

Boosting Fuzzy Rules for Nonlinear System Identification Through Unscented Kalman Filter

M. Eftekhari, M. Maghfoori Farsangi, and M. Zeinalkhani

Abstract—This paper presents a new hybrid methodology for learning Sugeno-type fuzzy models via subtractive clustering, Adaptive Boosting Regression (AdaBoostR) and Unscented Kalman Filter (UKF). The generated fuzzy models are used for modeling nonlinear benchmark processes. In the proposed procedure, first one fuzzy rule is generated by subtractive clustering algorithm from available data of a given nonlinear process. Then this fuzzy rule is considered as a base model and AdaBoostR is employed in order to combine some of the weak learners (i.e. rules). Parameters of a rule are coded as the state vector in UKF and then UKF is used for fine tuning of these parameters. Moreover, as the second proposed method, Linear Kalman Filter (LKF) is utilized for adjusting only the output membership functions parameters (first order sugeno's parameters) of base models (i.e. rules). Three case studies are considered for illustrating the applicability of our proposed boosting methods. Results apparently show the obtained fuzzy models are superior to Adaptive Neuro-Fuzzy Inference System (ANFIS) in terms of both modeling accuracy and computational requirements. Also the comparison results confirm that the obtained fuzzy models are well comparable with those of achieved by one of the powerful and recently developed fuzzy identification methods.

Index Terms—Boosting, fuzzy identification, unscented Kalman filter.

I. INTRODUCTION

FUZZY models become useful when a system cannot be defined in precise mathematical terms. The non-fuzzy or traditional representations require a well-structured model and well defined model parameters. However, in practice, there are uncertainties, unpredicted dynamics and other unknown phenomena that cannot be mathematically modeled. The main contribution of fuzzy modeling theory is its ability to handle many practical problems that cannot be adequately represented by conventional methods. Fuzzy modeling of nonlinear systems has been the focus of many scientific researches. Takagi and Sugeno (TS) [1] have proposed a search algorithm for a fuzzy controller and generalized their techniques to fuzzy identification. Jang [1] proposed an architecture and a learning procedure which combines fuzzy logic with neural networks for inference. The Adaptive Neuro-Fuzzy Inference System (ANFIS) is capable of constructing input-output mapping accurately based on both human knowledge and input-

output data pairs [1].

In recent studies, ensemble learning methods namely bagging and boosting are employed for learning and aggregating fuzzy models [2], [3]. These methods combine some fuzzy weak models (base models) for the sake of generating a strong model in terms of accuracy. In this study our emphasis is to develop accurate fuzzy models via boosting. Boosting has its roots in a theoretical framework for studying machine learning called the "PAC" learning model [2], [3]. Kearns and Valiant were the first who posed the question of whether a "weak" learning algorithm which performs just slightly better than random guessing in the PAC model can be "boosted" into an arbitrarily accurate "strong" learning algorithm [2], [3]. The common advantage of all boosting methods is obtaining a strong model from the combination of weak ones. Adaptive Boosting (AdaBoost) is the most well-known boosting algorithm which is used in classification tasks [3], [4]. As the typical boosting schemes apply to discrete classification schemes, one can utilize them with necessary modifications and enhancements in order to apply them in continuous problems [5]-[8]. In recent studies the boosting mechanisms of weak regressors have been demonstrated some good performance [5]-[8].

The intention of this study is to develop fuzzy models for nonlinear processes that are better than ANFIS in terms of modeling accuracy. The first step is generating one fuzzy rule (i.e. TS fuzzy rule) as the base model via subtractive clustering. In the second step, a variant of AdaBoost adapted for continuous problems are utilized in order to learn and combine weak models. An embedded learning algorithm in AdaBoost is required for adjusting parameters of weak learners. Based on the previous studies available in the literature, the gradient descent algorithm is usually used for this purpose. In this paper, for learning the parameters, we employ a particular version of Extended Kalman filter (EKF) namely as Unscented Kalman Filter (UKF). The ability of UKF for parameter tuning of fuzzy models has been investigated and justified in a recently published research by authors [9]. As the second proposed strategy, Linear Kalman Filter (LKF) is utilized only for tuning the TS rules output parameters. The efficiency of the proposed Kalman filter based boosting procedures is assessed by comparing them with the ANFIS. Three benchmark examples are given to illustrate the effectiveness of our proposed approaches.

The remainder of paper is organized as following. Subtractive clustering is described in the following which is a method for generating fuzzy rules from input-output data pairs. The third section discusses about the UKF algorithm. Section IV deals with combining fuzzy rules via boosting. The proposed boosting procedure and

Manuscript received July 6, 2012; revised September 21, 2013.

M. Eftekhari and M. Zeinalkhani, are with the Department of Computer Engineering, School of Engineering, Shahid Bahonar University of Kerman, Kerman, Iran, (e-mails: m.eftekhari@uk.ac.ir and zeinalkhani@gmail.com).

M. Maghfoori Farsangi is with the Department of Electrical Engineering, School of Engineering, Shahid Bahonar University of Kerman, Kerman, Iran, (e-mail: mMaghfoori@uk.ac.ir).

Publisher Item Identifier S 1682-0053(13)2035

parameter learning of weak models through UKF are given in Section V. Simulation results and discussions are presented in Section VI. Finally, Section VII concludes the paper.

II. FUZZY MODELING BASED ON CLUSTERING

Subtractive clustering is usually utilized to obtain a set of rules and avoiding the problems inherent in grid partitioning based clustering techniques (i.e. RB explosion) [9], [10]. This technique is employed since it allows a scatter input-output space partitioning [9], [10].

Subtractive clustering is, essentially, a modified form of the Mountain Method. Thus, let Z be the set of N data points obtained by concatenation of the inputs and output. In the algorithm, each point is seen as a potential cluster centre, for which some measure of potential is assigned according to (1)

$$p_i = \sum_{j=1}^N e^{-\alpha \|z_i - z_j\|^2} \quad (1)$$

where $\alpha = 4/r_a^2$ and $r_a > 0$ defines the neighborhood radius for each cluster center. Thus, the potential associated with each cluster depends on its distance to all the points, leading to clusters with high potential where neighborhood is dense. After calculating potential for each point, the one with higher potential is selected as the first cluster center. Let z_1^* be the center of the first group and p_1^* its potential. Then the potential for each z_i is reduced according to (2), especially for the points closer to the center of the cluster

$$p_i = p_i - p_1^* e^{-\beta \|z_i - z_1^*\|^2} \quad (2)$$

also, $\beta = 4/r_b^2$ and $r_b > 0$ represent the radius of the neighborhood for which significant potential reduction will occur. The radius for reduction of potential should be to some extent higher than the neighborhood radius to avoid closely spaced clusters, typically, $r_b = 1.25r_a$. Since the points closer to the cluster center will have their potential strongly reduced, the probability for those points to be chosen as the next cluster is lower. This procedure (selecting centers and reducing potential) is carried out iteratively until stopping criteria is satisfied. Additionally two threshold levels are defined, one above the point which is selected for a cluster center and the other below the point which is rejected.

By the end of clustering, a set of fuzzy rules will be obtained. Each cluster represents a rule. However, since the clustering is carried out in a multidimensional space, the related fuzzy sets must be obtained. As each axis refers to a variable, the centers of the Membership Functions (MFs) (which are Gaussian in this case) are obtained by projecting the center of each cluster in the corresponding axis. The widths are obtained on the basis of the radius r_a for each dimension. Therefore, number of produced clusters and subsequently the number of generated rules by subtractive clustering can be controlled via varying values of r_a for all dimensions (inputs and output).

After constructing a fuzzy model, it is needed to tune its parameters for acquiring more accuracy. Several parameter tuning methods are introduced in the literature in which gradient based methods are very fast and

vastly applicable techniques. Nevertheless, gradient based approaches suffer from some deficiencies such as: trapping in local optima and their dependence on the gradient. The latter disadvantage is more serious than the first one especially where the model contains some non-derivative functions. Parameter tuning is inapplicable if such the functions are in the models. Therefore, in recent years some derivative free approaches have been developed for parameter tuning of models. Kalman Filters are successful iterative algorithms that have been vastly utilized for parameter tuning [9]. Unscented Filter is one of the Kalman Filter based methods that is derivative free and provides an accuracy as the same as the Extended Kalman filters (that are derivative based). According to the above mentioned drawbacks of other methods and also based on the previous successful use of UKF in [9], in this paper, UKF is used as the parameter tuning algorithm. In the following, the UKF algorithm and its application in parameter tuning are described.

III. UNSCENTED KALMAN FILTER

UKF is a newly developed version of Kalman Filters, which was introduced to address the deficiencies of linearization in EKF (problems such as difficulties in calculating Jacobian matrices for linearization). In the unscented transformation [11], on which the UKF is based, a set of weighted sigma points are deterministically chosen so that certain properties of these points match those of prior distribution. Each point is then propagated through a nonlinear function and the properties of the transformed set are calculated. With this set of points, the unscented transform guarantees the same performance as the truncated second order Gaussian filter. In other words, the UKF has the advantage that works with the same order of calculations as an EKF but without the need to calculate Jacobian or Hessian matrices [9], [11].

The UKF algorithm used in this paper is presented in relation to a computationally simple subclass of state-space models with additive process and measurement noise which is also assumed in the case of EKF. In such a case, the system state need not be augmented with the noise random variables. This reduces the dimension of sigma points as well as the total number of sigma points used. This model is expressed as below

$$x_{k|k-1} = f(x_{k-1}) + v_k \quad (3)$$

$$t_k = h(x_k) + \omega_k \quad (4)$$

where x_k is the state vector at time k , t_k is the observation vector, v_k is the system noise vector, ω_k is the measurement noise vector, $f(\cdot)$ is the process model, and $h(\cdot)$ is the measurement model. Now let L be the dimension of the original state vector, Q and R are the covariance matrices of the process and measurement noise, respectively. Fig. 1 in the following shows the learning procedure by UKF that is a six step iterative algorithm. Number of iterations in this algorithm can be adjusted by user. The initial state vector is updated through some iterations of the UKF algorithm.

The details of different steps of UKF have been described in [9].

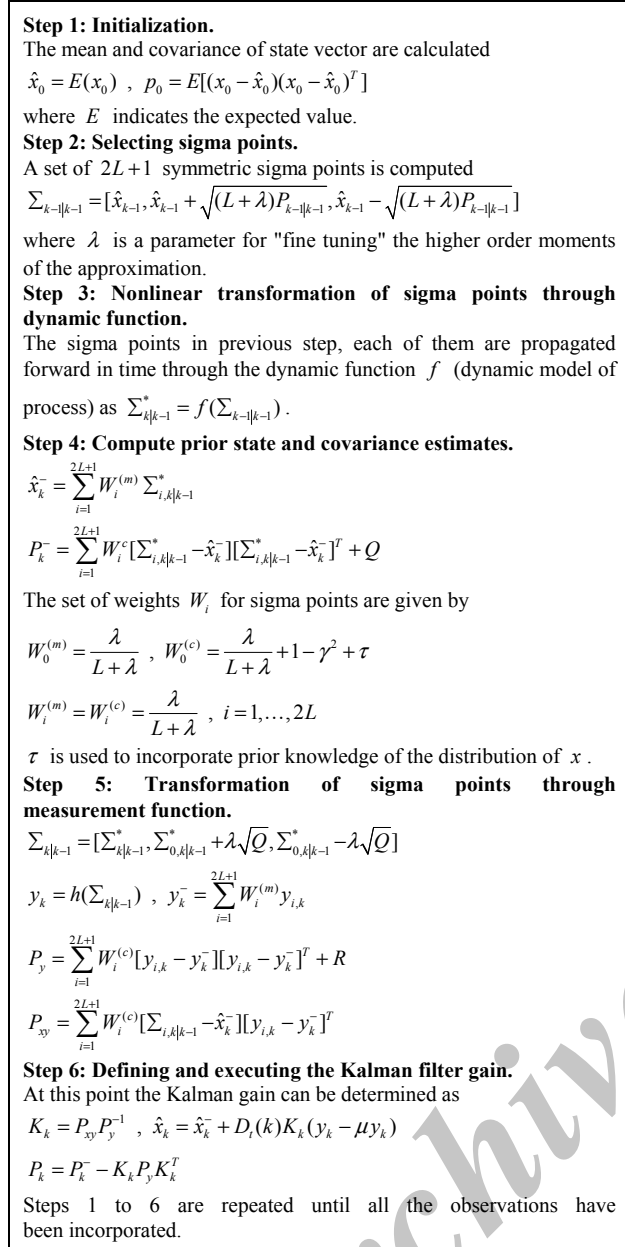


Fig. 1. The UKF iterative algorithm.

IV. COMBINING WEAK LEARNERS VIA ADABOOSTR

An ensemble is a set of classifiers constructed with a given algorithm. Each new example is classified by combining the predictions of every classifier from the ensemble. These predictions can be combined by taking the average (for regression tasks) or the majority vote (for classification tasks), or by taking more complex combinations. Two types of ensemble learning that are commonly used in studies are bagging and boosting. There are several variants of boosting, AdaBoost is the most well-known. These methods assign a weight to each example. Initially, all the examples have the same weight. In each iteration a new classifier, named base or weak, is constructed using the base learning method. The construction of the base classifier must take into account the weights distribution. Then, the weight of each example is adjusted, depending on the correctness of the prediction of the base classifier for that example. The final classification is obtained from a weighted vote of the base

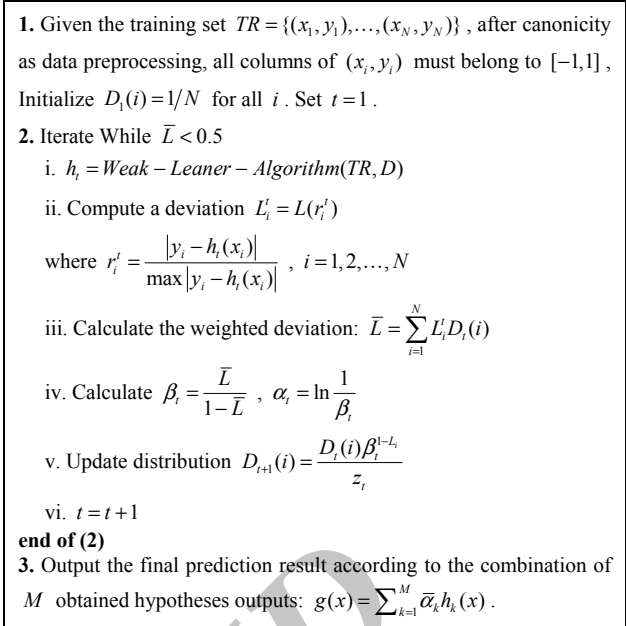


Fig. 2. Pseudo-code for AdaBoostR algorithm.

classifiers. An ensemble method is said to be a boosting method, if it has the ability of generating a strong classifier from some weak ones. Also, a model (classifier) is weak if it can guarantee the obtained classifier will have a training error smaller than fifty percent [2]-[8]. On the other hand, the boosting technique [2]-[8] has been successful in the development of highly efficient classifiers emerging on a basis of a collection of weak classifiers whose performance could be slightly better than random guessing [6]-[8].

In this study a variant of AdaBoost which is adapted for utilizing in regression problems are employed in order to combine fuzzy models. This algorithm is called AdaBoostR (Adaptive Boosting Regression) [6]-[8]. Fig. 2 gives a pseudo-code for this algorithm.

As can be seen, AdaBoostR is very similar to AdaBoost but the only difference between these two algorithms is related to computing the error of each base model. In the later algorithm, error is computed via a deviation factor which is named L_t^i (the error related to base model h_t on i th data sample). The value of L_t^i can be calculated by any of these given relations

$$L(r_t^i) = r_t^i \quad \text{or} \quad L(r_t^i) = (r_t^i)^2 \quad \text{or} \quad L(r_t^i) = 1 - e^{-r_t^i} \quad (5)$$

V. PROPOSED FUZZY MODELING PROCEDURE

In this work, first subtractive clustering is utilized for generating a base fuzzy model. Second, the AdaBoostR is used for combining these base models. Fuzzy base model in this study consists of only one TS fuzzy rule. Subtractive clustering can be employed for producing TS fuzzy models with linear outputs and radii parameters have important role in this procedure. The radius parameter for each dimension can have a value in the range $[0, 2]$. A general high value of radius (near 2) for all dimensions leads to fewer clusters and consequently fewer rules in corresponding TS fuzzy model. Therefore, by choosing an overall high value of radius for all dimensions one can generate a fuzzy model containing only one rule. In the following, the flowchart of our proposed fuzzy modeling procedure based on subtractive clustering is depicted in

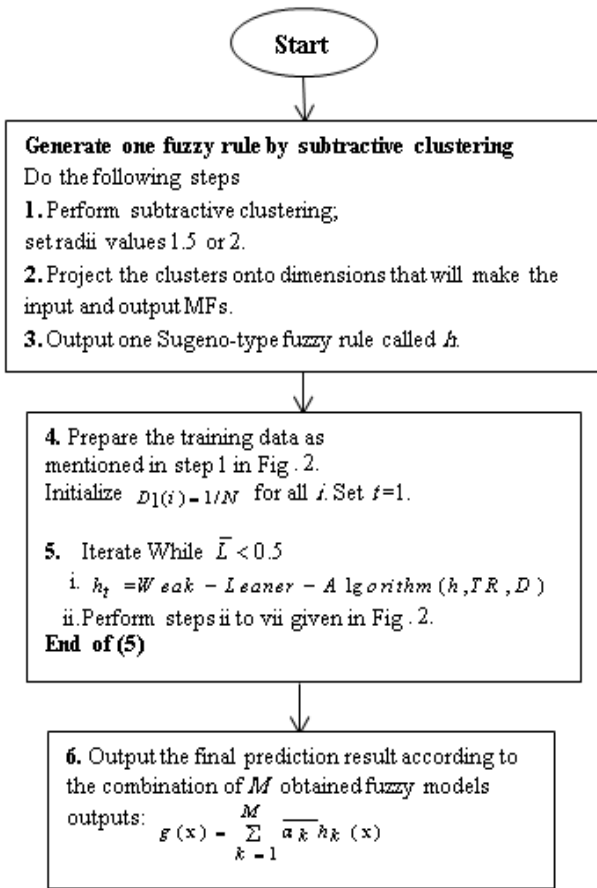


Fig. 3. Flowchart of the proposed fuzzy modeling procedure.

Fig. 3. The proposed method can be considered as a Fuzzy Inference System (FIS) in which the AdaBoostR is used for aggregating fuzzy rules.

In the first step of AdaBoostR main loop (Fig. 2, line 5-i), a learner must be utilized for learning base models (weak learners). As concluded from past researches [2], [3], [12], this learning algorithm usually is the gradient descent in which the distribution D_t is incorporated into learning procedure by multiplying into mean square error (MSE). In this research, the UKF is utilized in order to tune all the weak learner parameters.

Fuzzy MF tuning can be seen as a least-square optimization problem, where the error is the difference between the fuzzy model outputs and the target values corresponding to those outputs. For casting this optimization problem in a form suitable for UKF, the MF parameters of inputs and outputs in a TS-FIS model constitute the states of a nonlinear system. The outputs of fuzzy model are the nonlinear system outputs to which the UKF is applied. h in (4) represents the fuzzy model. As mentioned in Section III, vector t denotes the target vector for the fuzzy system outputs and $h(k)$ is used to denote the actual fuzzy model outputs at k th iteration and shown as

$$\begin{aligned} t &= [t_1, \dots, t_n] \\ h(k) &= [h_1(k), \dots, h_n(k)] \end{aligned} \quad (6)$$

UKF can be easily used for tuning base model parameters. For the sake of simplicity, a process with two inputs and one output is considered. The fuzzy model of this process for identification has n inputs (a subset of its previous inputs and output) and one output. As mentioned

in Section II, subtractive clustering creates a TS-FIS base model with one rule and one MF for each input. Gaussian MF for each input has 2 parameters and each output MF (linear) has $(n+1)$ parameters. Therefore, the number of MFs parameters for the base model is calculated as

$$N_b = 2n + n + 1 = 3n + 1 \quad (7)$$

subsequently, number of all states

$$N_{states} = N_b \quad (8)$$

let's, $C_{i,j}$ and $\sigma_{i,j}$ denote the center and half-width of the i th MF of the j th input, respectively, and $\alpha_n, \alpha_{n-1}, \dots, \alpha_0$ represent the output MF parameters. The nonlinear system state can then be represented as (kama bein $C_{1,n}$ va α_n)

$$x = [\sigma_{1,1}, C_{1,1}, \dots, \sigma_{1,n}, C_{1,n}, \alpha_n, \alpha_{n-1}, \dots, \alpha_0] \quad (9)$$

the nonlinear model for the UKF procedure can be stated as

$$\begin{aligned} x_{k|k-1} &= x_{k-1} \\ t_k &= h(x_k) \end{aligned} \quad (10)$$

where $h(x_k)$ is the fuzzy system nonlinear mapping between MF parameters and the outputs of fuzzy model. Initially, artificial process noise and measurement noise with small variances are added to the system model [9]

$$\begin{aligned} x_{k|k-1} &= x_{k-1} + v_k \\ t_k &= h(x_{k-1}) + \omega_k \end{aligned} \quad (11)$$

where v_k and ω_k are artificially added noise processes, Q and R matrices are tuning parameters which can be considered as the covariance matrices of the artificial noise with small positive initial values. In (3), $f(\cdot)$ is the identity mapping. t_k is the target output for the fuzzy system. Assume that the initial values of the state vector and artificially added noise processes are Gaussian and independent of each other. The Kalman gain K_k is set to zero at the first iteration. This Kalman gain is multiplied by $D_t(k)$ at the k th iteration of UKF algorithm when the k th observation (i.e. k th data sample that its weight is $D_t(k)$) is incorporated into UKF

$$\hat{x}_k = \hat{x}_k^- + D_t(k)K_k(y_k - \mu y_k) \quad (12)$$

also, for the case of comparison, a second approach is proposed and examined in which only the output parameters of base models (TS rules) are tuned through LKF.

VI. SIMULATION RESULTS

Three case studies are considered for doing experimental results. All of them are taken from the past literature namely Mackey-Glass time series, a well-known nonlinear system and a flexible robot arm [9], [13], [14]. More details of these case studies are given in the following. For each case study, the AdaBoostR algorithm was run 20 times (for both LKF-based and UKF-based methods) and the best results are reported in the following.

A. Case 1: Mackey-Glass Time Series

The Mackey-Glass time series is described as follows [13]

$$\dot{x} = \frac{ax(t-r)}{1+x^b(t-r)} - cx(t) \quad (13)$$

TABLE I
CHARACTERISTICS OF THE BEST MODELS OBTAINED BY UKF-BASED, LKF-BASED MODELS AND OBTAINED ANFIS MODEL

Code	Model characteristics	Generating method	Learning algorithm
M1	TS-FIS (3 rules)	Subtractive clustering $ra = 0.8$	ANFIS (100 iterations)
M2	Base model: one TS-rule	Subtractive clustering $ra = 1.5$	AdaBoost R (3 base models) and UKF (10 iterations)
M3	Base model: one TS-rule	Subtractive clustering $ra = 1.5$	AdaBoost R (3 base models) and LKF (10 iterations)

TABLE II
COMPARING PERFORMANCES OF THE BEST OBTAINED BOOSTED MODELS AND ANFIS

Code	Train error \pm STD	Test error \pm STD
M1	4.3828e-6 \pm 1e-7	4.1516e-6 \pm 2.3e-7
M2	Error of each weak learner	Error after combining
	$\bar{L}_1 = 0.3991$	4.1403e-6 \pm 1.5e-9
	$\bar{L}_2 = 0.4113$	
$\bar{L}_3 = 0.4001$		
M3	Error of each weak learner	Error after combining
	$\bar{L}_1 = 0.4619$	4.6761e-6 \pm 1.3e-7
	$\bar{L}_2 = 0.4743$	
$\bar{L}_3 = 0.4829$	5.6549e-6 \pm 1.8e-7	

TABLE III
COMPARING PERFORMANCES OF THE BEST OBTAINED BOOSTED MODELS AND ANFIS

Code	Train error \pm STD	Test error \pm STD
M1	1.3552e-3 \pm 3.1e-4	1.0709e-3 \pm 2.6e-4
M2	Error of each weak learner	Error after combining
	$\bar{L}_1 = 0.3195$	1.3178e-4 \pm 3.1e-7
	$\bar{L}_2 = 0.3407$	
$\bar{L}_3 = 0.4180$		
M3	Error of each weak learner	Error after combining
	$\bar{L}_1 = 0.4232$	2.4356e-4 \pm 1.4e-5
	$\bar{L}_2 = 0.4456$	
$\bar{L}_3 = 0.4487$	2.1122e-4 \pm 1.1e-5	

the parameters of Mackey-Glass time series are $a = 0.2$, $b = 10$, $c = 0.1$, and $r = 30$. The goal is to predict $x(t)$ from $x(t-1)$, $x(t-2)$ and $x(t-3)$ 1000 data points are obtained from (13) using the fourth order Runge-Kutta method with a step length of 1 and the initial condition $x(0) = 1.2$ where 500 pairs of data are used for training and the other 500 for test. The characteristics of obtained ANFIS model as well as those of the best models obtained by UKF-based and LKF-based models are given in Table I.

For the models introduced in Table I, MSE values for train and test data are given in Table II. The first row shows the characteristics of obtained ANFIS model and its related MSE values for train and test data. The second and third rows show the characteristics for the best models obtained by UKF-based and LKF-based methods, respectively, and their MSE values for train and test data. Also the standard deviations (i.e. STD) of MSE values for 20 runs of AdaBoostR are given as well. In both UKF-based and LKF-based methods 3 base models are generated by boosting algorithm.

B. Case 2: A Nonlinear System

The non-linear system studied in [14] is taken as the second example

$$y(k) = g(y(k-1), y(k-2)) + u(k) \quad (14)$$

where

$$g(y(k-1), y(k-2)) = \frac{y(k-1)y(k-2)(y(k-1)-0.5)}{1 + y^2(k-1) + y^2(k-2)} \quad (15)$$

The system output depends on both its past values and the current input. The goal is to approximate the non-linear component $g(y(k-1), y(k-2))$ of the system with a fuzzy model. As in [14], 400 simulated data points are generated from the system model (14). Starting from the equilibrium state (0,0), 200 samples of training data are obtained with a random input signal $u(k)$ uniformly distributed in $[-1.5, 1.5]$, followed by 200 samples of testing data obtained using a sinusoidal input signal $u(k) = \sin(2\pi k/25)$.

Table III, given below, presents MSE values for train and test data. The first row shows the characteristics of obtained ANFIS model and its related MSE values for train and test data. The second and third rows give the characteristics for the best models obtained via UKF-based and LKF-based boosting methods respectively and their MSE values for train and test data. Also, the standard deviations (i.e. STD) of MSE values for 20 runs of AdaBoostR are given as well. In both UKF-based and LKF-based methods 3 base models are generated by boosting algorithm.

As can be seen from the above Tables II and III, the first approach which combines TS rules via AdaBoostR and UKF is better than ANFIS in all cases regarding to modeling accuracy and number of iterations required for parameter learning of weak models. ANFIS uses a two pass algorithm in each iteration. In the first pass gradient descent is utilized for input MFs parameters tuning and in the second pass recursive least square is employed for output MF parameter adjusting. Therefore computational efforts

of our first proposed approach are less than ANFIS. Furthermore, authors' study about the UKF justifies the better optimization performance of UKF comparing to ANFIS [9].

Last rows of the above Tables II and III show the obtained results of our second proposed approach. In this approach AdaBoostR combines TS rules whereas employing LKF for base model parameter learning.

Employing LKF results in modeling accuracies that are a bit worse than those of ANFIS, but computational requirements of this approach is very less than the other ones.

C. Case 3: Flexible Robot Arm

The arm is installed on an electrical motor. The transfer function represents the relation between the measured reaction torque of the structure on the ground to the acceleration of the flexible arm. Therefore, the input is reaction torque of the structure and the output is-

TABLE IV
COMPARING PERFORMANCES OF THE BEST OBTAINED BOOSTED MODELS AND ANFIS

Code	Train error \pm STD	Test error \pm STD
M1	8.9935e-5 \pm 5e-7	7.943e-5 \pm 3.4e-7
	Error of each weak learner	Error after combining
M2	$\bar{L}_1 = 0.3543$ $\bar{L}_2 = 0.3444$ $\bar{L}_3 = 0.3653$	2.3456e-5 \pm 1.34e-7
	Error of each weak learner	Error after combining
M3	$\bar{L}_1 = 0.3691$ $\bar{L}_2 = 0.3770$ $\bar{L}_3 = 0.4456$	6.9135e-5 \pm 4.24e-7

TABLE V
COMPARING THE BEST RECENTLY REPORTED FUZZY MODEL AND THOSE OF THIS PAPER FOR THE FIRST CASE STUDY.

Ref.	No. of rules	Rule type	MSE train	MSE test
[14]	5 rules (initial) 1 rule (optimized)	Linear Sugeno	---	---
M2	3 rules	Linear Sugeno	4.2871e-6	4.1388e-6
M3	3 rules	Linear Sugeno	5.4749e-6	4.5461e-6

TABLE VI
COMPARING THE BEST RECENTLY REPORTED FUZZY MODEL AND THOSE OF THIS PAPER FOR THE SECOND CASE STUDY.

Ref.	No. of rules	Rule type	MSE train	MSE test
[9]	5 rules (initial) 3 rules (optimized)	Linear Sugeno	---	---
M2	3 rules	Linear Sugeno	1.3147e-4	9.9830e-5
M3	3 rules	Linear Sugeno	2.2956e-4	2.0022e-4

TABLE VII
COMPARING THE BEST RECENTLY REPORTED FUZZY MODEL AND THOSE OF THIS PAPER FOR THE THIRD CASE STUDY.

Ref.	No. of rules	Rule type	MSE train	MSE test
[14]	5 rules (initial) 1 rule (optimized)	Linear Sugeno	---	---
M2	3 rules	Linear Sugeno	2.8745e-5	2.3322e-5
M3	3 rules	Linear Sugeno	5.9240e-5	6.8711e-5

acceleration of the flexible arm. The applied input is a periodic sine sweep. 1000 data samples for the third example are taken randomly from the public domain for system identification [15].

Seventy percent of the data are selected randomly for training and the rest is used for test. Table IV, presents MSE values for train and test data. The first row shows the characteristics of obtained ANFIS model and its related MSE values for train and test data. The second and third rows give the characteristics for the best models obtained by UKF-based and LKF-based boosting methods respectively and their MSE values for train and test data. Also the standard deviations (i.e. STD) of MSE values for 20 runs of AdaBoostR are given as well in Table IV. In both UKF-based and LKF-based methods 3 base models are generated by boosting algorithm.

D. Comparisons

For completeness, in the following we are going to compare the results of this paper with those of obtained by one of the recently developed and powerful methods in the literature. The method has been introduced by details in [14].

Table V in the following gives the results of this paper and those of [14] for the first case study. As it can be seen,

the method introduced in [14] obtains only one rule while the proposed methods of this paper employ 3 rules. Therefore, the best fuzzy model in [14] is better than those of this paper in terms of number of obtained rules (i.e. model complexity). However, the proposed methods in this paper perform better than that of ref. [14] in terms of modeling accuracy (i.e. lower modeling errors).

For the second case study, the comparative results are listed in Table VI. The best results are given in the second row of Table VI that are those of obtained by UKF based boosting method. For more illustration the second row of Table V is highlighted. Also, the results obtained by LKF method are well comparable to those of [14]. Table VI apparently confirms the efficiency of the proposed boosting methodologies for nonlinear system identification. The best fuzzy model, which has been achieved in [14], utilizes 4 membership functions in 3 rules while the proposed models in this paper use 6 membership functions. This is the only difference between the obtained fuzzy models of this paper and that of the [14].

Similar to the previous case studies, for the third case, the results of comparing the best models obtained by the proposed methods with those of [14] are given in the Table VII in the following.

As it is apparent from Tables V-VII, the UKF-based

fuzzy models outperform the others in terms of modeling accuracy.

VII. CONCLUDING REMARKS

Two novel methods for boosting regression models are presented in this study. The proposed methods introduce a new aggregation method for combining fuzzy rules via AdaBoostR algorithm. Base models that are fuzzy rules are produced by subtractive clustering according to data samples. Despite of the past researches in the field of boosting algorithms in which gradient descent algorithms have been usually used for learning the parameters of base models, in present study UKF is employed. As it has shown in the literature, UKF can preserve the accuracy of EKF as well as some added advantages that are its derivative free property (despite of EKF and gradient descent methods) and solving deficiencies of linearization in EKF. Due to these benefits, UKF is utilized as embedded algorithm in AdaBoostR for learning weak learners' parameters. All parameters of base models including input MFs parameters and output ones are sequentially ordered as state vector in UKF. This procedure that combines subtractive clustering, AdaBoostR and UKF algorithms is the first proposed approach. The obtained results reveal the superiority of this approach in all cases than the ANFIS.

The second proposed method employs AdaBoostR and LKF for fuzzy modeling. Although modeling accuracies (obtained by this approach) are worse than both UKF-based approach and ANFIS for all cases, but they are well comparable to those of obtained by ANFIS. Also, computational requirements of this approach are significantly less than the other ones.

Consequently, the comparison results reveal that the proposed UKF based method outperforms recently developed fuzzy models in terms of modeling accuracy. The results obtained from three case studies show that UKF based method optimization leads to more accurate models. Also, simulation results illustrate that the proposed UKF-based boosting algorithm is easy to implement, computationally effective and particularly results in accurate fuzzy models for nonlinear systems.

REFERENCES

- [1] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference systems," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665-685, May 1993.
- [2] W. Pedrycz and K. C. Kwak, "Boosting of granular models," *Fuzzy Sets and Systems*, vol. 157, no. 22, pp. 2934-2953, Nov. 2006.
- [3] T. O. Zyera, R. Alhajja, and K. Barkera, "Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening," *J. of Network and Computer Applications*, vol. 30, no. 1, pp. 99-113, Jan. 2007.
- [4] H. Drucker, Boosting Using Neural Nets, in: A. Sharkey (Ed.), *Combining Artificial Neural Nets: Ensemble and Modular Learning*, Springer, 1999.

- [5] N. Duffy and D. Helmbold, "Boosting methods for regression," *Machine Learning*, vol. 47, no. 2-3, pp. 153-200, May/June. 2002.
- [6] S. Borra and A. D. Ciaccio, "Improving non-parametric regression methods by bagging and boosting," *Comput. Statist. Data Anal.*, vol. 38, no. 4, pp. 407-420, Feb. 2002.
- [7] N. Rooney, D. Patterson, A. Tsymbal, and S. Anand, "Random sub spacing for regression ensembles," in *Proc. 17th Int. FLAIRS Conf. on Artificial Intelligence*, pp. 532-537, Mar. 2004.
- [8] M. Assaad, R. Bone, and H. Cardot, "A new boosting algorithm for improved time-series forecasting with recurrent neural networks," *Information Fusion*, vol. 9, no. 1, pp. 41-55, Jan. 2008.
- [9] M. Eftekhari and S. D. Katebi, "Extracting compact fuzzy rules for nonlinear system modeling using subtractive clustering, GA and unscented filter," *Applied Mathematical Modelling*, vol. 32, no. 12, pp. 2634-2651, Dec. 2008.
- [10] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Systems*, vol. 2, no. 3, pp. 267-278, Mar. 1994.
- [11] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," in *Proc. of the IEEE*, vol. 92, no. 3, pp. 401-422, Mar. 2004.
- [12] R. S. Zemel and T. Pitassi, "A gradient-based boosting algorithm for regression problems," in *Advances in Neural Information Processing Systems, Cambridge, MA, USA*, vol. 13, pp. 696-702, Dec. 2001.
- [13] M. Eftekhari, S. D. Katebi, M. Karimi, and A. H. Jahanmiri, "Eliciting transparent fuzzy model using differential evolution," *Applied Soft Computing*, vol. 8, no. 1, pp. 466-476, Jan. 2008.
- [14] M. Eftekhari, M. Eftekhari, M. Majidi, and H. Nezamabadipour, "Securing interpretability of fuzzy models for modeling nonlinear MIMO systems using a hybrid of evolutionary algorithms," *Iranian J. of Fuzzy Systems*, vol. 9, no. 1, pp. 61-77, Feb. 2012.
- [15] B. L. R. De Moor (Ed.), *DalSy: Database for the Identification of Systems, Department of Electrical Engineering, ESAT/SISTA*, Technical Report 97-70, K. U. Leuven, Belgium, URL: <http://www.esat.kuleuven.ac.be/sista/daisy/>.

Mahdi Eftekhari received his B.Sc. in computer engineering from Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran in September 2001. He obtained his M.Sc. and Ph.D. degrees in Artificial Intelligence from the same department in 2004 and 2008, respectively. He has been a faculty member of Computer Engineering Department at Shahid Bahonar University of Kerman, Kerman, Iran since 2008. His research interests include fuzzy systems and modeling, evolutionary algorithms, data mining, machine learning and application of intelligent methods in bioinformatics. He is the author and co-author of about 50 papers in cited journals and conferences. Dr. Eftekhari is a member of Iranian Society of Fuzzy Systems.

Malihe Maghfoori Farsangi received her B.Sc. degree in Electrical Engineering from Ferdowsi University, Iran in 1995, and Ph.D. degree in Electrical Engineering from Brunel Institute of Power Systems, Brunel University, UK in 2003. Since 2003, she has been with Kerman University, Kerman, Iran, where she is currently an Associative Professor of Electrical Engineering. Her research interests include power system control and stability and computational intelligence.

Mohsen Zeinalkhani was born in Abhar, Iran in 1988. He received his B.Sc. in computer engineering from Department of Computer Engineering, Kashan University, Iran in September 2009. He obtained his M.Sc. degree in Artificial Intelligence from Department of Computer Engineering, Shahid Bahonar University of Kerman, and Kerman, Iran. His research interests include fuzzy systems, machine learning, data mining and pattern recognition.