World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:10, 2008

Neuro-Fuzzy Network based on Extended Kalman Filtering for Financial Time Series

Chokri Slim

***Abstract*—**The neural network's performance can be measured by efficiency and accuracy. The major disadvantages of neural network approach are that the generalization capability of neural networks is often significantly low, and it may take a very long time to tune the weights in the net to generate an accurate model for a highly complex and nonlinear systems. This paper presents a novel Neuro-fuzzy architecture based on Extended Kalman filter. To test the performance and applicability of the proposed neuro-fuzzy model, simulation study of nonlinear complex dynamic system is carried out. The proposed method can be applied to an on-line incremental adaptive learning for the prediction of financial time series. A benchmark case studie is used to demonstrate that the proposed model is a superior neuro-fuzzy modeling technique.

***Keywords*—**Neuro-fuzzy, Extended Kalman filter, nonlinear systems, financial time series.

I. INTRODUCTION

FUZZY rule-based systems and artificial neural networks originated from different philosophies and were originally considered independent of each other. Later studies revealed that they actually have a close relation. Buckley et al. [1] discussed the functional equivalence between neural networks and fuzzy expert systems. The integration of fuzzy logic and neural networks has given birth to an emerging technology field, fuzzy neural networks. The theory of fuzzy logic provides a mathematical strength to capture the uncertainties associated with human cognitive processes, such as thinking and reasoning, also it provides a mathematical morphology to emulate certain perceptual and linguistic attributes associated with human cognition. While fuzzy theory provides an inference mechanism under cognitive uncertainty, computational neural networks offer exciting advantages such as learning, adaptation, fault-tolerance, parallelism and generalization. The computational neural networks are capable of coping with computational complexity, nonlinearity and uncertainty. It is interesting to note that fuzzy logic is another powerful tool for modeling uncertainties associated with human cognition, thinking and perception [2,3].

Many authors have proposed various neuro-fuzzy models as well as complex training algorithms. Of these, Jang [4] proposed the famous neuro-fuzzy model ANFIS (adaptive network-based fuzzy inference system), which has been successfully applied in various fields. In ANFIS, the hybrid learning algorithm is adopted that integrates the BP successfully applied in various fields. In ANFIS, the hybrid learning algorithm is adopted that integrates the BP (Backward Propagation) algorithm with the recursive least squares algorithm to adjust parameters. ANFIS was later extended to the coactive ANFIS in [5] and to the generalized ANFIS in [6]. Horikawa et al. [7] proposed a neuro-fuzzy model using sigmoid functions to generate the bell-shape input membership functions and trained it with the BP algorithm. However, some practical difficulties associated with gradient descent are slow convergence and ineffectiveness at finding a good solution [8].

Kalman filtering estimation is a topic that has received very little attention in the field of fuzzy neural networks There have been a few papers published recently on fuzzy observer design; however, these papers usually deal with the noise-free case. That is, fuzzy observers are designed for systems that are not affected by noise. In our paper entitled "Neural Network for Modeling Nonlinear Time Series: A New Approach", [2], we have developed a Neural network based on Extended Kalman Filter. In this paper we extended this work to derive a neuro- fuzzy network based on Extended Kalman filter (EFK) to predict and estimate state of non linear dynamic systems. We demonstrate its performance, and compare it with ANFIS system and classical neural networks using gradient descent on some non linear dynamic systems.

This paper is organized as follows. In Section 2, we briefly review some fundamental notions of fuzzy system and the neuro-fuzzy models. In Section 3, we first precisely formulate the proposed model, and then fuzzify the system model. The EKF algorithm is then derived in this section. In Section 4, computer simulation is shown to compare the new model with classical neural network trained by the BP algorithm and the ANFIS scheme. In section 5, The proposed model can be applied to an on-line incremental adaptive learning for the prediction of financial time series. Finally, section 5 contains some concluding remarks and suggestions for further research.

II. BACKGROUND

A. Fuzzy Logic

Since its introduction in 1965 by Zadeh [9], fuzzy set theory has found applications in a wide variety of disciplines. Modeling and control of dynamic systems belong to the fields in which fuzzy set techniques have received considerable attention, not only from the scientific community but also from industry.

The purpose of fuzzy logic is to map one space (input) to

Chokri Slim is with Bestmod Laboratory, Higher Institute of Management, University of Tunis, Tunisia (phone: 00216-827298; fax: 00216-602404; e-mail: chokri.slim@iscae.rnu.tn).

another (output) with relative precision (normally through if-then rules). It is a better tool for simulating human thinking and allows the computer to understand and compute like a human. There are several advantages of fuzzy logic.

• It is easy to understand.
• It has the tolerance of imprecise data.
• It can bring human knowledge to the system directly.
• It can be integrated with other systems smoothly, for example, Neural Network and Control System.
• It has stronger power when solving difficult nonlinear problems.
• It has great flexibility.

The implementation of fuzzy logic has four steps:

1. Fuzzification: Mapping the input to the degree of membership using membership function. Define the surfaces (fuzzification): Each variable is decomposed into a set of fuzzy regions (or states) called the "fuzzy sets". These fuzzy sets are assigned certain names from the set N that span the variables domains. They do not have crisp, clearly defined boundaries. In the end each fuzzy set is represented by its membership function. A membership function is a curve which defines how the points in the input space (elements of N) are mapped to a membership value (or degree of membership), a real number between 0 and 1. Mathematically, a fuzzy set *f* is defined by a set of ordered pairs.

$$f = \{(x, \mu\ (x))/x \varepsilon N, \mu\ (x) \varepsilon\, 0,1]\} \quad (1)$$

where $\mu\ (x)$ denotes a membership function.

Membership functions can be of various types: triangular, trapezoidal, Gaussian, sigmoidal, polynomial, etc.

2. If the antecedent includes more than one part, the fuzzy operators are applied to them to get the single number: To define a link between the input and output variables a rule base is created. Linguistic rules are of the form:

IF <x is A> THEN <y is B>

where *x* and *y* are scalar variables and *A* and *B* are linguistic values defined by fuzzy sets. The phrase " *x is A*" is called the antecedent or premise, while "*y is B*" is called the consequent or conclusion. Fuzzy rules form a fuzzy rule base. The number of rules varies with the number of the variables. The idea is to try to identify all the possible combinations of inputs. Thus, if there are three input variables, each described by five fuzzy sets, the required number of rules would be 5^3 =125. Fuzzified inputs cause some rules to be activated and to contribute to an overall output which is calculated using the so called " Mamdani inference" [10]. Mamdani inference applies min and max operators for fuzzy AND (intersection) and OR (union) operators. It also requires that the output membership function is a fuzzy set (unlike " Sugeno inference" where the output is either constant or linear).

3. To get the membership function of output using the result of step 2 (or step 1 if the antecedent only includes one part). The implementation method in this step can be min function or prod function.

4. Defuzzifying. The result of step 4 is a membership function to the whole range of output and it has to be defuzzified to get a specific number. The common method used to defuzzify is centroid calculation.

B. Neuro-Fuzzy

Since both the fuzzy logic and Neural Network can simulate the thinking of a human being, it is intuitive to combine them in order to take advantage of both of their strengths. There are three different approaches for combining neural networks and fuzzy systems :

• Concurrent Neural-Fuzzy Models: In this model, the fuzzy system and Neural. Network are used concurrently to the same task. The fuzzy system is used either before or after the processing of Neural Network. They are not related tightly. The Neural Network does not change any parameters in the fuzzy system.

• Cooperative Neuro-Fuzzy Models : In this model a neural network or just a simple neural learning algorithm is used to learn certain parameters of fuzzy sets, fuzzy rules, or weights of the fuzzy system. After learning the Neural Network it no longer exists. The result is a pure fuzzy system.

• Hybrid Neuro-Fuzzy Inference System: This is new model which can be interpreted either as a Neural Network or as a fuzzy system. The Neural Network and fuzzy system are no longer separated. Modern Neuro-Fuzzy approaches come from this type.

The research in this study uses a hybrid Neuro-Fuzzy model. It uses the type of fuzzy rules with certainty factors, in which a two phase learning scheme is developed. In neuro-fuzzy models, two major types of learning are required: structure learning algorithms to find appropriate FL rules; and parameter learning algorithms to fine-tune the membership functions and other parameters. There are several ways that structure learning and parameter learning can be combined in a neuro-fuzzy system. They can be performed sequentially: structure learning is used first to find the appropriate structure of a neuro-fuzzy system; and parameter learning is then used to fine-tune the parameters. In some situations, only parameter learning or structure learning is necessary when structure (fuzzy rules) or parameters (membership functions) are provided by experts, and the structure in some neuro-fuzzy systems [11] is fixed. Identification of fuzzy rules has been one of the most important aspects in the design of Fuzzy Inference Sysyem. Identified rules and concise rules can provide an initial structure of networks so that learning processes can be fast, reliable and highly intuitive.

III. Description of the Methodologie

A. The Proposed Neuro-Fuzzy Model

The proposed neuro-fuzzy model is a multilayer neural network-based fuzzy system and the system has a total of five layers. In this connectionist structure, the input and output nodes represent the input states and output reponse, respectively, and in the hidden layers, there are nodes functioning as membership functions (MFs) and rules. This eliminates the disadvantage of a normal feedforward multi-layer network, which is difficult for an observer to understand or to modify.

Throughout the simulation examples presented in this paper, all the MFs used are bell-shaped (Gaussian) functions defined in (2):

$$\mu_A(x) = \exp-((x-c)^2/\sigma^2) \qquad (2)$$

A Gaussian membership function is determined by c and σ: c represents the centre of the MF; and σ determines the width of the MF. A detailed description of the components of the model's structure and functionalities, and the philosophy behind this architecture are given below.

1. Input Layer

Nodes in this layer are input nodes that represent input linguistic variables as crisp values. The nodes in this layer only transmit input values to the next layer, the membership function layer. Each node is connected to only those nodes of layer 2, which represent the linguistic values of corresponding linguistic variables.

2. Fuzzy Input Layer

Nodes in this layer act as membership functions to represent the terms of the respective linguistic variables. The input values are fed to fuzzy input layer that calculates the membership degrees. This is implemented using Gaussian membership functions with two parameters, mean (or centre, c) and variance (or width, σ).This layer implements fuzzification for the inputs. It represents fuzzy quantisation of input variables.

$$y_t^{(FI)} = \exp(-((x_t^{(I)} - c)^2/\sigma^2) \qquad (3)$$

3. Rule Nodes Layer

The third layer contains rule nodes that evolve through learning. Evolving means all nodes on the third layer are created during learning. The rule nodes represent prototypes of input-output data associations that can be graphically represented as associations of hyper-spheres from the fuzzy input and the fuzzy output space. Hence, the functions of the layer are

$$y_t^{(R)} = \min_{i\varepsilon I_t} y_i^{(F)} \qquad (4)$$

where I_t is the set of indices of the nodes in fuzzy layer that are connected to node t in Rule layer and $y_i^{(F)}$ is the output of node i in Fuzzy input layer.

4. Fuzzy Output Layer

The fourth layer is fuzzy output layer where each node represents fuzzy quantisation of the output variables. The activation of the node represents the degree to which this membership function is supported by all fuzzy rules together. The connection weights w_{kj} of the links connecting nodes k in fuzzy output layer to nodes j in rule nodes layer represent conceptually the CFs of the corresponding fuzzy rules when inferring fuzzy output values.

$$y_t^{(FO)} = \max_{i\varepsilon I_k}(y_i^{(R)} w_{ki}^{(F)}) \qquad (5)$$

where I_k is the set of indices of the nodes in Rule layer that are connected to the node k in Fuzzy output layer.

5. Output Layer

This represents the output variables of the system. These nodes and the links attached to them act as a defuzzifier. A node in this layer computes a crisp output signal. The output variable layer makes the defuzzification for fuzzy output variables.

The input--output relationship of the units in each layer are defined by the following equations:

$$y_t = \frac{\sum_{k\varepsilon It} w_{tk}^{(FO,R)} y_k^{(FO)} \sigma_{tk} c_{tk}}{\sum_{k} w_{tk}^{(FO,R)} y_k^{(FO)} \sigma_k} \qquad (6)$$

Where I_l is the set of indices of the nodes in Fuzzy output layer which are connected to the node l in output layer and c_{lk} and σ_{lk} are respectively, the centroid and width of the membership function of the output linguistic value represented by k in Fuzzy output layer.

B. The Extended Kalman Filter

Having identified the system model using the above Fuzzy neural network model, it is useful to recast this model in the state-space form to perform state estimation the non linear state space of the neuro-fuzzy network is given by:

$$\begin{aligned} \Theta_{t+1} &= \Theta_t \\ Y_t &= h(\Theta_t) \end{aligned} \qquad (7)$$

The state of the nonlinear system can then be represented as:

$$\Theta_t = \left[w_{11} \ldots w_{1k}, \sigma_{11}, \ldots, \sigma_{1k}, c_{11}, \ldots c_{1k}\right]$$

Where $h(\Theta_t)$ is the fuzzy system's nonlinear mapping between the membership function parameters and the output of the fuzzy system. In order to execute a stable Kalman filter algorithm, we need to add some artifcial process noise and measurement noise to the system model.

$$\begin{aligned} \Theta_{t+1} &= \Theta_t + \omega_t \\ Y_t &= h(\Theta_t) + \xi_t \end{aligned} \tag{8}$$

Equations given by (8) represents respectively the transition equation and the observation equation [13], with Θ_t is the state vector of the system at time t and Y_t is the observation vector at time t. the ω_t is the process noise or the vector of innovations, with zero mean and variance Q_t, ξ_t is additive measurement noise is withe zero mean and variance R_t. We assume that the noise vectors are uncorreleted with covariances:

$$P \begin{bmatrix} Q_t & 0 \\ 0 & R_t \end{bmatrix}$$

The Kalman filter [12] addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by a linear stochastic difference equation. But what happens if the process to be estimated and (or) the measurement relationship to the process is non-linear? Some of the most interesting and successful applications of Kalman filtering have been such situations. A Kalman filter that linearizes about the current mean and covariance is referred to as an extended Kalman filter or EKF [14]. In something akin to a Taylor series, we can linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even in the face of non-linear relationships.

The complete set of EKF equations is shown below.

$$\begin{aligned} \hat{\Theta}_{t+1/t} &= \hat{\Theta}_{t/t} \\ P^{\Theta}_{t+1/t} &= \mathbf{M}_t P^{\Theta}_{t/t} \mathbf{M}_t^{T} + Q_t \\ K_{t+1} &= P^{\Theta}_{t+1/t} G^{T}_{t+1} (G_{t+1} P^{\Theta}_{t+1/t} G^{T}_{t+1} + R_{t+1})^{-1} \\ \hat{\Theta}_{t+1/t+1} &= \hat{\Theta}_{t+1/t} + K_{t+1}(Y_{t+1} - h(\hat{\Theta}_{t+1/t})) \\ P^{\Theta}_{t+1/t+1} &= P^{\Theta}_{t+1/t}(I - K_{t+1}\mathbf{G}_{t+1}) \\ \hat{Y}_{t+1/t} &= h(\hat{\Theta}_{t+1/t}) \end{aligned} \tag{9}$$

$$\mathbf{M}_t = \frac{\partial H(\hat{\Theta}_{t/t})}{\partial \hat{\Theta}_{t/t}} ; \quad \mathbf{G}_{t+1} = \frac{\partial h(\hat{\Theta}_{t+1/t},)}{\partial \hat{\Theta}_{t+1/t}}$$

IV. Simulation Study

Time series prediction is a very important practical problem with a diverse range of applications from economic and business planning to signal processing and control. The time series used in this work was generated by the chaotic Mackey-Glass differential equation [13] defined by Eq. (10) below. This equation demonstrates chaotic behaviour when $s > 17$; and higher values of s yield higher dimensional chaos. In this work a value of $s = 18$ was employed and illustrates figure 1 the first 1000 points of this series using an initial condition of y_0 = 0.7 and the paremeters are : $a = 0.2, b = 0.9$ and $c = 10$.

$$y_t = by_{t-1} + a\frac{y_{t-s}}{1 + y^{c}_{t-s}}. \tag{10}$$

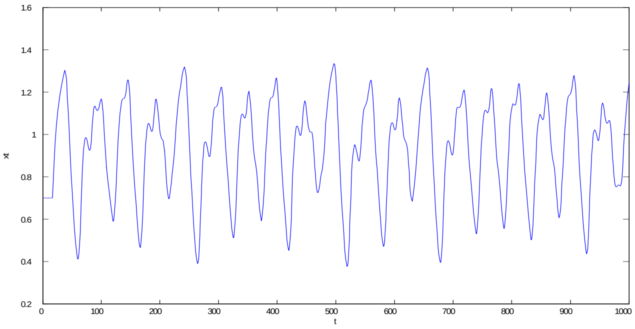

Fig. 1 The chaotic behaviour of the Mackey-Glass time series at s = 18

The prediction of future values of this series is a benchmark problem which has been considered by a number of researchers [2-4,7,9]. The problem can be formulated as given values y_{t-m} , $y_{t-m+1}, \ldots, y_{t-1}$; determine y_{t-1+n}, where m and n are fixed positive integers and t is the series index. In this work we choose $n = 1$ and $m = 6$.

The simulation for a 6-input single-output problem for single or multi-step prediction can be implemented on the fuzzy neural network illustrated in Figure 2 when each input domain is partitioned into 3 fuzzy sets. The number of distinct fuzzy sets in each input domain, resulting in 18 nodes (i.e. 6 inputs and 3 fuzzy sets per input). The initial values of the weights between the input and hidden layers was determined by choosing a partitioning strategy for the fuzzy

TABLE I
COMPARATIVE RESULTS FOR MACKEY--GLASS CHAOTIC TIME SERIES

Method	Training RMSE	Test RMSE
Classical NN	0.014	0.010
ANFIS	0.012	0.007
New model	0.010	0.006

sets across each input domain. As each input domain is identical the problem reduces to selecting fuzzy sets across the range of the time series points. The proposed fuzzy neural network was readily implemented on the MATLAB software trained with extended kalman filter discussed in section. To compare the performance of the fuzzy neural network similar sized classical neural networks were also implemented. A summary of the implementation results obtained are presented in Table I.

All the simulations used 700 points from the series as training data (from $t = 1$ to $t = 700$) and used a further 300 points as test data (from $t = 701$ to $t = 1000$). The results are presented in terms of the accuracy of the prediction using the root-mean-square error metric (RMSE).

V. Empirical Study :TUNINDEX Forecasting

In order To illustrate the application potential of the proposed model on real data set we applied it to forecast the Tunisian Stock price index TUNINDEX. The data set were daily closed values from Jan . 05, 1998 to Dec. 31, 2002. Fig. 2 shows a temporal plot of the TUNINDEX .

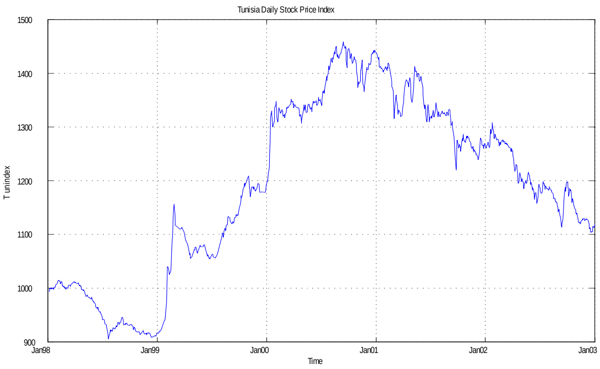

Fig. 2 Tunisian Stock Price Index (TUNNDEX)

The predictions are based on previous daily closed values of the TUNINDEX. Because the TUNINDEX time series has an irregular cycle it is very difficult to predict with linear methods. We used the daily values from Jan 05, 1998 to Dec 31, 2000 as training data and the the daily values from Jan 02, 2001 to Dec 31, 2001 as test data. Because the algorithm uses the test data to evaluate the performance of the networks, we used an additional data set from Jan 02, 2002 to Dec 31, 2002 for validation. We compare the proposed neuro-fuzzy network to classical neural network and ANFIS model.

TABLE II
Comparative Results for Mackey TUNINDEX

Method	Training RMSE	Test RMSE	Val RMSE
Classical NN	0.018	0.016	0.011
ANFIS	0.015	0.012	0.010
New model	0.012	0.010	0.009

As can be seen from Table II, the proposed neuro-fuzzy network trained by the EKF outperformed the classical and the ANFIS networks. The predicted (proposed neuro-fuzzy networks) and the measured TUNINDEX are displayed in Fig. 3.

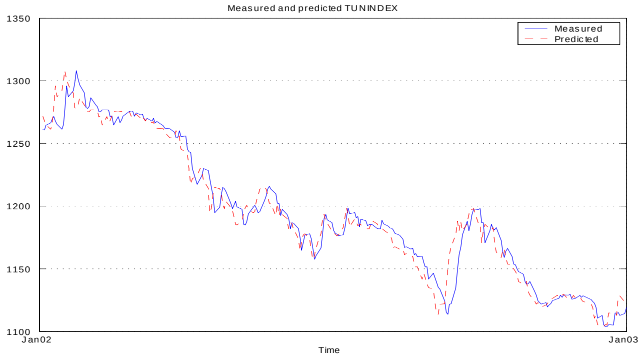

Fig. 3 Comparison of measured output and predicted (proposed network output) of the TUNINDEX

The solid line corresponds to the measured variable and the dashed line to the predicted TUNINDEX. From this figure, it is seen that the prediction values are satisfactory.

Fig. 4 present the predicted errors given by the proposed network and the ANFIS network. We can see clearly that the predicted values given by the proposed neuro-fuzzy network are better than given by the ANFIS network.

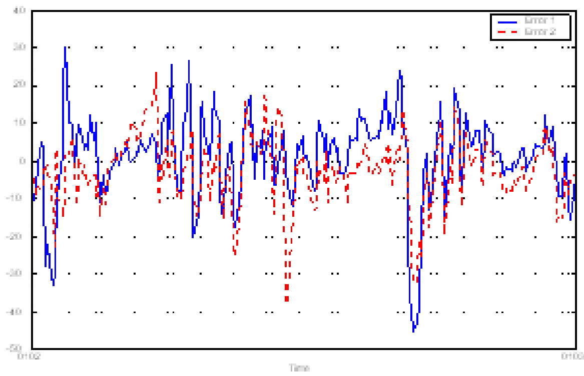

Fig. 4 Comparison of predicted errors given by the proposed network and the ANFIS network

VI. Conclusion

The previous section clearly demonstrated the effectiveness of the proposed model for the prediction of the Mackey-Glass time series and the Tunisian Stock price index TUNINDEX . In particular, the prediction results compare favourably with a conventional neural network technique and the neur-fuzzy ANFIS approach. Such results emphasise the benefits of the fusion of fuzzy and neural network technologies as it facilitates an accurate initialisation of the network in terms of the parameters of the fuzzy reasoning system. This increase in transparency of the neurofuzzy approach overcomes the drawback of a black-box description associated with conventional neural networks providing an improvement in prediction accuracy.

We believe that the encouraging results obtained herein with respect to the neural design in combination with existing non linear dynamic techniques, has a great potential for the forecasting of financial, economic and other time series generated by complex market driven systems.

ACKNOWLEDGMENT

The author wishes to thank the anonymous referees for their careful reading of the manuscript and their fruitful comments and suggestions.

REFERENCES

[1] J.J. Buckley, Y. Hayashi, Fuzzy neural networks: a survey, *Fuzzy Sets and Systems* 66 pp 1-13 1994
[2] S.Chokri, T Abdelwahed, Neural Network for Modeling Nonlinear Time Series: A New Approach. Springer-Verlag Berlin *Lecture Note in Computing Science*. 2659 pp 159-168 2003.
[3] H. Takagi, Fusion techniques of fuzzy systems and neural networks, and fuzzy systems and genetic algorithms, SPIE 2061 pp 402-413 1995.
[4] J.-S.Jang,ANFIS: adaptive-network-based fuzzy inference system, *IEEE Trans.Systems*,Man, Cybernet.23 pp 665—685 1993.
[5] Eiji Mizutani,J.-S.Jang,Coactive neural fuzzy modeling, in:Proc.of *IEEE Internat.Conf. on Neural Networks*, Vol.2,Perth,Australia, 760 –765 1995.
[6] M.F.Azeem,et al., Generalization of adaptive neuro-fuzzy inference systems, *IEEE Trans.Neural Networks* 11 pp 1332 –1346 2000.
[7] Shin-ichi Horikawa,et al., On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm, *IEEE Trans.Neural Networks* 3 pp 801 –806 1992.
[8] G. Puskorius, L. Feldkamp, Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks, *IEEE Trans. Neural Networks* 5 pp 279—297 1994.
[9] L. Zadeh. Fuzzy Sets, *Inf. Control*, vol 8 pp 338-353 1965.
[10] Mamdani, E. H. and S. Assilian. ,"An experiment in linguistic synthesis with a fuzzy logic controller." *Int. J. Man-Machine Studies* 7 1975.
[11] S. Shah, F. Palmieri, M. Datum, Optimal ltering algorithms for fast learning in feedforward neural Networks, *Neural Networks* 5 pp 779—787 1992.
[12] Kalman, R. E., " A New Approach to linear Filtering and Prediction Problems," *Transaction of ASME-Journal of basic Engineering*, pp 35-45 1960.
[13] Aoki, M. , *State-Space Modeling of Time Series*, Berlin: Springer-Verlag, 1987.
[14] S. Singhal, L. Wu, *Training multilayer perceptrons with the extended Kalman algorithm*, in: D. Touretzky Ed.,Advances in Neural Information Processing Systems, Vol. 1. Morgan Kaufmann, San Mateo, CA, pp 133-140 1989.