t \tag{2}$$

and Gaussian processes

$$\begin{aligned} p(\boldsymbol{w}_t|Y_t) &= N(\hat{\boldsymbol{w}}_t, P_t) \\ p(\boldsymbol{y}_t|\boldsymbol{w}_t) &= N(\boldsymbol{f}_t(\boldsymbol{w}_t), R_t) \\ p(\boldsymbol{w}_t|Y_{t-1}) &= N(\hat{\boldsymbol{w}}_{t-1}, P_{t-1}) \\ p(\boldsymbol{y}_t|Y_{t-1}) &= N(\boldsymbol{f}_t(\hat{\boldsymbol{w}}_{t-1}), S_t) \end{aligned} \tag{3}$$

where $\hat{\boldsymbol{w}}$ denotes the MAP estimate of the true state $\boldsymbol{w}$, the Kalman filter is obtained as a point estimate of the distribution in equation (1), i.e.

$$\begin{aligned} e_t &= \boldsymbol{y}_t - \boldsymbol{f}_t(\hat{\boldsymbol{w}}_{t-1}) \qquad (prediction\ \ error) \\ S_t &= R_t + C_t P_{t-1} C_t^T \end{aligned}$$

[1] We use fixed width Gaussians for reasons of numerical stability. Width parameters are, strictly speaking, hyperparameters, and should be optimised at a higher level of Bayesian inference.

$$
\begin{aligned}
K_t &= P_{t-1}C_t^T S_t^{-1} \qquad (Kalman\ \ gain) \\
\hat{\boldsymbol{w}}_t &= \hat{\boldsymbol{w}}_{t-1} + K_t e_t \\
P_t &= (I - K_t C_t) P_{t-1}
\end{aligned} \tag{4}
$$

Here R_t is a user-supplied noise level. For nonlinear systems (such as neural networks), the likelihood function is no longer a Gaussian in the weights, so we cannot perform the analysis and therefore have to make approximations. The first order extended Kalman filter is an estimate of the parameter vector based upon a lincarisation of the model around the current state estimate

$$
\boldsymbol{f}_t(\boldsymbol{w}_t) = \boldsymbol{f}_t(\hat{\boldsymbol{w}}_{t-1}) + \nabla_w \boldsymbol{f}_t(\hat{\boldsymbol{w}}_{t-1})(\boldsymbol{w}_t - \hat{\boldsymbol{w}}_{t-1}) \tag{5}
$$

and this basically represents the state of the art in sequential nonlinear modelling.

3. KALMAN FILTERS FOR NON-STATIONARY DATA

The fact that the entire data history is contributing via the recursively updated prior in (4) can inhibit the network from responding to evolution of the data generator. One simple method of alleviating this problem involves the addition of a state evolution equation (representing a change in state between observations) which merely adds a little noise (system noise) to the existing state estimate. If we denote the weight estimate at time t given the information gained up to time $t-1$ by $\boldsymbol{w}_{t|t-1}$, then the previously trivial evolution equation becomes

$$
\boldsymbol{w}_{t|t-1} = \boldsymbol{w}_{t-1|t-1} + \mu \tag{6}
$$

If the system noise distribution is N(0,Q) then the prior covariance evolution becomes

$$
P_{t|t-1} = P_{t-1|t-1} + Q \tag{7}
$$

Letting $Q = qI$, for small q, this has the effect of widening the Gaussian prior centred on the old weights, thus allowing the likelihood a greater influence and helping inhibit convergence of the filter. (This also has the drawback of making the algorithm more sensitive to outliers, but this will always be a problem in sequential learning.)

4. RESOURCE-ALLOCATING GAUSSIAN RBF NETWORKS

The Resource-allocating RBF (RAN) was introduced by Platt[3] and modified by Kadirkamanathan and Niranjan.[4] Platt's RAN is a Gaussian RBF which processes information sequentially, adapting the weights using the LMS algorithm unless novelty is detected in the data (or alternatively, the network's prediction is unacceptably inaccurate), in which case a new basis function is added. The conditions which determine whether the model order should be increased are that the prediction error and the distance from the datum to the nearest existing centre (d) should both exceed some user-specified critical values, i.e.

$$
\begin{aligned}
e_t &> e^c \\
d_{nearest\ unit} &> d_t^c
\end{aligned} \tag{8}
$$

The critical distance is allowed to decrease as data arrive, with the intention of inhibiting the reckless addition of units early in the training without restricting unit addition at a later stage.

$$d_i^c = max(\gamma^t d_{max}^c, d_{min}^c) \quad , \quad \gamma \in (0,1) \tag{9}$$

When both criteria are satisfied, then the new basis function is centred on the input datum and weighted to fit the target exactly. (Note that this may not be a desirable procedure if there is a significant noise component in the data.)

Kadirkamanathan and Niranjan provide a geometric perspective (F-projections) which, when simplified in the interests of computational tractability, reduces essentially to Platt's RAN with the LMS algorithm being replaced with the EKF.[5] When adding basis functions in the EKF procedure, the prior covariance P is augmented with a multiple of the identity matrix - again a somewhat arbitrary prescription. It was argued that this modification resulted in the construction of more compact networks, and this was illustrated in various examples.

While these papers concerned themselves with applications to stationary environments, in which case the entire data set can be cycled through if desired, the EKF–RAN is potentially of use in the analysis of non-stationary data (albeit with no guarantee of convergence). It should be borne in mind, however, that there are several deficiencies with this approach:–

- model order adaptation relies upon user supplied thresholds with no a priori indication of suitable values,
- there is no mechanism for reducing model order even though excessive model order can have a detrimental effect on filter performance,
- the linearised approximation in the extended Kalman filter used in the calculation of the filter gain may not be appropriate for the high degrees of nonlinearity in neural networks,
- the filter relies upon an initial estimate of the covariance P which, even if valid initially, may become inaccurate in non-stationary environments,
- there is no principled way to initialise the new covariance matrix entries when the model order is increased in a neural network,
- the weight initialisation prescription is designed to work with Gaussian basis functions and, as it fits the data exactly, is not robust against outliers.

5. PREVIOUS INVESTIGATIONS

In a previous paper,[6] we investigated the performance of some RAN variations on an artificial time series generated by the quadratic map. As the calculation of the gain in the extended Kalman filter relies on a linearisation of the network function, it is a poor approximation given that some quadratic terms in the log likelihood are being discarded, along with higher order terms. For this reason, we studied the effects of replacing the extended Kalman filter by algorithms which better approximated the non-linearity of the network. For example, by using the Hessian of the network, we can expand the network output (as a function of the weights) to second order about the current value, thus giving the *second order* extended Kalman filter.[7] Alternatively, we can still linearise the model, but make use of the *(recursively) iterated EKF*. The iterated EKF re-linearises the model about each new weight vector prediction *using the same data point and the same prior*, iterating until convergence is achieved. If the iteration converges, then this will give a more accurate approximation to the maximum of the posterior than the EKF, but it cannot circumvent the limitations in performance due to its

failure to capture significant nonlinear effects. Ideally, the posterior should be maximised directly using some conventional optimisation scheme. Once an optimum has been found, the (Gaussian) prior for the next datum can be constructed by calculating the Hessian of the log posterior. This is generally far more computationally expensive, but this should not be a problem unless the data are arriving at too high a frequency. We therefore also included the BFGS quasi-Newton algorithm in our investigation.

It was found that, for fixed size networks, the BFGS could consistently outperform the EKF and the iterated EKF in stationary situations, but in the non-stationary cases, the EKF seemed better able to cope in regions where the series was undergoing significant changes in structure. This was conjectured to be due to the loss of information in the EKF (due to linearisation of the network) making it more adaptable to a changing environment – the increased accuracy of other optimisation techniques can result in an undesirably restrictive prior. (The second order EKF was found to be unreliable - if the current weight estimate is too far from the true optimum for that time-step, then the Hessian is not guaranteed to be positive definite, and thus the filter fails.)

In the RAN tests, it was found that the model order was highly sensitive to the particular values of the many user-supplied parameters, and it was demonstrated that, merely by making small changes to the critical distances, large networks could be constructed whose predictive abilities were, on average, no better than fixed size networks with many fewer basis functions. It was also found that the flexibility of the basic EKF algorithm more often than not led to the construction of more compact networks than the other algorithms considered.

For that reason, we shall only be considering the EKF algorithm in this paper.

6. METHODOLOGY

In this paper we aim to compare the performance of the EKF and Resource-allocating RBF networks with simpler techniques in modelling non-stationary time series. To do this, we use two real world data sets with clear non-stationary characteristics. These are electricity load demand and the spot foreign exchange market for Deutsche Mark against French Franc.

In both cases, the last part of the data set was used for testing the models one-step ahead prediction. Two assessment techniques were used: normalised RMSE and graphs of the error time series. The normalised RMSE was calculated as:

$$\text{RMSE} = \sqrt{\frac{\sum_p (o_p - t_p)^2}{\sum_p (t_p - \bar{t})^2}} \tag{10}$$

where p is an index that runs over all the patterns in the test set, o_p and t_p are respectively the model output and target value for pattern p, and $\bar{t}$ is the mean value of the target values on the test set. This expression has the value 0 for a perfect match between model and target, and the value 1 if the model just outputs the target mean $\bar{t}$. This measure should be interpreted with caution because it is sensitive to outliers and does not detect whether errors are correlated. For a good model, the errors should be zero mean and uncorrelated. Visual inspection of the error graphs can tell us a lot about how close to this ideal a given model is. We are particularly interested in the performance in regions of non-stationarity.

Both fixed (i.e. models whose parameters are fixed after recursive training on the 'training data' subset of the data) and adaptive (i.e. models whose parameters are adapted continuously) models were fitted to the data. In line with best practice, both linear and non-linear models were used.

The fixed models used were:

- the optimal linear model

- a fixed size RBF network with $z \log z$ activation functions (such activation functions have certain advantages over the more usual Gaussians[8]).

The adaptive models were:

- Fixed RBF with adaptive bias.[6] All the weights of the model are set on the training set, with the exception of the bias weights to the output nodes which are adjusted on-line on the test set.
- Optimal linear with adaptive bias. In this model, the bias (or offset) term is modified during testing.
- Fixed size RBF with EKF on-line training. The weights in this model are continually modified as described in section 3..
- RAN with EKF on-line training. The number of hidden units in the network can be increased at any time, and the weights are continually modified.

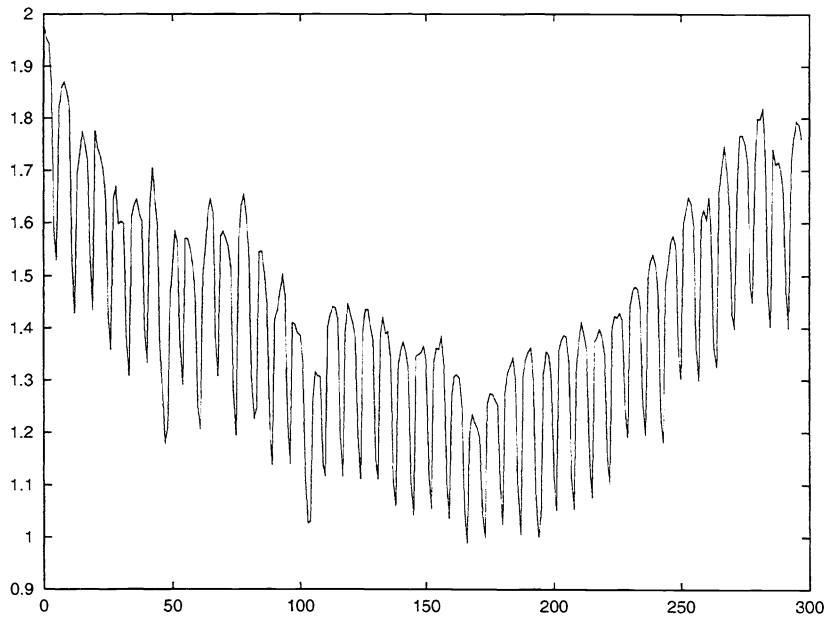

Figure 1: *Electricity Load Demand Data.*

7. ELECTRICITY LOAD FORECASTING

This data set contains 300 samples representing averaged daily electricity load demand (Figure 1). The data exhibits the overlay of two seasonality effects: a reasonably regular weekly pattern (with lower consumption at weekends) has a year-long pattern of seasonality superimposed. This yearly pattern shows up as a drop in the mean demand level in the section corresponding to the warmer part of the year.

The models were trained on the first 93 patterns, and tested on the remaining 198 at the end of the time series. A window of seven previous values was used as the input. Figure 2 shows the prediction error (innovations) sequence obtained from models with fixed parameters and with an adaptive bias. (By comparing with Figure 1, the large prediction error spikes can be seen to be associated with outliers in the load data. Such irregular features are most likely due to unusual weather conditions or public holidays, leading to atypical power requirements.)

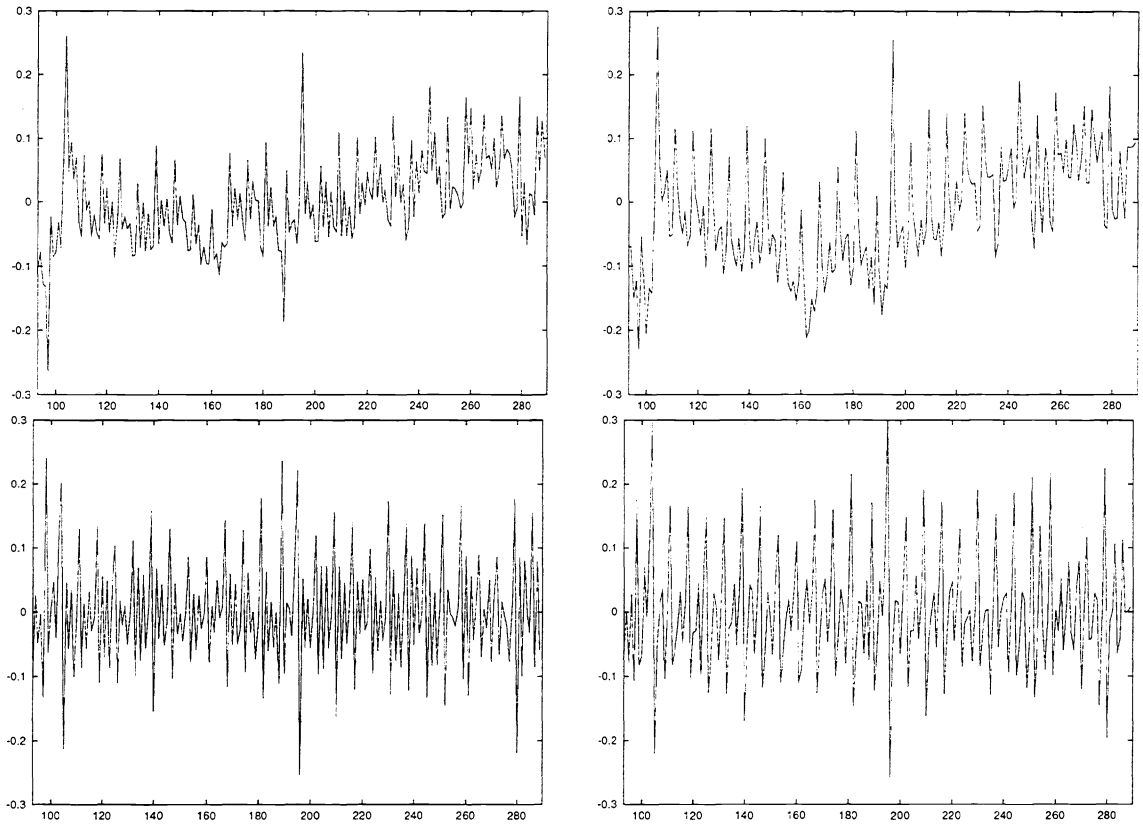

Figure 2: *Prediction errors for fixed parameter and adaptive bias models on electricity load test data. The four cases are: optimal linear model (top left), fixed RBF with 18 uniform centres (top right), optimal linear with adaptive bias (bottom left), RBF with adaptive bias and 18 uniform centres (bottom right).*

The errors of the fixed models are clearly non-stationary in the mean, in that the average level of the graph first falls below 0 and then rises above 0. It can be seen that the adaptive bias reduces the prediction errors in the central region (between time steps 120 to 180), which is where the network requires the greatest flexibility. The effect is not so significant towards the end of the sequence, as the trend is back towards a region of data space which has already been modelled. Although the error sequence has zero mean, closer inspection of the graphs reveals that it is not uncorrelated: there is a definite weekly pattern (repeating every 7 steps). It is also noticeable that the adaptive bias approach tends to over-compensate for large errors. An error of large magnitude is often followed by an error of slightly smaller magnitude but opposite sign.

Figure 3 shows the prediction errors for 10 hidden unit RBF networks trained with EKF and RANs starting with 1 basis function. It can be seen that the errors are reduced considerably in comparison with those shown in Figure 2. This shows that adapting all the model parameters definitely improves the performance on this data set. It is plausible that a linear model trained with EKF would also perform well, although we have not tested this. System noise played an important role in increasing the adaptability (or 'plasticity') of the model. When this parameter was reduced to a lower value, the errors in the central region increased in magnitude.

System noise was also useful in the RAN framework, where it restricted unnecessary growth. With no noise, 34 hidden units were added, whereas with noise 23 units were added. A drawback of RANs is that they are

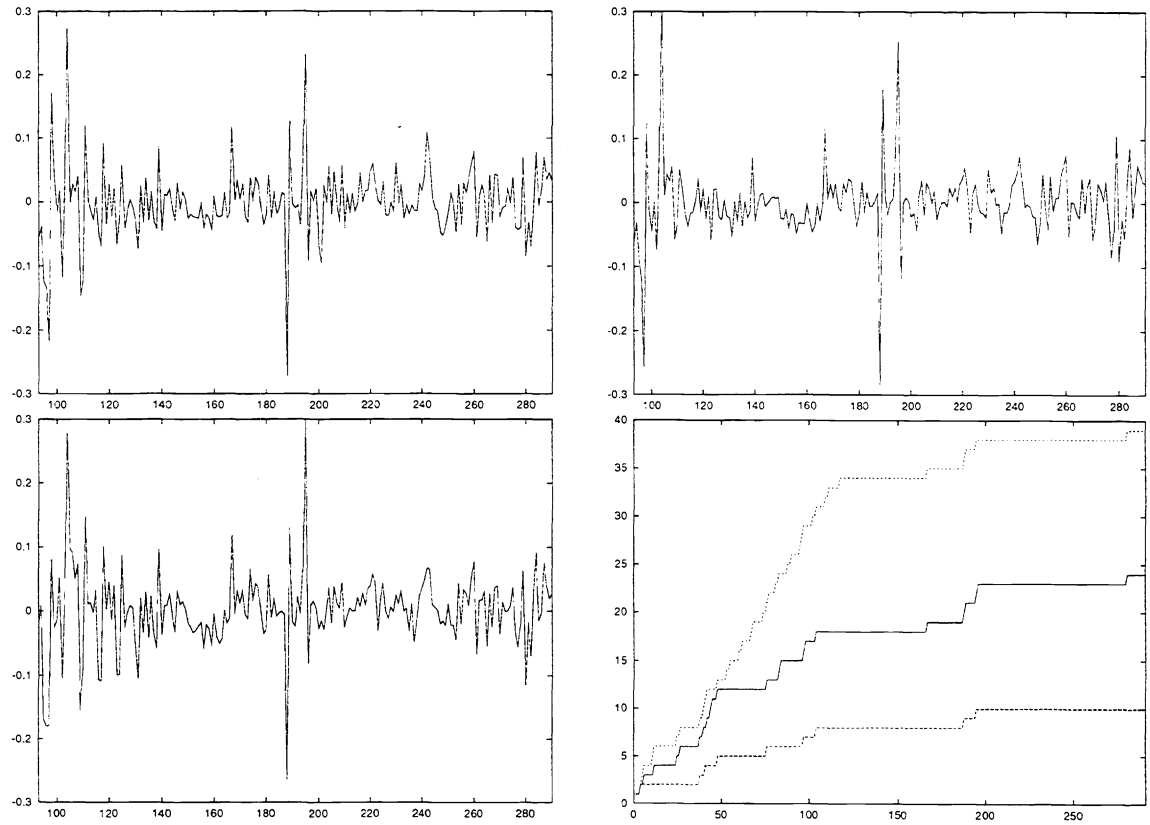

Figure 3: *Prediction errors for EKF networks. The three cases are: fixed size (10 basis functions) (top left), RAN parameter set 2 (top right), RAN parameter set 3 (bottom left). The growth of three different RAN networks is also shown (lower dotted line from parameter set 2, middle solid line from parameter set 1, upper dotted line from parameter set 3) (bottom right).*

extremely sensitive to changes in the novelty criterion. This is illustrated in the last graph of Figure 3. Parameter set 1 is represented by the middle, solid, line, and led a final network (at the end of the test set) with 24 hidden units. By doubling the critical error value e^c in parameter set 2, the final number of units was reduced to 10. Halving the size of maximum critical distance d^c_{max}, as in parameter set 3, lead to a network with 39 units. Despite these significant variations in network size, the effect of these changes on the prediction error sequence is near negligible.

Table 1 confirms these comments.

8. FOREIGN EXCHANGE SPOT RATE FORECASTING

The second data set that we consider in this paper comes from the foreign exchange spot markets. Over the past few years the currencies of some countries in the European Union have been members of the Exchange Rate Mechanism (ERM). The purpose of the ERM is to attempt to control exchange rates between the member currencies so that they all lie within certain tolerances of the 'ideal' rate. Although the market is free, the central

Model Type	Norm. RMSE	Model Type	Norm. RMSE
Fixed Optimal Linear	0.3596	EKF fixed size RBF	0.2850
Fixed RBF	0.4631	RAN set 1	0.2856
Adaptive bias Opt. Lin.	0.4361	RAN set 2	0.2769
Adaptive bias RBF	0.4726	RAN set 3	0.3013

Table 1: Results on electricity load demand data.

banks of these countries intervene when necessary to maintain the status quo.

For some currencies in the ERM, there have been non-stationary effects on some occasions due to efforts by traders to make profits by forcing a central bank to intervene in a certain market through the sheer volume of selling. One such event took place in early August 1993 in the Deutsche Mark/French Franc market when Francs were sold heavily. The data in Figure 4 shows 700 daily prices (at close of trading) for this market from 30 September 1991 to 6 June 1994. The first 400 data points (from 30 September 1991 to 11 April 1993) were used for training and the last 300 (from 12 April 1993 to 6 June 1994) were used for testing. The large jump at time step 480 occurred when the French and German central banks were forced to admit defeat and allowed the Franc to slip against the Mark. In later trading, a return was made to a level closer to the pre-crisis exchange rate.

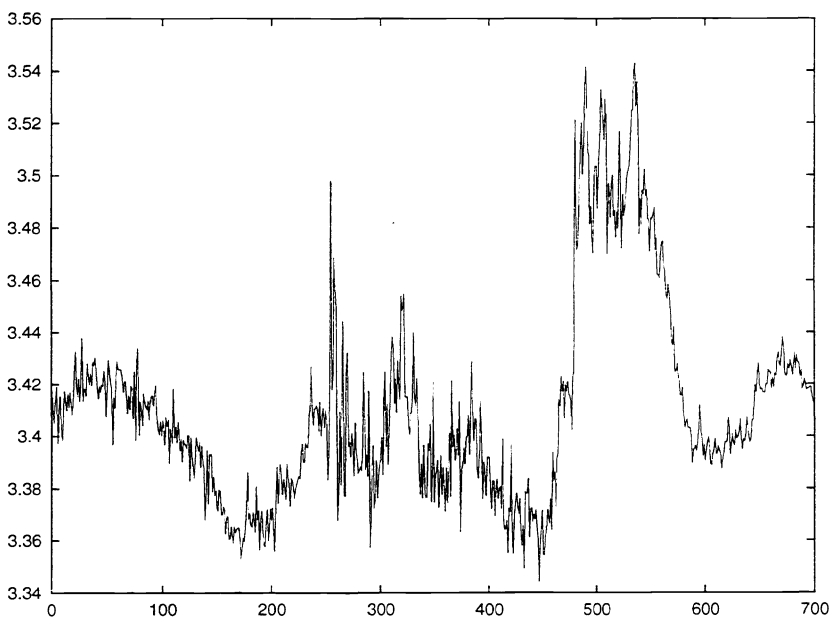

Figure 4: *Deutsche Mark/French Franc daily closing prices.*

Figure 4 shows no obvious seasonal effects in the data, but more detailed studies of high frequency market data have demonstrated some subtle 'day of week' effects. For this reason, a window of 5 previous values was used as input to the EKF and RAN models. A window of 1 value was used in the fixed and adpative bias models.

Figure 5 shows the test set results achieved with the non-linear models. The first graph shows how a fixed parameter model is unable to cope with the ERM crisis period, with errors that are consistently large and of positive sign. All three adaptive models cope much better, and there is little obvious indication of non-stationarity in the error sequence. The experiments confirmed that system noise was effective in improving the plasticity of

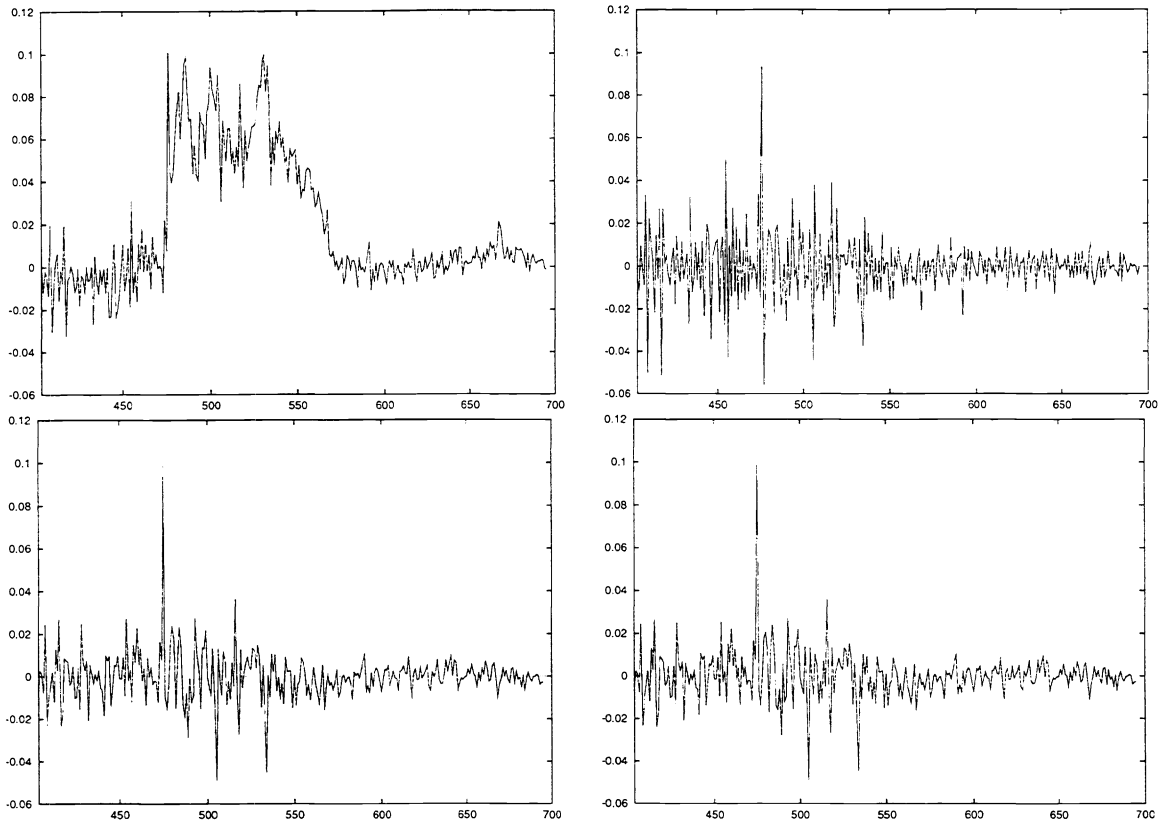

Figure 5: *Prediction errors for non-linear models on the foreign exchange test set. The four cases are: fixed RBF with 15 uniform centres (top left), fixed size RBF with adaptive bias and 15 uniform centres (top right), fixed size RBF trained with EKF and 1 hidden unit (bottom left), RAN (bottom right).*

the RBF models.

As in the electricity load demand experiments, the RAN framework was very sensitive to the novelty criteria values. The network for which the graph is plotted in Figure 5 was generated by parameter set 1 and had 30 basis functions at the end of the test set. When the lower bound on the critical distance d^c_{min} was halved (so that more basis functions would be added during the test set) in parameter set 2, the number of basis functions in the final network was 112. On the other hand, doubling the critical error value e^c in parameter set 3 gave a network with 4 basis functions. The first and last of these networks had very similar error sequences and their normalised RMSE was the same to within 0.5%.

One feature of all the models was that the predictions were close to a one-step lag of the actual time series. This feature of predicting financial markets has often been noted in the past. Because of this, it is always valuable to compare results with those of the naive model that simply predicts the current value for the next time step. The error figures for all these models are contained in table 2.

Model Type	Norm. RMSE	Model Type	Norm. RMSE
Naive predictor	0.2438	EKF fixed size RBF	0.2351
Fixed RBF	0.7224	RAN set 1	0.2336
Adaptive bias RBF	0.2996	RAN set 3	0.2345

Table 2: Results on foreign exchange data.

9. DISCUSSION OF RESULTS AND FUTURE RESEARCH

The main features found in this investigation are :

- With care, RBF networks trained using the Extended Kalman Filter seem to outperform simpler methods of adapting to non-stationary data. For definitive statements to be made further experimentation on real world data is required.
- The introduction of system noise is an effective method of improving the plasticity of the RBF models.
- Linear models often perform well, particularly with adaptive bias.
- RAN networks did not outperform fixed size nets trained with EKF.
- Previous observations of RAN sensitivity to parameter settings are confirmed on real world data.
- System noise leads to the construction of more compact RAN networks when using the original novelty criterion.
- The non-stationarity inhibits the convergence of the prior covariance P and hence the "error bar" matrix S. System noise also serves to increase P and S.

While the RAN procedure is indeed capable of constructing a compact network with adequate modelling accuracy , it requires a considerable time investment to determine an appropriate set of parameters for any given application. This is in addition to a certain amount of tuning required for the EKF training of a fixed size network. While certain modifications of the original prescription, by way of improving the training algorithm and novelty criterion, can lead to significant improvements when modelling stationary data, they do not, in general, lead to any consistent improvements in the non-stationary case.[2] Applications in non-stationary environments therefore require considerable trial-and-error experimentation in order to obtain an acceptable level of performance.

It is clear, therefore, that more sophisticated measures of non-stationarity will be required to construct a more appropriate novelty criterion if RAN networks are to be a viable tool in the tracking of non-stationary series.

10. ACKNOWLEDGEMENTS

This work was supported in part by EPSRC grants number GR/K51815 and GR/J75425. We are grateful to Midland Electricity plc and Pareto Partners for making available the electricity load demand and foreign exchange data respectively.

[2] Although in order for definitive statements to be made, the above experiment will need to be repeated on a wider range of data sets.

11. REFERENCES

[1] D Lowe and A R Webb. Time series prediction by adaptive networks: A dynamical systems perspective. *IEE Proceedings–F*, 138:17–24, 1991.

[2] David Lowe. On the iterative inversion of RBF networks: A statistical interpretation. In *Second IEE International Conference on Artificial Neural Networks*, 1991.

[3] John C Platt. A resource allocating network for function interpolation. *Neural Computation*, 3:213–225, 1991.

[4] Visakan Kadirkamanathan and Mahesan Niranjan. A function estimation approach to sequential learning with neural networks. *Neural Computation*, 5:954–975, 1993.

[5] John V Candy. *Signal Processing : The Model-Based Approach.* McGraw-Hill, 1986.

[6] David Lowe and Alan McLachlan. Modelling of nonstationary processes using radial basis function networks. In *Fourth IEE International Conference on Artificial Neural Networks*, 1995.

[7] Andrew H Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, 1970.

[8] David Lowe. On the use of non-local and non-positive definite basis functions in radial basis function networks. In *Fourth IEE International Conference on Artificial Neural Networks*, 1995.