Adaptive Fuzzy Filtering in a Deterministic Setting

Mohit Kumar, Member, IEEE, Norbert Stoll, and Regina Stoll

Abstract—Many real-world applications involve the filtering and estimation of process variables. This study considers the use of interpretable Sugeno-type fuzzy models for adaptive filtering. Our aim in this study is to provide different adaptive fuzzy filtering algorithms in a deterministic setting. The algorithms are derived and studied in a unified way without making any assumptions on the nature of signals (i.e., process variables). The study extends, in a common framework, the adaptive filtering algorithms (usually studied in signal processing literature) and p-norm algorithms (usually studied in machine learning literature) to semilinear fuzzy models. A mathematical framework is provided that allows the development and an analysis of the adaptive fuzzy filtering algorithms. We study a class of nonlinear LMS-like algorithms for the online estimation of fuzzy model parameters. A generalization of the algorithms to the p-norm is provided using Bregman divergences (a standard tool for online machine learning algorithms).

Index Terms—Adaptive filtering algorithms, Bregman divergences, p-norm, robustness, Sugeno fuzzy models.

I. INTRODUCTION

A REAL-WORLD complex process is typically characterized by a number of variables whose interrelations are uncertain and not completely known. Our concern is to apply, in an online scenario, the fuzzy techniques for such processes aiming at the filtering of uncertainties and the estimation of variables. The adaptive filtering algorithms applications are not only limited to the engineering problems but also, e.g., to medicinal chemistry, where it is required to predict the biological activity of a chemical compound before its synthesis in the laboratory [1]. Once a compound is synthesized and tested experimentally for its activity, the experimental data can be used for an improvement of the prediction performance (i.e., online learning of the adaptive system). Adaptive filtering of uncertainties may be desired, e.g., for an intelligent interpretation of medical data that are contaminated by the uncertainties arising from the individual variations due to a difference in age, gender, and body conditions [2].

We focus on a process model with n inputs (represented by the vector \( x \in \mathbb{R}^n \)) and a single output (represented by the scalar \( y \)). Adaptive filtering algorithms seek to identify the unknown parameters of a model (being characterized by a vector \( w^* \)) using input–output data pairs \( \{x(j), y(j)\} \) related via

\[
y(j) = M(x(j), w^*) + n_j
\]

where \( M(x(j), w^*) \) is the model output for an input \( x(j) \), and \( n_j \) is the underlying uncertainty. If the chosen model \( M \) is nonlinear in parameter vector \( w^* \) (as is the case with neural and fuzzy models), the standard gradient-descent algorithm is mostly used for an online estimation of \( w^* \) via performing following recursions:

\[
w_j = w_{j-1} - \mu \frac{\partial E_r(w, j)}{\partial w} w_{j-1}
\]

\[
E_r(w, j) = \frac{1}{2} [y(j) - M(x(j), w)]^2
\]

where \( \mu \) is the step size (i.e., learning rate).

If we relax our model to be linear in \( w^* \) i.e., input–output data are related via

\[
y(j) = G_j^T w^* + n_j,
\]

where \( G_j \) is the regressor vector then a variety of algorithms are available in the literature for an adaptive estimation of linear parameters [3]. The most popular algorithm is the LMS because of its simplicity and robustness [4], [5]. Many LMS-like algorithms have been studied for linear models [3], [4] while addressing the robustness, convergence, and steady-state error issues. A particular class of algorithms takes the update form as

\[
w_j = w_{j-1} - \mu \phi(G_j^T w_{j-1} - y(j)) G_j
\]

where \( \phi \) is a nonlinear scalar function such that different choices of the functional form lead to the different algorithms, as stated in Table I. The generalization of LMS algorithm to p-norm (\( 2 \leq p < \infty \)) is given by the update rule [9], [10]:

\[
w_j = f^{-1}(f(w_{j-1}) - \mu |G_j^T w_{j-1} - y(j)| G_j).
\]

Here, \( f \) (a \( p \) indexing for \( f \) is understood), as defined in [10], is the bijective mapping \( f : \mathbb{R}^K \to \mathbb{R}^K \) such that

\[
f = [f_1 \cdots f_K]^T,
\]

\[
f_i(w) = \frac{\text{sign}(w_i)}{\|w||_q} w_i^{q-1}
\]

where \( w = [w_1 \cdots w_K]^T \in \mathbb{R}^K, q \) is dual to \( p \) (i.e., \( 1/p + 1/q = 1 \), and \( \| \cdot \|_q \) denotes the \( q \)-norm defined as

\[
\|w\|_q = \left( \sum_i |w_i|^q \right)^{1/q}
\]
The inverse $f^{-1}: R^K \rightarrow R^K$ is given as

$$f^{-1} = \left[ f_1^{-1} \cdots f_K^{-1} \right]^T,$$

where $v = [v_1 \cdots v_K]^T \in R^K$.

Sugeno-type fuzzy models are linear in consequents and nonlinear in antecedents (i.e., membership functions parameters). When it comes to the online estimation of fuzzy model parameters, the following two approaches, in general, are used.

1) The antecedent parameters are adapted using gradient-descent and the consequent parameters by the recursive least-squares algorithm [11], [12].

2) A combination of data clustering and recursive least-squares algorithm is applied [13], [14].

The wide use of gradient-descent algorithm for adaptation of nonlinear fuzzy model parameters (e.g., in [15]) is due to its simplicity and low computational cost. However, gradient-descent-based algorithms for nonlinear systems are not justified by rigorous theoretical arguments [16]. Only a few papers dealing with the mathematical analysis of the adaptive fuzzy algorithms have appeared till now. The issue of algorithm stability has been addressed in [17]. The authors in [18] introduce an “energy gain” criteria are provided in [19].

To the knowledge of the authors, the fuzzy literature still lacks

1) the development and deterministic mathematical analysis (in terms of filtering performance) of the methods that extend the Table I type algorithms (i.e., LMF, LMMN, sign error, etc.) to the interpretable fuzzy models;

2) the generalization of the algorithms with error nonlinearities (i.e., Table I type algorithms) to the $\rho$-norms that are missing, even for linear in parameters models;

3) the development and deterministic mathematical analysis (in terms of filtering performance) of the $\rho$-norm algorithms [e.g., of type (2)] for an adaptive estimation of the parameters of an interpretable fuzzy model.

This paper is intended to provide the aforementioned studies in a unified manner. This is done via solving a constrained regularized nonlinear optimization problem in Section II. Section III provides the deterministic analysis of the algorithms with emphasis on filtering errors. Simulation studies are provided in Section IV followed by some remarks and, finally, the conclusion.

II. ADAPTIVE FUZZY ALGORITHMS

Sugeno-type fuzzy models are characterized by two types of parameters: consequents and antecedents. If we characterize the antecedents using a vector $\theta$ and consequents using a vector $\alpha$, then the output of a zero-order Takagi–Sugeno fuzzy model could be characterized as

$$F(x) = G^T(x, \theta)\alpha, \quad c\theta \geq h$$

where $G(\cdot)$ is a nonlinear function (which is defined by the shape of membership functions), and $c\theta \geq h$ is a matrix inequality to characterize the interpretability of the model. The details of (5) can be found in e.g., [19] as well as in the Appendix. A straightforward approach to the design of an adaptive fuzzy filter algorithm is to update, at time time $j$, the model parameters ($\alpha_{j-1}, \theta_{j-1}$) based on current data pair $(x(j), y(j))$, where we seek to decrease the loss term $|y(j) - G^T(x(j), \theta_j)\alpha_j|^2$; however, we do not want to make big changes in initial parameters ($\alpha_{j-1}, \theta_{j-1}$). That is

$$\alpha_j, \theta_j = \arg \min_{(\alpha, \theta, c\theta \geq h)} \left\{ \frac{1}{2} |y(j) - G^T(x(j), \theta)\alpha|^2 + \frac{\mu_i^{-1}}{2} \|\alpha - \alpha_{j-1}\|^2 + \frac{\mu_j^{-1}}{2} \|\theta - \theta_{j-1}\|^2 \right\}$$

where $\mu_i > 0$, $\mu_i, j > 0$ are the learning rates for antecedents and consequents, respectively, and $\|\cdot\|$ denotes the Euclidean norm (i.e., we write the 2-norm of a vector instead of $\|\cdot\|_2$ as $\|\cdot\|$). The terms $|\alpha - \alpha_{j-1}|^2$ and $|\theta - \theta_{j-1}|^2$ provide regularization to the adaptive estimation problem. To study the different adaptive algorithms in a unified framework, the following generalizations can be provided to the loss as well as regularization term.

1) The loss term is generalized using a function $L_j(\alpha, \theta)$.

Some examples of $L_j(\alpha, \theta)$ include

$$L_j(\alpha, \theta) = \begin{cases} |y(j) - G^T(x(j), \theta)\alpha| \\ \frac{1}{2} |y(j) - G^T(x(j), \theta)\alpha|^2 \\ \frac{q}{2} |y(j) - G^T(x(j), \theta)\alpha|^2 \\ + \frac{q}{2} |y(j) - G^T(x(j), \theta)\alpha|^4 \end{cases}$$

2) The regularization terms are generalized using Bregman divergences [9], [20]. The Bregman divergence $d_F(u, w)$ [21], which is associated with a strictly convex twice differentiable function $F$ from a subset of $R^K$ to $R$, is defined for $u, w \in R^K$, as follows:

$$d_F(u, w) = F(u) - F(w) - (u - w)^T f(w)$$

where $f = \nabla F$ denotes the gradient of $F$. Note that $d_F(u, w) \geq 0$, which is equal to zero only for $u = w$ and strictly convex in $u$. Some of the examples of Bregman divergences are as follows.

a) Bregman divergence associated to the squared $q$-norm: If we define $F(w) = (1/2)\|w\|^2_q$, then the corresponding Bregman divergence $d_q(u, w)$ is defined as

$$d_q(u, w) = \frac{1}{2}\|u\|_q^2 - \frac{1}{2}\|w\|_q^2 - (u - w)^T f(w)$$

where $f$ is given by (3). It is easy to see that for $q = 2$, we have $d_2(u, w) = (1/2)\|u - w\|^2$. 

Authorized licensed use limited to: Universitaetsbibl Rostock. Downloaded on December 1, 2009 at 03:11 from IEEE Xplore. Restrictions apply.
b) Relative entropy: For a vector $w = [w_1 \cdots w_K] \in R^K$ (with $w_i \geq 0$), if we define $F(w) = \sum_{i=1}^{K} (w_i \ln w_i - w_i)$, then the Bregman divergence is the unnormalized relative entropy:
\[ d_{RE}(u, w) = \sum_{i=1}^{K} \left( u_i \ln \frac{u_i}{w_i} - u_i + w_i \right). \]

Bregman divergences have been widely studied in learning and information theory (see, e.g., [22]–[28]). However, our idea is to replace the regularization terms $(\mu_j^{-1}/2)||\alpha - \alpha_{j-1}||^2$ and $(\mu_{q,j}^{-1}/2)||\theta - \theta_{j-1}||^2$ in (6) by the generalized terms $\mu_j^{-1}d_{G}(\alpha, \alpha_{j-1})$ and $\mu_{q,j}^{-1}d_{G}(\theta, \theta_{j-1})$, respectively. This approach, in the context of linear models, was introduced for deriving predictive algorithms in [29] and filtering algorithms in [9]. It is obvious that different choices of function $F$ result in the different filtering algorithms. Our particular concern in this text is to provide a $p$-norm generalization to the filtering algorithms.

For the $p$-norm ($2 \leq p < \infty$) generalization of the algorithms, we consider the Bregman divergence associated to the squared $q$-norm [i.e., $F(w) = (1/2)||w||_q^2$, where $q$ is dual to $p$ (i.e., $1/p + 1/q = 1$)].

In view of these generalizations, the adaptive fuzzy algorithms take the form
\[ (\alpha_j, \theta_j) = \arg \min_{(\alpha, \theta, \beta, \varphi) \geq 0} \left[ L_j(\alpha, \theta) + \mu_j^{-1}d_q(\alpha, \alpha_{j-1}) + \mu_{q,j}^{-1}d_q(\theta, \theta_{j-1}) \right]. \] (8)

For a particular choice $L_j(\alpha, \theta) = (1/2)||y(j) - G^T(x(j), \theta)||^2$ and $q = 2$, problem (8) reduces to (6). For a given value of $\theta$, we define
\[ \hat{\alpha}(\theta) = \arg \min_{\alpha} E_j(\hat{\alpha}(\theta), \theta) \]
\[ E_j(\alpha, \theta) = L_j(\alpha, \theta) + \mu_j^{-1}d_q(\alpha, \alpha_{j-1}) \]
so that the estimation problem (8) can be formulated as
\[ \theta_j = \arg \min_{\theta} \left[ E_j(\hat{\alpha}(\theta), \theta) + \mu_{q,j}^{-1}d_q(\theta, \theta_{j-1}), c\theta \geq h \right] \] (9)
\[ \alpha_j = \hat{\alpha}(\theta_j). \] (10)

Expressions (9) and (10) represent a generalized update rule for adaptive fuzzy filtering algorithms that can be particularized for a choice of $L_j(\alpha, \theta)$ and $q$. For any choice of $L_j(\alpha, \theta)$ listed in (7), $E_j(\alpha, \theta)$ is convex in $\alpha$ and, thus, could be minimized w.r.t. $\alpha$ by setting its gradient equal to zero. This results in
\[ \mu_j^{-1}f(\alpha) - \mu_j^{-1}f(\alpha_{j-1}) - \phi(y(j) - G^T(x(j), \theta)\alpha)G(x(j), \theta) = 0 \] (11)
where function $\phi$ is given as
\[ \phi(e) = \begin{cases} \text{sign}(e), & \text{for sign error} \\ e, & \text{for LMS} \\ e^3, & \text{for LMF} \\ ae + be^3, & \text{for LMMN}. \end{cases} \]

The minimizing solution $\hat{\alpha}$ must satisfy (11). Thus
\[ \hat{\alpha} = f^{-1}(f(\alpha_{j-1}) + \mu_j \phi(y(j) - G^T(x(j), \theta)\alpha)G(x(j), \theta)). \] (13)

For a given $\theta$, (13) is implicit in $\hat{\alpha}$ and could be solved numerically. It follows from (13) that for a sufficient small value of $\mu_j$, it is reasonable to approximate the term $G^T(x(j), \theta)\alpha_{j-1}$ on the right-hand side of (13) with the term $G^T(x(j), \theta)\alpha_{j-1}$, as has been done in [9] to obtain the explicit update. Thus, an approximate but in closed form, the solution of (13) is given as
\[ \hat{\alpha}(\theta) = f^{-1}(f(\alpha_{j-1}) + \mu_j \phi(y(j) - G^T(x(j), \theta)\alpha_{j-1})G(x(j), \theta)). \] (14)

Here, $\hat{\alpha}(\theta)$ has been written to indicate the dependence of the solution on $\theta$. Since $d_{q}(\theta, \theta_{j-1}) = (1/2)||\theta||^2 - (1/2)||\theta_{j-1}||^2 - \theta^T(\theta - \theta_{j-1})Tf(\theta_{j-1})^Tf(\theta_{j-1}),$ (9) is equivalent to
\[ \theta_j = \arg \min_{\theta} \left[ E_j(\hat{\alpha}(\theta), \theta) + \frac{\mu_j^{-1}}{2}||\theta||^2_q - \mu_{q,j}^{-1}\theta^Tf(\theta_{j-1}), c\theta \geq h \right] \] (15)
as the remaining terms are independent of $\theta$. There is no harm in adding a $\theta$-independent term in (15):
\[ \theta_j = \arg \min_{\theta} \left[ E_j(\hat{\alpha}(\theta), \theta) + \frac{\mu_j^{-1}}{2}||\theta||^2_q - \frac{\mu_{q,j}^{-1}}{2}||f(\theta_{j-1})||^2, c\theta \geq h \right]. \] (16)

For any $2 \leq p < \infty$, we have $1 < q \leq 2$, and thus, $||\theta||_q \geq ||\theta||_2$. This makes
\[ \frac{\mu_j^{-1}}{2}||\theta||^2_q - \frac{\mu_{q,j}^{-1}}{2}||f(\theta_{j-1})||^2 \frac{\mu_{q,j}^{-1}}{2}||f(\theta_{j-1})||^2 \geq \frac{\mu_{q,j}^{-1}}{2}||\theta||^2_q - \frac{\mu_{q,j}^{-1}}{2}||f(\theta_{j-1})||^2 \]
\[ = \frac{\mu_{q,j}^{-1}}{2}||\theta - f(\theta_{j-1})||^2. \] (18)

Solving the constrained nonlinear optimization problem (16), as we will see, becomes relatively easy by slightly modifying (decreasing) the level of regularization being provided in the estimation of $\theta_j$. The expression (17), i.e., last three terms of optimization problem (16), accounts for the regularization in the estimation of $\theta_j$. For a given value of $\mu_{q,j}$, a decrease in the level of regularization would occur via replacing the expression (17) in the optimization problem (16) by expression (18):
\[ \theta_j = \arg \min_{\theta} \left[ E_j(\hat{\alpha}(\theta), \theta) + \frac{\mu_j^{-1}}{2}||\theta - f(\theta_{j-1})||^2, c\theta \geq h \right]. \] (19)

For a viewpoint of an adaptive estimation of vector $\theta$, nothing goes against considering (19) instead of (16), since any
desired level of regularization could be still achieved via adjusting in (19) the value of free parameter $\mu_{\theta,j}$. The motivation of considering (19) is derived from the fact that it is possible to reformulate the estimation problem as a least-squares problem. To do so, define a vector

$$r(\theta) = \left[ \frac{1}{\sqrt{E_j(\alpha(\theta), \theta)}} \right]$$

where $E_j(\alpha(\theta), \theta) \geq 0$, and $\mu_{\theta,j} > 0$. Now, it is possible to rewrite (19) as

$$\theta_j = \arg \min_{\theta} [||r(\theta)||^2, c\theta \geq h]. \quad (20)$$

To compute $\theta_j$ recursively based on (20), we suggest following Gauss–Newton like algorithm:

$$\begin{align*}
\theta_j &= \theta_{j-1} + J^T(\theta_{j-1})
\end{align*}$$

$$s(\theta) = \arg \min_{s} [||r(\theta) + r(\theta)s||^2, cs \geq h - c\theta] \quad (22)$$

where $r(\theta)$ is the Jacobian matrix of vector $r$ w.r.t. $\theta$ computed by the method of finite differences. Fortunately, $r(\theta)$ is a full-rank matrix, since $\mu_{\theta,j} > 0$. The constrained linear least-squares problem (22) can be solved by transforming it first to a least distance programming (see [30] for details). Finally, (10) in view of (14) becomes

$$\alpha_j = f^{-1}(f(\alpha_{j-1}) + \mu_\alpha \phi(y(j) - G(x(j), \theta_j)\alpha_{j-1}) G(x(j), \theta_j)). \quad (23)$$

III. DETERMINISTIC ANALYSIS

We provide in this section a deterministic analysis of adaptive fuzzy filtering algorithms (21)–(23) in terms of filtering performance. For this, consider a fuzzy model that fits given input–output data \{$(x(j), y(j))$\}$_{j=0}^k$ according to

$$y(j) = G^T(x(j), \theta_j)\alpha^* + v_j \quad (24)$$

where \(\alpha^*\) is some true parameter vector (that is to be estimated), $\theta_j$ is given by (21), and $v_j$ accommodates any disturbance due to measurement noise, modeling errors, mismatch between $\theta_j$ and global minima of (20), and so on. We are interested in the analysis of estimating \(\alpha^*\) using (23) in the presence of a disturbance signal $v_j$. That is, we take $\alpha_j$ as an estimate of $\alpha^*$ at the $j$th time instant and try to calculate an upper bound on the filtering errors. In filtering setting, it is desired to estimate the quantity $G^T(x(j), \theta_j)\alpha^*$ using an adaptive model. The $\alpha_{j-1}$ is \textit{a priori} estimate of $\alpha^*$ at the $j$th time index, and thus, \textit{a priori} filtering error can be expressed as

$$e_{f,j} = G^T(x(j), \theta_j)\alpha^* - G^T(x(j), \theta_j)\alpha_{j-1}. \quad (26)$$

One would normally expect $|G^T(x(j), \theta_j)\alpha^* - G^T(x(j), \theta_j)\alpha_{j-1}|^2$ to be the performance measure of an algorithm. However, the squared error as a performance measure does not seem to be suitable for a uniform analysis of all the algorithms. We introduce a generalized performance measure $P_{\phi}(y, \bar{y})$ that is defined for scalars $y$ and $\bar{y}$ as

$$P_{\phi}(y, \bar{y}) = \int_{y}^{\bar{y}} (\phi(r) - \phi(y))dr \quad (25)$$

where $\phi$ is a continuous strictly increasing function with $\phi(0) = 0$. It can be easily seen that for $\phi(e) = e$, we have normal squared error i.e., $P_{\phi}(y, \bar{y}) = (\bar{y} - y)^2/2$. A different but integral-based loss function, called matching loss for a continuous, increasing transfer function $\Psi$, was considered in [31] and [32] for a single neuron model. The matching loss for $\psi$ was defined in [32] as

$$M_{\Psi}(y, \bar{y}) = \int_{y}^{\bar{y}} (\Psi(r) - y)rdr. \quad (26)$$

If we let $\Omega(r) = \int \phi(r)dr$, then

$$P_{\phi}(y, \bar{y}) = \int_{\Omega(y)}^{\Omega(\bar{y})} (\bar{y} - y)\phi(y). \quad (26)$$

Note that in definition (25), the continuous function $\phi$ is not an arbitrary function but its integral $\Omega(r) = \int \phi(r)dr$ must be a strictly convex function. The strictly increasing nature of $\phi(r)$ [i.e., strict convexity of $\Omega(r)$] enables us to assess the mismatch between $y$ and $\bar{y}$ using $P_{\phi}(y, \bar{y})$. Fig. 1 illustrates the physical meaning of $P_{\phi}(a, b)$. The value $P_{\phi}(a, b)$ is equal to the area of the shaded region in the figure. One could infer from Fig. 1 that a mismatch between $a$ and $b$ could be assessed via calculating the area of the shaded region [i.e., $P_{\phi}(a, b)$] provided the given function $\phi$ is strictly increasing. Usually, $P_{\phi}$ is not symmetric, i.e., $P_{\phi}(y, \bar{y}) \neq P_{\phi}(\bar{y}, y)$. One of such types of performance measures [i.e., $M_{\Psi}(y, \bar{y})$] was considered previously in [22].

In our analysis, we assess the instantaneous filtering error [i.e., the mismatch between $G^T(x(j), \theta_j)\alpha_{j-1}$ and $G^T(x(j), \theta_j)\alpha^* \} by calculating $P_{\phi}(G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*)$. For the case $\phi(e) = e$, we have $P_{\phi}(G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) = |e_{f,j}|^2/2$. The filtering performance of an algorithm, which is run from $j = 0$ to $j = k$, can be evaluated by calculating the sum

$$\sum_{j=0}^{k} P_{\phi}(G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*).$$

Similarly, the magnitudes of disturbances $v_j = y(j) - G^T(x(j), \theta_j)\alpha^*, j = 0, \ldots, k$ will be assessed by calculating
the mismatch between $y(j)$ and $G^T(x(j), \theta_j)\alpha^*$ as follows:

$$
\sum_{j=0}^{k} P_\delta (y(j), G^T(x(j), \theta_j)\alpha^*).
$$

Finally, the robustness of an algorithm (i.e., sensitivity of filtering errors toward disturbances) can be assessed by calculating an upper bound on the ratio, e.g., (29). The term $d_\delta(\alpha^*, \alpha_{-1})$ in the denominator of (29) assesses the disturbance due to a mismatch between initial guess $\alpha_{-1}$ and the true vector $\alpha^*$.

Lemma 1: Let $m$ be a scalar and $G_j \in \mathbb{R}^{K}$ such that $\alpha_j = f^{-1}(f(\alpha_{j-1}) + mG_j)$, and then

$$
d_\delta(\alpha_{j-1}, \alpha_j) \leq m^2 \frac{(p-1)}{2} ||G_j||^2_p.
$$

Proof: See [10, Lemma 4].

Lemma 2: If $\alpha_j$ and $\alpha_{j-1}$ are related via (23), then

$$
d_\delta(\alpha^*, \alpha_{j-1}) - d_\delta(\alpha^*, \alpha_j) + d_\delta(\alpha_{j-1}, \alpha_j)
$$

$$
= \mu_j \phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1}) \phi(y(j) - G^T(x(j), \theta_j)\alpha_j)\alpha_{j-1}^T G(x(j), \theta_j). \tag{30}
$$

Proof: The proof follows simply by using the definitions of $d_\delta(\alpha^*, \alpha_{j-1}), d_\delta(\alpha^*, \alpha_j)$, and $d_\delta(\alpha_{j-1}, \alpha_j)$.

In view of Lemma 1 and (23), we have

$$
d_\delta(\alpha_{j-1}, \alpha_j) \leq \mu_j^2 \frac{||\phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1})||^2}{2}
$$

$$
\times (p-1) \|G(x(j), \theta_j)\|_p^2. \tag{27}
$$

Theorem 1: The estimation algorithm (21)–(23) with $\phi$ being a continuous strictly increasing function and $\phi(0) = 0$, for any $2 \leq p < \infty$, with a learning rate

$$
0 < \mu_j \leq \frac{2P_\delta(y(j), G^T(x(j), \theta_j)\alpha_{j-1})}{\text{den}} \tag{28}
$$

where

$$
\text{den} = \phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1}) (p-1)
$$

$$
\times ||\phi(y(j)) - \phi(G^T(x(j), \theta_j)\alpha_{j-1})|| \|G(x(j), \theta_j)\|_p^2
$$

achieves an upper bound on filtering errors such that

$$
\sum_{j=0}^{k} \mu_j^2 P_\delta(G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha_j) + d_\delta(\alpha^*, \alpha_{-1}) \leq 1 \tag{29}
$$

where $q$ is dual to $p$, and $\mu_j^2 > 0$ is given as

$$
\mu_j^2 = \mu_j \frac{\phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1})}{\phi(y(j)) - \phi(G^T(x(j), \theta_j)\alpha_{j-1})}. \tag{30}
$$

Here, we assume that $y(j) \neq G^T(x(j), \theta_j)\alpha_{j-1}$, since there is no update of parameters (i.e., $\alpha_j = \alpha_{j-1}$) if $y(j) = G^T(x(j), \theta_j)\alpha_{j-1}$.

Proof: Define $\delta_j = d_\delta(\alpha^*, \alpha_{j-1}) - d_\delta(\alpha^*, \alpha_j)$ and using Lemma 2, we have

$$
\delta_j = \mu_j \phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1}) (\alpha^* - \alpha_{j-1})^T G(x(j), \theta_j)
$$

$$
- d_\delta(\alpha_{j-1}, \alpha_j). \tag{31}
$$

Using (27), we have

$$
\delta_j \geq \mu_j \phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1}) (\alpha^* - \alpha_{j-1})^T G(x(j), \theta_j)
$$

$$
- \frac{p-1}{2} \mu_j^2 \phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1})^T \|G(x(j), \theta_j)\|_p^2.
$$

That is

$$
\delta_j \geq \mu_j^2 \phi(y(j)) - \phi(G^T(x(j), \theta_j)\alpha_{j-1})(\alpha^* - \alpha_{j-1})^T G(x(j), \theta_j)
$$

$$
- \frac{p-1}{2} \mu_j \phi(y(j) - G^T(x(j), \theta_j)\alpha_{j-1}) \mu_j^2
$$

$$
\times \|\phi(y(j)) - \phi(G^T(x(j), \theta_j)\alpha_{j-1})\| \|G(x(j), \theta_j)\|_p^2.
$$

Since $\mu_j$ satisfies (28), the previous inequality reduces to

$$
\delta_j \geq \mu_j^2 \phi(y(j)) - \phi(G^T(x(j), \theta_j)\alpha_{j-1})(\alpha^* - \alpha_{j-1})^T G(x(j), \theta_j)
$$

$$
- \mu_j^2 P_\delta(y(j), G^T(x(j), \theta_j)\alpha_{j-1}). \tag{32}
$$

It can be verified using definition (26) that

$$
\phi(y(j)) - \phi(G^T(x(j), \theta_j)\alpha_{j-1})(\alpha^* - \alpha_{j-1})^T G(x(j), \theta_j)
$$

$$
= P_\delta(G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*)
$$

$$
- P_\delta(y(j), G^T(x(j), \theta_j)\alpha_j) + P_\delta(y(j), G^T(x(j), \theta_j)\alpha_{j-1})
$$

and thus, inequality (32) is further reduced to

$$
\delta_j \geq \mu_j^2 P_\delta(G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*)
$$

$$
- \mu_j^2 P_\delta(y(j), G^T(x(j), \theta_j)\alpha_j).
$$

In addition

$$
\sum_{j=0}^{k} \delta_j = d_\delta(\alpha^*, \alpha_{-1}) - d_\delta(\alpha^*, \alpha_k)
$$

$$
\leq d_\delta(\alpha^*, \alpha_{-1}), \quad \text{since } d_\delta(\alpha^*, \alpha_k) \geq 0
$$

resulting in

$$
d_\delta(\alpha^*, \alpha_{-1}) \geq \sum_{j=0}^{k} \mu_j^2 P_\delta(G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*)
$$

$$
- \sum_{j=0}^{k} \mu_j^2 P_\delta(y(j), G^T(x(j), \theta_j)\alpha_j)
$$

from which the inequality (29) follows.

Following inferences could be immediately made from Theorem 1.

1) For the special case $\phi(e) = e$ and taking $\alpha_{-1} = 0$, the results of Theorem 1 are modified as follows:

$$
\sum_{j=0}^{k} \mu_j ||G^T(x(j), \theta_j)\alpha_{j-1} - G^T(x(j), \theta_j)\alpha^*||^2 \leq 1
$$

where

$$
\mu_j \leq \frac{1}{(p-1) \|G(x(j), \theta_j)\|_p^2}.
$$
Choosing
\[ \mu_j = \frac{1}{(p - 1)U_p^2}, \quad \text{where } U_p \geq \|G(x(j), \theta_j)\|_p \]
we get
\[ \sum_{j=0}^{k} \|G^T(x(j), \theta_j)\alpha_{j-1} - G^T(x(j), \theta_j)\alpha^*\|^2 \]
\[ \leq \sum_{j=0}^{k} \|y(j) - G^T(x(j), \theta_j)\alpha^*\|^2 + (p - 1)U_p^2\|\alpha^*\|_q^2 \]
which is formally equivalent to [9, Th. 2]

2) Inequality (29) illustrates the robustness property in the sense that if, for the given positive values \( \{\mu_j\}_{j=0}^{K} \), the disturbances \( \{v_j\}_{j=0}^{K} \) are small, i.e.,
\[ \sum_{j=0}^{k} \mu_j^a P_\phi (y(j), G^T(x(j), \theta_j)\alpha*) \]
is small, then the filtering errors
\[ \sum_{j=0}^{k} P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) \]
remain small. The positive value \( \mu_j^a \) represents a weight given to the \( j \)th data pair in the summation.

3) If we define an upper bound on the disturbance signal as
\[ v_{\phi, max} = \max_{j} P_\phi (y(j), G^T(x(j), \theta_j)\alpha^*) \]
then it follows from (29) that
\[ \sum_{j=0}^{k} \mu_j^a P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) \]
\[ \leq (k + 1) \mu_{\max} v_{\phi, max} + d_q(\alpha^*, \alpha_{-1}) \]
where \( \mu_{\max} = \max_{j} \mu_j^a \). That is
\[ \frac{1}{k + 1} \sum_{j=0}^{k} \mu_j^a P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) \]
\[ \leq \mu_{\max} v_{\phi, max} + d_q(\alpha^*, \alpha_{-1}) \]
As \( d_q(\alpha^*, \alpha_{-1}) \) is finite, we have
\[ \frac{1}{k + 1} \sum_{j=0}^{k} \mu_j^a P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) \]
\[ \leq \mu_{\max} v_{\phi, max}, \quad \text{when } k \to \infty. \]
Inequality (33) shows the stability of the algorithm against disturbance \( v_j \) in the sense that if disturbance signal \( P_\phi (y(j), G^T(x(j), \theta_j)\alpha^*) \) is bounded (i.e., \( v_{\phi, max} \) is finite), then the average value of filtering errors assessed as
\[ \frac{1}{k + 1} \sum_{j=0}^{k} \mu_j^a P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) \]
remains bounded.

4) In the ideal case of zero disturbances (i.e., \( v_j = 0 \)), we have \( P_\phi (y(j), G^T(x(j), \theta_j)\alpha^*) = 0 \), and
\[ \sum_{j=0}^{k} \mu_j^a P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) \]
\[ \leq d_q(\alpha^*, \alpha_{-1}). \]
Since \( d_q(\alpha^*, \alpha_{-1}) \) is finite and \( \mu_j^a P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) \geq 0 \), there must exist a sufficient large index \( T \) such that
\[ \sum_{j=0}^{T} \mu_j^a P_\phi (G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^*) = 0. \]
In other words
\[ G^T(x(j), \theta_j)\alpha_{j-1} = G^T(x(j), \theta_j)\alpha^*, \quad \forall j \geq T \]
since \( \mu_j^a > 0 \). This shows the convergence of the algorithm at time index \( T \) toward true parameters.

Theorem 1 is an important result due to the generality of function \( \phi \), and thus, offers a possibility of studying, in a unified framework, different fuzzy adaptive algorithms corresponding to the different choices of continuous strictly increasing function \( \phi \). Table II provides a few examples of \( \phi \) (plotted in Fig. 2), leading to the different \( p \)-norms algorithms listed in the table as \( A_{1,p}, A_{2,p}, \ldots, p \geq 2 \), perform better than the commonly used gradient-descent algorithm.

2) For the estimation of linear parameters (keeping membership functions fixed), LMS is the standard algorithm known for its simplicity and robustness. This study provides a family of algorithms characterized by \( \phi \) and \( p \). It will be shown that the \( p \)-norm generalization of the algorithms corresponding to the different choices of \( \phi \) may achieve better performance than the standard LMS algorithm.

3) We will show the robustness and convergence of the filtering algorithms.

4) For a given value of \( p \), the algorithms corresponding to the different choices of \( \phi \) may prove being better than the standard choice \( \phi(e) = e \) (i.e., squared error loss term).

A. First Example

Consider the problem of filtering noise from a chaotic time series. The time series is generated by simulating the
few examples of adaptive fuzzy filtering algorithms

The estimation algorithm (21-23) for a learning rate

\[ \mu_j \leq 2P_\phi \left( y(j), G^T(x(j), \theta_j)\alpha_{j-1} \right) \]

\[ \sum_{k} \mu_j^k P_\phi \left( G^T(x(j), \theta_j)\alpha_{j-1}, G^T(x(j), \theta_j)\alpha^* \right) \]

achieves

\[ \sum_{k} \mu_j^k P_\phi \left( y(j), G^T(x(j), \theta_j)\alpha^* \right) + d_q(\alpha^*, \alpha_1) \leq 1, \text{ where } \mu_j^0 = \frac{\phi \left( y(j) - G^T(x(j), \theta_j)\alpha_{j-1} \right)}{\phi(\hat{y}(j)) - \phi \left( G^T(x(j), \theta_j)\alpha_{j-1} \right)}\]

The fourth-order Runge–Kutta method was used for the simulation of time series, and 500 input–output data pairs, starting from \( t = 124 \) to \( t = 623 \), were extracted for an adaptive estimation of parameters \((\alpha^*, \theta^*)\) using different algorithms. If \((\alpha_{t-1}, \theta_{t-1})\) denotes the a priori estimation at time \( t \), then filtering error of an algorithm is defined as

\[ e_{f,t} = x(t) - \hat{y}(t) \]

where

\[ \hat{y}(t) = G^T \left( [x(t-24) \ x(t-18) \ x(t-12) \ x(t-6)]^T, \theta_{t-1} \right) \alpha_{t-1}. \]

We consider a total of 30 different algorithms i.e., \( A_{1,p}, \ldots, A_{5,p} \) for \( p = 2, 2.2, 2.4, 2.6, 2.8, 3 \), running from \( t = 124 \) to \( t = 623 \), for the prediction of the desired value \( x(t) \). We choose, for an example sake, the trapezoidal type of membership functions (defined by 36) such that the number of membership functions assigned to each of the four inputs [i.e., \( x(t-24), x(t-18), x(t-12), x(t-6) \)] is equal to 3. The initial guess about model parameters was taken as

\[ \theta_{123} = \left[ \begin{array}{c} 0.3 \ T \\ 0.6 \ T \\ 0.9 \ T \\ 1.2 \ T \end{array} \right] \]

where \( \alpha_{123} = [0]_{81 \times 1} \). The matrix \( c \) and vector \( h \) are chosen in such a way that the two consecutive knots must remain separated at least by a distance of 0.001 during the recursions of the algorithms.

The gradient-descent algorithm (1), if intuitively applied to the fuzzy model (5), needs following considerations.

1) The antecedent parameters of the fuzzy model should be estimated with the learning rate \( \mu_\theta \), while the consequent

Fig. 2. Different examples of function \( \phi \).

Mackey–Glass differential delay equation

\[ \frac{dx}{dt} = \frac{0.2x(t-17)}{1 + x^10(t-17)} - 0.1x(t) \]

\[ y(t) = x(t) + n(t) \]

where \( x(0) = 1.2, x(t) = 0, \) for \( t < 0, \) and \( n(t) \) is a random noise chosen from a uniform distribution on the interval \([-0.2, 0.2]\). The aim is to filter the noise \( n(t) \) from \( y(t) \) to estimate \( x(t) \) by using a set of past values i.e., \( [x(t-24), x(t-18), x(t-12), x(t-6)] \).

Assume that there exists an ideal fuzzy model characterized by \((\alpha^*, \theta^*)\) that models the relationship between input vector \( [x(t-24), x(t-18), x(t-12), x(t-6)]^T \) and output \( x(t) \). That is

\[ x(t) = G^T \left( [x(t-24) \ x(t-18) \ x(t-12) \ x(t-6)]^T, \theta^* \right) \alpha^* \]

\[ y(t) = G^T \left( [x(t-24) \ x(t-18) \ x(t-12) \ x(t-6)]^T, \theta^* \right) \alpha^* + n(t) \]
parameters with the learning rate $\mu$. That is

$$
\begin{bmatrix}
\theta_j \\
\alpha_j
\end{bmatrix} = \begin{bmatrix}
\theta_{j-1} \\
\alpha_{j-1}
\end{bmatrix} - \begin{bmatrix}
\mu_\theta & 0 \\
0 & \mu
\end{bmatrix} \begin{bmatrix}
\frac{\partial E_r(\theta, \theta, j)}{\partial \theta} \\
\frac{\partial E_r(\theta, \theta, j)}{\partial \alpha}
\end{bmatrix}_{\theta_{j-1}, \alpha_{j-1}}.
$$

$$
E_r(\alpha, \theta, j) = \frac{1}{2} |y(j) - G^T(x(j), \theta)\alpha|^2.
$$

Introducing the notation $\Theta_j = [\theta_j^T \alpha_j^T]^T$, the gradient-descent update takes the form

$$
\Theta_j = \Theta_{j-1} - \begin{bmatrix}
\mu_\theta & 0 \\
0 & \mu
\end{bmatrix} \left( \frac{\partial E_r(\Theta, j)}{\partial \Theta} \right)_{\Theta_{j-1}}.
$$

Here, $\mu$ may or may not be equal to $\mu_\theta$. In general, $\mu_\theta$ should be less than $\mu$ to avoid any oscillations of the estimated parameters.

2) During the gradient-descent estimation of membership functions parameters, in the presence of disturbances, the knots (elements of vector $\theta$) may attempt to come close (or even cross) one another. That is, inequalities (38) and (39) do not hold good and, thus, result in a loss of interpretability and estimation performance. For a better performance of gradient-descent, the knots must be prevented from crossing one another by modifying the estimation scheme as

$$
\Theta_j = \begin{cases}
\Theta_{j-1} - \begin{bmatrix}
\mu_\theta & 0 \\
0 & \mu
\end{bmatrix} \left( \frac{\partial E_r(\Theta, j)}{\partial \Theta} \right)_{\Theta_{j-1}}, & \text{if } \epsilon \theta_j \geq h \\
\Theta_{j-1}, & \text{otherwise.}
\end{cases}
$$

Each of the aforementioned algorithms and gradient-descent algorithm (34) is run at $\mu = \mu_\theta = 0.9$. The filtering performance of each algorithm is assessed via computing energy of the filtering error signal $e_{f,t}$. The energy of a signal is equal to the squared $L_2$-norm. The energy of filtering error signal $e_{f,t}$, from $t = 124$ to $t = 623$, is defined as

$$
\sum_{t=124}^{623} |e_{f,t}|^2.
$$

A higher energy of filtering errors means the higher magnitudes of filtering errors and, thus, a poor performance of the filtering algorithm.

Fig. 3 compares the different algorithms by plotting their filtering errors energies at different values of $p$. As seen from Fig. 3, all the algorithms $A_{1,p}, \ldots, A_{5,p}$ for $p = 2, 2.2, 2.4, 2.6, 2.8, 3$ perform better than the gradient-descent since gradient-descent method is associated to a higher energy of the filtering errors. As an illustration, Fig. 4 shows the time plot of the absolute filtering error $|e_{f,t}|$ for gradient-descent and the algorithm $A_{4,p}$ at $p = 2.4$.

B. Second Example

Now, we consider a linear fuzzy model (membership functions being fixed)

$$
y(j) = [\mu_{A_{11}}(x(j)) \mu_{A_{21}}(x(j)) \mu_{A_{31}}(x(j)) \mu_{A_{41}}(x(j))] \alpha^* + v_j
$$

where $\alpha^* = [0.25 -0.5 1 -0.3]^T$, $x(j)$ takes random values from a uniform distribution on $[-1, 1]$, $v_j$ is a random noise chosen from a uniform distribution on the interval $[-0.2, 0.2]$, and the membership functions are defined by (37) taking $\theta = (-1, -0.3, 0.3, 1)$.

Algorithm (23) was employed to estimate $\alpha^*$ for different choices of $\phi$ (listed in Table II) at $p = 3$ (as an example of $p > 2$). The initial guess is taken as $\alpha_{-1} = 0$. For comparison, the LMS algorithm is also simulated. Note that the LMS algorithm is just a particularization of (23) for $p = 2$ and $\phi(e) = e$. That is,
A2.2 in Table II, in the context of linear estimation, is the LMS algorithm. The performance of an algorithm was evaluated by calculating the instantaneous a priori error in the estimation of $\alpha^*$ as $\|\alpha^* - \alpha_{j-1}\|^2$. The learning rate $\mu_j$ for each algorithm, including LMS, is chosen according to (28) as follows:

$$
\mu_j = \frac{2P_0(y(j), G_j^T \alpha_{j-1})}{\phi(y(j)-G_j^T \alpha_{j-1})\{\phi(y(j))\phi(G_j^T \alpha_{j-1})\}(p-1)\|G_j\|^2_p}
$$

where

$$
G_j = [\mu_{A_{11}}(x(j)) \mu_{A_{12}}(x(j)) \mu_{A_{13}}(x(j)) \mu_{A_{14}}(x(j))]^T.
$$

For an assessment of the expected error values i.e., $E[\|\alpha^* - \alpha_{j-1}\|^2]$, 500 independent experiments have been performed. The 500 independent time plots of values $\|\alpha^* - \alpha_{j-1}\|^2$ have been averaged to obtain the time plot of $E[\|\alpha^* - \alpha_{j-1}\|^2]$, shown in Fig. 5. The better performance of algorithms $(A_{1,3}, A_{2,3}, A_{3,3}, A_{4,3}, A_{5,3})$ than the LMS can be seen in Fig. 5.

This example indicates that the $p$-norm generalization of the algorithms makes a sense since $A_{2,p}$ algorithm for $p = 3$ (a value of $p > 2$) proved being better than the $p = 2$ case (i.e., LMS).

C. Robustness and Convergence

To study the robustness properties of the algorithms, we investigate the sensitivity of the filtering errors toward disturbances. To make this more precise, we plot the curve between disturbance energy and filtering errors energy and analyze the curve. In the aforementioned example (i.e., second example), the energy of filtering errors will be defined as

$$
E_f(j) = \sum_{i=0}^{j} [G_i^T \alpha^* - G_i^T \alpha_{i-1}]^2
$$

and the total energy of disturbances as

$$
E_d(j) = \sum_{i=0}^{j} |v_i|^2 + \|\alpha^* - \alpha_{i-1}\|^2
$$

where the term $\|\alpha^* - \alpha_{i-1}\|^2$ accounts for the disturbance due to a mismatch between the initial guess and true parameters. Fig. 6 shows the plot between the values $\{E_d(j)\}_{j=0}^{1000}$ and $\{E_f(j)\}_{j=0}^{1000}$ for each of the algorithms $(A_{1,3}, A_{2,3}, A_{3,3}, A_{4,3}, A_{5,3})$. The curves for $(A_{2,3}, A_{4,3}, A_{5,3})$ are not distinguishable. The curves in Fig. 6 have been averaged over 100 independent experiments. The initial guess $\alpha_{i-1}$ is chosen in each experiment randomly from a uniform distribution on $[-1,1]$. The curves in Fig. 6 show the robustness of the algorithms in the sense that a small energy of disturbances does not lead to a large energy of filtering errors. Thus, if the disturbances are bounded, then the filtering errors also remain bounded [i.e., bounded-input bounded-output (BIBO) stability].

To verify the convergence properties of the algorithms, the plots of Fig. 5 are redrawn via running the algorithms $(A_{1,3}, A_{2,3}, A_{3,3}, A_{4,3}, A_{5,3})$ in an ideal case of zero disturbances (i.e., $v_j = 0$). Fig. 7 shows, in this case, the convergence of the algorithms toward true parameters.

D. Third Example

Our third example has been taken from [9], where at time $t$, a signal $y_t$ is transmitted over a noisy channel. The recipient is required to estimate the sent signal $y_t$ out of the actually observed signal

$$
r_t = \sum_{i=0}^{k-1} u_{i+1} y_{t-i} + v_t
$$

where $v_t$ is zero-mean Gaussian noise with a signal-to-noise ratio of 10 dB. The signal is estimated using an adaptive filter

$$
\hat{y}_t = G_t^T \alpha_{t-1}, \quad G_t = [r_{t-m} \cdots r_t \cdots r_{t+m}] \in R^{2m+1}
$$
where $\alpha_i \in R^{2m+1}$ are the estimated filter parameters at time $t$. We take the values $k = 10$ and $m = 15$ as in [9]. The transmitted signal $y_t$ is chosen to be zero-mean Gaussian with unit variance; however, $y_t$ was a binary signal in [9] (which is obviously not quite the same as our framework). The vector $u \in R^k$, describing the channel, was chosen in [9] in two different manners.

1) In the first case, $u$ is chosen from a Gaussian distribution with unit variance and then normalized to make $\|u\| = 1$.
2) In the second case, $u_k = s_k e^{\omega i}$, where $s_k \in \{-1, 1\}$, $\omega \in \{−10, 10\}$ are distributed uniformly and then normalized to make $\|u\| = 1$.

Algorithm (23) was used to estimate the filter parameters taking different choices of $\phi$ at $p = 2.5$ in the first case. However, the second case (with $u$ being “sparse”), as discussed in [9], favors fairly large value of $p$ [i.e., $p = 2\ln(2m + 1)$]. The learning rate is chosen to be equal to (28) as in the previous example. The instantaneous filtering error is defined as

$$e_{f,t} = y_t - G_t^T \alpha_{t-1}.$$ 

The performance of different filtering algorithms is assessed by calculating the root mean square of filtering errors:

$$\text{RMSFE}(t) = \left( \frac{1}{t+1} \sum_{i=0}^{t} e_{f,i}^2 \right)^{1/2}.$$ 

The time plots of root mean square of filtering errors, averaged over 100 independent experiments, are shown in Fig. 8. Fig. 8(a) shows the faster convergence of algorithm $A_{1,p}$ than $A_{2,p}$ while Fig. 8(b) shows the faster convergence of $A_{3,p}$ than $A_{2,p}$. This indicates that the algorithms corresponding to the different choices of $\phi$ (i.e., $A_{1,p}$, $A_{3,p}$, etc.) may prove being better in some sense than the standard choice $\phi(e) = e$ (i.e., algorithm $A_{2,p}$). Hence, the proposed framework that offers the possibility of developing filtering algorithms corresponding to the different choices of $\phi$ is a useful tool.

The provided simulation studies clearly indicate the potential of our approach in adaptive filtering. The first example shows the better filtering performance of our approach than the most commonly used gradient-descent algorithm for estimating the parameters of nonlinear neural/fuzzy models. Some hybrid methods, e.g., clustering for membership functions and RLS algorithm for consequents, have been suggested in the literature for an online identification of the fuzzy models. It is a well-known fact that RLS optimizes the average (expected) performance under some statistical assumptions, while LMS optimizes the worst-case performance. Since our algorithms are the generalized versions of LMS and possess LMS-like robustness properties (as indicated in Theorem 1), we do not compare RLS algorithm with our algorithms. Moreover, for a fair comparison, RLS must be generalized as we generalized LMS in our analysis. More will be said on the generalization of the RLS algorithm in Section V.

V. SOME REMARKS

This text outlines an approach to adaptive fuzzy filtering in a broad sense, and thus, several related studies could be made. In particular, we would like to mention the following.

1) We studied the adaptive filtering problem using a zero-order Takagi–Sugeno fuzzy model due to the simplicity. However, the approach can be applied to any semilinear model with linear inequality constraints, e.g., first-order Takagi–Sugeno fuzzy models, radial basis function (RBF) neural networks, B-spline models, etc. The approach is valid for any model characterized by parameters set $\Theta$ such that $\Theta = \Theta_t + \Theta_n$.

$$y = G^T(x, \Theta_n)\Theta_t, \quad e\Theta_n \geq h.$$ 

2) We have considered for the $p$-norm generalization of the algorithms the Bregman divergence associated to the squared $q$-norm. Another important example of Bregman divergences is the relative entropy, which are defined between vectors $u = [u_1 \cdots u_K]^T$ and $w = [w_1 \cdots w_K]^T$ (assuming $u_i, w_i \geq 0$ and $\sum_{i=1}^{K} u_i = \sum_{i=1}^{K} w_i = 1$) as follows:

$$d_{\text{REL}}(u, w) = \sum_{i=1}^{K} u_i \ln \frac{u_i}{w_i}.$$ 

The relative entropy is the Bregman divergences $d_F(u, w)$ for $F(u) = \sum_{i=1}^{K} (u_i - u_i^*)$. It is possible to derive and analyze different exponentiated gradient [29] type fuzzy filtering algorithms via using relative entropy as regularizer and following the same approach. However, some additional efforts are required to handle the unity sum constraint on the vectors.

3) We arrived at the explicit update form (14) by making an approximation. A natural question that arises is if any improvement in the filtering performance (assessed in Theorem 1) could be made by solving (13) numerically instead of approximating. An upper bound on the filtering errors (as in Theorem 1) could be calculated in this case too; however, we would then be considering the a posteriori filtering errors

$$\sum_{j=0}^{p} P_{\alpha_{j}}(G_j^T(x(j), \theta_j)\alpha_j, G_j^T(x(j), \theta_j)\alpha^*)$$.
4) Our emphasis in this study was on filtering. In machine learning literature, one is normally interested in the prediction performance of such algorithms [10], [33], [34]. The presented algorithms could be evaluated, in a similar manner as Theorem 1, by calculating an upper bound on the prediction errors \( \sum_{j=0}^{k} P_{j} \left( G^{T} (x(j), \theta_{j}) \alpha_{j-1}, y(j) \right) \).

5) This study offers the possibility of developing and analyzing new fuzzy filtering algorithms by defining a continuous strictly increasing function \( \phi \). An interesting research direction is to optimize the function \( \phi \) for the problem at hand. For example, in the context of linear estimation, an expression for the optimum function that minimizes the steady-state mean-square error is derived in [8].

6) Other than the LMS, the recursive least squares (RLS) is a well-known algorithm that optimizes the average performance under some stochastic assumptions on the signals. The deterministic interpretation of RLS algorithm is that it solves a regularized least-squares problem

\[
\mathbf{w}_{k} = \arg \min_{\mathbf{w}^*} \left[ \sum_{j=0}^{k} \left| y(j) - G^{T} w^* \right|^2 + \mu^{-1} \left\| w^* \right\|^2 \right].
\]

Our future study is concerned with the generalization of RLS algorithm, in the context of interpretable fuzzy models, based on the solution of following regularized least-squares problem

\[
(\alpha_{k}, \theta_{k}) = \arg \min_{(\alpha^*, \theta^*, \alpha^* \theta^* \geq h)} J_k
\]

\[
J_k = \sum_{j=0}^{k} L_j(\alpha^*, \theta^*) + \mu^{-1} d_{\theta}(\alpha^*, \alpha_{-1}) + \mu_{\theta}^{-1} d_{\theta}(\theta^*, \theta_{-1})
\]

where some examples of loss term \( L_j(\alpha, \theta) \) are provided in Table II.

VI. CONCLUSION

Much work has been done on applying fuzzy models in function approximation and classification tasks. We feel that many real-world applications (e.g., in chemistry [35], biomedical engineering [2], etc.) require the filtering of uncertainties from the experimental data. The nonlinear fuzzy models by virtue of membership functions are more promising than the classical linear models. Therefore, it is essential to study the adaptive fuzzy filtering algorithms. Adaptive filtering theory for linear models has been well developed in the literature; however, its extension to the fuzzy models is complicated by the nonlinearity of membership functions and the interpretability constraints. The contribution of the manuscript (summarized in Theorem 1, its inferences, and Table II) is to provide a mathematical framework that allows the development and an analysis of the adaptive fuzzy filtering algorithms. The power of our approach is the flexibility of designing the algorithms based on the choice of function \( \phi \) and the parameter \( p \). The derived filtering algorithms have the desired properties of robustness, stability, and convergence. This paper is an attempt to provide a deterministic approach to study the adaptive fuzzy filtering algorithms, and the study opens many research directions, as discussed in Section V. Future work involves the study of fuzzy filtering algorithms derived using relative entropy as a regularizer, optimizing the function \( \phi \) for the problem at hand, and a generalization of RLS algorithm to \( p \)-norm in the context of interpretable fuzzy models.

APPENDIX

TAKAGI–SUGENO FUZZY MODEL

Let us consider an explicit mathematical formulation of a Sugeno-type fuzzy inference system that assigns to each crisp value (vector) in input space a crisp value in output space. Consider a Sugeno fuzzy inference system \( F_s: X \to Y \), mapping \( n \)-dimensional input space \( (X = X_1 \times X_2 \times \cdots \times X_n) \) to
one-dimensional real line, consisting of $K$ different rules. The $i$th rule is in form

$$
\text{if } x_i \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \cdots \text{ and } x_n \text{ is } A_{in} \text{ then } y = c_i
$$

for all $i = 1, 2, \ldots, K$, where $A_{i1}, A_{i2}, \ldots, A_{in}$ are nonempty fuzzy subsets of $X_1, X_2, \ldots, X_n$, respectively, such that the membership functions $\mu_{A_{ij}}: X_j \rightarrow [0,1]$ fulfill $\sum_{i=1}^{K} \prod_{j=1}^{n} \mu_{A_{ij}}(x_j) > 0$ for all $x_j \in X_j$, and values $c_1, \ldots, c_K$ are real numbers. The different rules, by using “product” as conjunction operator, can be aggregated as

$$
F_i(x_1, x_2, \ldots, x_n) = \frac{\sum_{i=1}^{K} c_i \prod_{j=1}^{n} \mu_{A_{ij}}(x_j)}{\sum_{i=1}^{K} \prod_{j=1}^{n} \mu_{A_{ij}}(x_j)} \tag{35}
$$

Let us define a real vector $\theta$ such that the membership functions can be constructed from the elements of vector $\theta$. To illustrate the construction of membership functions based on knot vector ($\theta$), consider the following examples.

1) Trapezoidal membership functions: Let

$$
\theta = (a_1, t_1^1, \ldots, t_1^{P_1}, b_1, \ldots, a_n, t_n^1, \ldots, t_n^{P_n}, b_n)
$$

such that for the $i$th input ($x_i \in [a_i, b_i]$), $a_i < t_i^1 < \cdots < t_i^{P_i} < b_i$ holds $\forall i = 1, \ldots, n$. Now, $P_i$ trapezoidal membership functions for the $i$th input ($\mu_{A_{i1}}, \mu_{A_{i2}}, \ldots, \mu_{A_{iP_i}}$) can be defined as

$$
\mu_{A_{i1}}(x_i, \theta) = \begin{cases} 
1, & \text{if } x_i \in [a_i, t_i^1] \\
-\alpha x_i + t_i^1, & \text{if } x_i \in [t_i^1, t_i^2] \\
\alpha x_i - t_i^1, & \text{otherwise}
\end{cases}
$$

$$
\mu_{A_{ij}}(x_i, \theta) = \begin{cases} 
\frac{x_i - t_i^{j-3}}{t_i^{j-2} - t_i^{j-3}}, & \text{if } x_i \in [t_i^{j-3}, t_i^{j-2}] \\
1, & \text{if } x_i \in [t_i^{j-2}, t_i^{j-1}] \\
-\alpha x_i + t_i^{j-1}, & \text{if } x_i \in [t_i^{j-1}, t_i^j] \\
0, & \text{otherwise}
\end{cases}, \quad 2 \leq j \leq P_i
$$

$$
\mu_{A_{Pi}}(x_i, \theta) = \begin{cases} 
\frac{x_i - t_i^{P_i-3}}{t_i^{P_i-2} - t_i^{P_i-3}}, & \text{if } x_i \in [t_i^{P_i-3}, t_i^{P_i-2}] \\
1, & \text{if } x_i \in [t_i^{P_i-2}, b_i] \\
0, & \text{otherwise}
\end{cases}
\tag{36}
$$

2) One-dimensional clustering-criterion-based membership functions: Let

$$
\theta = (a_1, t_1^1, \ldots, t_1^{P_1}, b_1, \ldots, a_n, t_n^1, \ldots, t_n^{P_n}, b_n)
$$

such that for the $i$th input, $a_i < t_i^1 < \cdots < t_i^{P_i} < b_i$ holds for all $i = 1, \ldots, n$. Now, consider the problem of assigning two different memberships (say $\mu_{A_{i1}}$ and $\mu_{A_{i2}}$) to a point $x_i$ such that $a_i < x_i < t_i^1$, based on the following clustering criterion:

$$
[\mu_{A_{i1}}(x_i), \mu_{A_{i2}}(x_i)] = \arg \min_{u_1, u_2} \left[ u_1^2(x_i - a_i)^2 + u_2^2(x_i - t_i^1)^2 \right] \text{, } u_1 + u_2 = 1.
$$

This results in

$$
\mu_{A_{i1}}(x_i) = \frac{(x_i - t_i^1)^2}{(x_i - a_i)^2 + (x_i - t_i^1)^2}, \quad x_i \in [a_i, t_i^1]
$$

$$
\mu_{A_{i2}}(x_i) = \frac{(x_i - a_i)^2}{(x_i - a_i)^2 + (x_i - t_i^1)^2}, \quad x_i \in [a_i, t_i^1]
$$

Thus, for the $i$th input, $P_i$ membership functions ($\mu_{A_{i1}}, \mu_{A_{i2}}, \ldots, \mu_{A_{iP_i}}$) can be defined as

$$
\mu_{A_{i1}}(x_i, \theta) = \begin{cases} 
1, & x_i \leq a_i \\
\frac{(x_i - t_i^1)^2}{(x_i - a_i)^2 + (x_i - t_i^1)^2}, & x_i \in [a_i, t_i^1] \\
0, & \text{otherwise}
\end{cases}
$$

$$
\mu_{A_{ij}}(x_i, \theta) = \begin{cases} 
\frac{(x_i - a_i)^2}{(x_i - a_i)^2 + (x_i - t_i^1)^2}, & x_i \in [a_i, t_i^1] \\
\frac{(x_i - t_i^j)^2}{(x_i - t_i^{j-2})^2 + (x_i - b_i)^2}, & x_i \in [t_i^{j-2}, t_i^j] \\
0, & \text{otherwise}
\end{cases}, \quad 2 \leq j \leq P_i
$$

$$
\mu_{A_{Pi}}(x_i, \theta) = \begin{cases} 
\frac{(x_i - t_i^{P_i-2})^2}{(x_i - t_i^{P_i-2})^2 + (x_i - b_i)^2}, & x_i \in [t_i^{P_i-2}, b_i] \\
0, & \text{otherwise}
\end{cases}
\tag{37}
$$

The total number of possible $K$ rules depends on the number of membership functions for each input $i.e., K = \Pi_{i=1}^{n} P_i$, where $P_i$ is the number of membership functions defined over the $i$th input. For any choice of membership functions (which can be constructed from a vector $\theta$), (35) can be rewritten as a function of $\theta$:

$$
F_i(x_1, x_2, \ldots, x_n) = \frac{\sum_{i=1}^{K} \mu_{A_{ij}}(x_i, \theta)}{\sum_{i=1}^{K} \mu_{A_{ij}}(x_i, \theta)}
$$

$$
G_i(x_1, x_2, \ldots, x_n, \theta) = \frac{\prod_{j=1}^{n} \mu_{A_{ij}}(x_j, \theta)}{\prod_{j=1}^{n} \mu_{A_{ij}}(x_j, \theta)}
$$

Let us introduce the following notation: $\alpha = [\alpha_{ij}]_{i=1, \ldots, K} \in R^K$, $x = [x_1, \ldots, x_n] \in R^n$, $G = [G_i(x, \theta)]_{i=1, \ldots, K} \in R^K$. Now, (35) becomes

$$
F_i(x) = G^T(x, \theta) \alpha.
$$

In this expression, $\theta$ is not allowed to be any arbitrary vector, since the elements of $\theta$ must ensure the following:

1) In the case of trapezoidal membership functions

$$
a_i < t_i^1 < \cdots < t_i^{P_i} < b_i \quad \forall i = 1, \ldots, n \tag{38}
$$
2) In the case of one-dimensional clustering-criterion-based membership functions

\[ a_i < t_i^1 < \cdots < t_i^{P_i-2} < b_i, \quad \forall i = 1, \ldots, n \tag{39} \]

To preserve the linguistic interpretation of fuzzy rule base [36]. In other words, there must exist some \( \epsilon_i > 0 \) for all \( i = 1, \ldots, n \) such that for trapezoidal membership functions

\[ t_i^{j+1} - t_i^j \geq \epsilon_i \quad \text{for all} \quad j = 1, 2, \ldots, (2P_i - 3) \]

\[ b_i - t_i^{P_i-2} \geq \epsilon_i. \]

These inequalities can be written in terms of a matrix inequality \( c \theta \geq h \) [18], [37]–[42]. Hence, the output of a Sugeno-type fuzzy model

\[ F_\theta(x) = G^T(x, \theta) \alpha, \quad c \theta \geq h \]

is linear in consequents (i.e., \( \alpha \)) but nonlinear in antecedents (i.e., \( \theta \)).

REFERENCES


Mohit Kumar (M’08) received the B.Tech. degree in electrical engineering from the National Institute of Technology, Hamirpur, India, in 1999, the M.Tech. degree in control engineering from the Indian Institute of Technology, Delhi, India, in 2001, and the Ph.D. degree (summa cum laude) in electrical engineering from Rostock University, Rostock, Germany, in 2004.

From 2001 to 2004, he was a Research Scientist with the Institute of Occupational and Social Medicine, Rostock. He is currently with the Center for Life Science Automation, Rostock. His current research interests include robust adaptive fuzzy identification, fuzzy logic in medicine, and robust adaptive control.

Regina Stoll received the Dip.-Med., the Dr.med. in occupational medicine, and the Dr.med.habil in occupational and sports medicine from Rostock University, Rostock, Germany, in 1980, 1984, and 2002, respectively.

She is the Head of the Institute of Preventive Medicine, Rostock. She is a faculty member with the Medicine Faculty and a Faculty Associate with the College of Computer Science and Electrical Engineering, Rostock University. She also holds an adjunct faculty member position with the Industrial Engineering Department, North Carolina State University, Raleigh. Her current research interests include occupational physiology, preventive medicine, and cardiopulmonary diagnostics.

Norbert Stoll received the Dip.-Ing. in automation engineering and the Ph.D. degree in measurement technology from Rostock University, Rostock, Germany, in 1979 and 1985, respectively.

He was the Head of the analytical chemistry section with the Academy of Sciences of GDR, Central Institute for Organic Chemistry, until 1991. From 1992 to 1994, he was an Associate Director of the Institute of Organic Catalysis, Rostock. Since 1994, he has been a Professor of measurement technology with the Engineering Faculty, University of Rostock.

From 1994 to 2000, he was the Director of the Institution of Automation at Rostock University. Since 2003, he has been the Vice President of the Center for Life Science Automation, Rostock. His current research interests include medical process measurement, lab automation, and smart systems and devices.