Variational Bayes for a Mixed Stochastic/Deterministic Fuzzy Filter
Mohit Kumar, Norbert Stoll, and Regina Stoll

Abstract—This study, under the variational Bayes (VB) framework, infers the parameters of a Takagi–Sugeno fuzzy filter having deterministic antecedents and stochastic consequents. The aim of this study is to take advantages of the VB framework to design fuzzy-filtering algorithms, which include an automated regularization, incorporation of statistical noise models, and model-comparison capability. The VB method can be easily applied to the linear-in-parameters models. This paper applies the VB method to the nonlinear filters without using Taylor expansion for a linear approximation of some nonlinear function. It is assumed that the nonlinear parameters (i.e., antecedents) of the fuzzy filter are deterministic, while linear parameters are stochastic. The VB algorithm, by maximizing a strict lower bound on the data evidence, makes the approximate posterior of linear parameters as close to the true posterior as possible. The nonlinear deterministic parameters are tuned in a way to further increase the lower bound on data evidence. The VB paradigm can be used to design an algorithm that automatically selects the most-suitable fuzzy filter out of the considered finite set of fuzzy filters. This is done by fitting the observed data as a stochastic combination of the different Takagi–Sugeno fuzzy filters such that the individual filters compete with one another to model the data.

Index Terms—Fuzzy filtering, probability distribution, Takagi–Sugeno fuzzy model, variational Bayes (VB).

I. INTRODUCTION

Fuzzy systems, which are based on fuzzy-set theory [1], [2], have been proposed in the literature to deal with the uncertainties. Our recent work on fuzzy methods for a proper handling of the uncertainties associated with a few real-world applications has been reported in [3]–[10]. The use of fuzzy systems in data-driven modeling is a topic that has been widely studied by researchers [11]–[36] due to the successful applications of fuzzy techniques in data mining, prediction, control, classification, simulation, and pattern recognition.

The robustness of an identification method has become a key issue in the presence of the uncertainties in the data. Therefore, several robust methods of fuzzy identification have been developed [31]–[49]. The robustness against outliers was achieved in [31] by optimizing a robust objective function. Regularization is a general method to improve the robustness of identification algorithms [34]. Regularization was used in [32] and [37] to convert the fuzzy-identification problem to a well-posed problem. Hong et al. [35] suggested a regularized orthogonal least-squares algorithm combined with a D-optimality that is used for subspace-based rule selection for a linear-in-parameter fuzzy model. A learning algorithm based on input-to-state stability approach was introduced in [33]. Kumar et al. [39] considered the identification of fuzzy models with uncertain data using semidefinite programming and second-order cone programming. The min–max approach to fuzzy-model parameters estimation, which tries to minimize the worst-case effect of disturbances on the estimation errors, was studied in [38], [40], [41], [43], and [44]. The criterion of integral-squared error with exponential forgetting was used in [42] for a robust estimation of fuzzy-model parameters. A clustering algorithm (which is termed as robust fuzzy-regression agglomeration) and a robust cost function were used in [45] for fuzzy modeling with outliers. The fuzzy-modeling problem was studied in a probabilistic Bayesian-learning framework using the extended relevance vector machine [36]. Support-vector regression techniques have been integrated with the fuzzy systems for a robust behavior [48], [49]. The fuzzy-model parameters were learned by a combination of fuzzy clustering and linear support-vector regression [47]. The ε-insensitive learning of fuzzy-model parameters was suggested in [46]. The fuzzy-model parameter-estimation algorithms, which are presented in [28] and [30], have the following features.

1) Several researchers estimate the membership-functions-related parameters (i.e., antecedents) based on some criteria (e.g., fuzzy-clustering criterion), while the estimation of linear parameters (i.e., consequents) is based on a different criterion (e.g., support-vector regression). However, a more elegant approach is to estimate both antecedents and consequents based on the filtering criterion (e.g., $H^\infty$-optimal filtering).

2) A mathematical framework should be available to design the fuzzy-filtering algorithms, as well as for their analysis in terms of stability, robustness, and steady-state error.

The Bayesian framework based on Bayes’ theorem is a powerful technique for the statistical inference of model parameters. The variational Bayes (VB) framework has the advantage of being less computationally intensive than other Bayesian methods. The VB method approximates the posterior distributions over a model in an analytical manner [50]. The VB method minimizes the Kullback–Leibler (KL) divergence of the approximate posterior from the true posterior density [51]. The expectation–maximization (EM) algorithm is a special case of VB [52]. The
VB, by virtue of being Bayesian method, has the following advantages: It can incorporate regularizing priors, complex noise models, and perform model comparisons. The VB method has been applied to the nonlinear models in [53]–[55] using Taylor expansion. However, the convergence of the VB method with the use of Taylor expansion (as given in [55]) is not guaranteed.

The Bayesian-inference problem determines the parameters $w$ of a model $m$ using available data $y$ based on Bayes’s theorem

$$ p(w|y, m) = \frac{p(y|w, m)p(w|m)}{p(y|m)}.$$  

Bayes’s theorem provides the posterior probability of the parameters given the data and the model. The analytical evaluation of posterior probability distribution is not possible in every case. Thus, it is approximated by a variational distribution

$$ q(w) \approx p(w|y, m)$$

where $q(w)$ is restricted to belong to a family of distributions of simpler form. This form is selected by minimizing the difference (in terms of KL divergence) between $q$ and true posterior. The KL divergence of $p(w|y, m)$ from $q(w)$ is defined as

$$ \text{KL}(q||p) = \int q(w) \log \frac{q(w)}{p(w|y, m)} dw.$$  

The logarithmic evidence for the data is given as

$$ \log p(y|m) = \log \int p(y, w|m) dw = \log \int q(w) \frac{p(y, w|m)}{q(w)} dw \geq \int q(w) \log \frac{p(y, w|m)}{q(w)} dw = \mathcal{F}(q(w), m) $$

where we have made use of Jensen’s inequality. Any probability distribution $q(w)$ gives rise to a lower bound $\mathcal{F}(q(w), m)$ on the logarithmic evidence. The lower bound $\mathcal{F}(q(w), m)$ is the negative of a quantity known as free energy. Since

$$ \log p(y|m) = \mathcal{F}(q(w), m) + \text{KL}(q||p) $$

minimizing $\text{KL}(q||p)$ is equivalent to maximizing $\mathcal{F}(q(w), m)$ over $q(w)$. Therefore, posterior distribution $p(w|y, m)$ is inferred by estimating $q(w)$ correctly, i.e., by maximizing $\mathcal{F}(q(w), m)$ over $q(w)$.

A Takagi–Sugeno fuzzy filter, as explained in Appendix A, can be mathematically represented as

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

The filter, as seen from (1), is characterized by two types of parameters: antecedents ($\theta$) and consequents ($\alpha$). Expression (1) shows that the output of the fuzzy filter is linear in consequents (i.e., in the elements of vector $\alpha$) while being nonlinear in antecedents (i.e., in the elements of vector $\theta$). We study a type of filter with the following characteristics:

1) the nonlinear parameters $\theta$ being considered as deterministic;

2) the linear parameters $\alpha$ being considered as random variables.

We focus on a process with $n$ inputs (which is represented by the vector $x \in \mathbb{R}^n$) and a single output (which is represented by the scalar $y$). It is assumed that inputs–output data pairs $\{x(j), y(j)\}$ are related via

$$ y(j) = G^T (x(j), \theta) \alpha + n_j $$

where $n_j$ is the additive Gaussian uncertainty with mean 0 and a variance of $1/\phi$. The fuzzy-filtering algorithms should seek to estimate the vector $\theta$ and evaluate the posterior probability distribution of $\alpha$. The considered fuzzy-filtering problem in the VB framework is stated in Problem 1.

**Problem 1:** Given $N$ pairs of inputs–output data $\{x(j), y(j)\}_{j=1}^N$ and a structure $m$ (i.e., membership type, number of membership functions, and rules) of a Takagi–Sugeno filter of type (1) such that data satisfy (2), estimate $\theta$ and the variational distributions $(q(\alpha), q(\phi))$ by maximizing the lower bound on the quantity as follows:

$$ \log p(y(1), \ldots, y(N)|x(1), \ldots, x(N), \theta, m). $$

Let us introduce the following notations:

$$ Y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix} \in \mathbb{R}^N, \quad B(\theta) = \begin{bmatrix} G^T (x(1), \theta) \\ \vdots \\ G^T (x(N), \theta) \end{bmatrix} \in \mathbb{R}^{N \times K}, $$

$$ v = [n_1 \cdots n_N]^T \in \mathbb{R}^N. $$

Now, we have

$$ Y = B(\theta) \alpha + v $$

where $v$ is an additive Gaussian uncertainty with mean 0 and a variance of $1/\phi$

$$ p(v) \sim N(0, \frac{1}{\phi} I). $$

Problem 1 can be rewritten as follows.

**Problem 2:** Given $N$ pairs of input–output data $\{x(j), y(j)\}_{j=1}^N$ and a structure $m$ (i.e., membership type, number of membership functions and rules) of a Takagi–Sugeno filter of type (1) such that data satisfy (3), estimate $\theta$ and the variational distributions $(q(\alpha), q(\phi))$ by maximizing the lower bound on the quantity as follows:

$$ \log p(Y|B(\theta), m). $$

Our next concern is to fit the observed data as a stochastic combination of the different Takagi–Sugeno fuzzy filters such that the individual filters compete with one another to model the data. Assume that there are $S$ number of fuzzy filters (with their structures as $m^i$) such that the output of the $i$th filter (i.e., $B(\theta^i)\alpha^i$) should match to the observed output vector $Y$ in some optimal manner. Let $s_i$ (where $s_i = 1, 2, \ldots, S$) be a discrete indicator random variable whose value represents the chosen filter for data modeling, i.e.,

$$ \text{If } s_i = 1, \quad Y = B(\theta^1) \alpha^1 + v $$

$$ \vdots $$

$$ \text{If } s_i = S, \quad Y = B(\theta^S) \alpha^S + v. $$
Let \( \pi = [\pi_1, \ldots, \pi_S]^T \in R^S \), with \( 0 \leq \pi_s \leq 1 \) and \( \sum_{s=1}^{S} \pi_s = 1 \), be a vector of mixing proportions (i.e., the proportions by which individual fuzzy filters’ outputs are mixed to match the observed output vector). The discrete distribution of the indicator variable \( s_i \) is given as

\[
p(s_i = 1|\pi) = \pi_1, \ldots, p(s_i = S|\pi) = \pi_S.
\]

We model the probability density function of the observed output data as a weighted average of the individual fuzzy filters’ output density functions

\[
p(Y|\pi, \{B(\theta^{(s)})\}_{s=1}^{S}, \{m^{(s)}\}_{s=1}^{S}, \phi, \{\alpha^{s}\}_{s=1}^{S}) = \sum_{s=1}^{S} p(s_i|\pi)p(Y|s_i, B(\theta^{(s)}), \alpha^{s}, \phi, m^{s}).
\]

**Problem 3:** Given \( N \) pairs of input–output data \( \{x(j), y(j)\}_{j=1}^{N} \) and \( S \) different structures \( \{m^{(s)}\}_{s=1}^{S} \) of the Takagi–Sugeno fuzzy filters of type (1) such that data satisfy (4), estimate \( \{\theta^{s}\}_{s=1}^{S} \) and the variational distributions \( \{(q(\alpha^{s}), q(\phi)) \}_{s=1}^{S} \) by maximizing the lower bound on the quantity as follows:

\[
\log p(Y|\{B(\theta^{(s)})\}_{s=1}^{S}, \{m^{(s)}\}_{s=1}^{S}).
\]

**Remark 1:** Our approach looks similar to the well-known statistical modeling technique of “finite-mixture models.” The finite-mixture models are widely used for clustering where it is assumed that there exists a finite set of random sources whose convex combination might have generated the observed data. Our approach, however, is different in the sense that a component model (i.e., individual filter) tries to fit all the \( N \) pairs of data rather than a subset of the complete dataset, i.e., our approach, unlike finite-mixture modeling, does not partition the total dataset into \( S \) different clusters such that the data belonging to a cluster is modeled as having been generated by one of the component models in the set. Therefore, Problem 3 allows the different filters to compete with one another to model the data.

The aim of this study is to develop fuzzy-filtering algorithms based on the solutions of Problems 2 and 3. The motivation of the work is to take simultaneously the following advantages of the VB framework in designing fuzzy-filtering algorithms:

1. an automated regularization through priors;
2. model comparison capability (i.e., an automatic selection of the most-suited model);
3. handling uncertainty via incorporating statistical noise models.

The VB method is not a new technique and has been widely studied by the researchers. The VB method can be easily applied to the linear-in-parameters models. A contribution of this study is to extend VB method to the nonlinear fuzzy filters. The nonlinear deterministic parameters are estimated in such a way that the lower bound on data evidence is increased. The text also provides an algorithm that automatically selects the most-suitable fuzzy filter out of the considered finite set of fuzzy filters and infers its parameters. To the best knowledge of the authors, this is the first study that applies VB method to the introduced mixed stochastic/deterministic fuzzy filter to solve Problems 2 and 3.

**II. VARIATIONAL BAYESIAN INFERENCE OF A STOCHASTIC COMBINATION OF FUZZY FILTERS**

This section presents our approach to solve the Problem 3. Following distributions are chosen for the parameters priors:

\[
p(\alpha^s|m_0^s, \Lambda_0^s) = N(\alpha^s|m_0^s, (\Lambda_0^s)^{-1})
\]

\[
p(\phi|a_0, b_0) = \text{Ga}(\phi|a_0, b_0)
\]

\[
p(\pi|c_0d_0) = \text{Dir}(\pi|c_0d_0), \quad d_0 = \left[ \frac{1}{S}, \ldots, \frac{1}{S} \right]^T \in R^S.
\]

Gamma and Dirichlet distributions are defined as follows:

\[
\text{Ga}(\phi|a_0, b_0) = \frac{1}{\Gamma(a_0)b_0^{-a_0}} \phi^{a_0-1} e^{-\phi/b_0}, \quad \text{for } \phi > 0
\]

and \( a_0, b_0 > 0 \).

\[
\text{Dir}(\pi|c_0d_0) = \frac{\Gamma(c_0)}{\Gamma(c_0/S)^S} \pi^{(c_0/S)-1} \ldots \pi^{(c_0/S)-1}
\]

where \( \pi_1, \ldots, \pi_S \geq 0, \sum_{j=1}^{S} \pi_j = 1, \) and \( c_0 > 0 \). The logarithmic evidence for the data is given as

\[
\log p(Y|\{B(\theta^{(s)})\}_{s=1}^{S}, \{m^{(s)}\}_{s=1}^{S})
\]

\[
= \log \int d\pi d\alpha^1 \ldots d\alpha^S d\phi p(\pi|c_0d_0)p(\alpha^1|m_0^1, \Lambda_0^1) \ldots p(\alpha^S|m_0^S, \Lambda_0^S)p(\phi|a_0, b_0)
\]

\[
p(Y|\pi, \{B(\theta^{(s)})\}_{s=1}^{S}, \{m^{(s)}\}_{s=1}^{S}, \phi, \{\alpha^{s}\}_{s=1}^{S}).
\]

Using (5), we have

\[
\log p(Y|\{B(\theta^{(s)})\}_{s=1}^{S}, \{m^{(s)}\}_{s=1}^{S})
\]

\[
= \log \int d\pi d\alpha^1 \ldots d\alpha^S d\phi p(\pi|c_0d_0)p(\alpha^1|m_0^1, \Lambda_0^1) \ldots p(\alpha^S|m_0^S, \Lambda_0^S)p(\phi|a_0, b_0)
\]

\[
\sum_{s=1}^{S} p(s_i|\pi)p(Y|s_i, B(\theta^{(s)}), \alpha^{s}, \phi, m^{s}).
\]

For simplicity, we define \( \alpha = \{\alpha^1, \ldots, \alpha^S\}, m_0 = \{m_0^1, \ldots, m_0^S\}, \Lambda_0 = \{\Lambda_0^1, \ldots, \Lambda_0^S\} \) with \( p(\alpha|m_0, \Lambda_0) = \prod_{s=1}^{S} p(\alpha^s|m_0^s, \Lambda_0^s) \). The above integral can be written as

\[
\log p(Y|\{B(\theta^{(s)})\}_{s=1}^{S}, \{m^{(s)}\}_{s=1}^{S})
\]

\[
= \log \int d\pi d\alpha d\phi p(\pi|c_0d_0)p(\alpha|m_0, \Lambda_0)p(\phi|a_0, b_0)
\]

\[
\sum_{s=1}^{S} p(s_i|\pi)p(Y|s_i, B(\theta^{(s)}), \alpha^{s}, \phi, m^{s}).
\]

Introducing an arbitrary distribution \( q(\pi, \alpha, \phi) \) to lower bound the data evidence, we have the equation as shown at the bottom of the next page, where \( p(Y|s_1, \ldots) \) should be understood as \( p(Y|s_1, B(\theta^{s_1}), \alpha^{s_1}, \phi, m^{s_1}) \). We restrict our method to use the
approximation
\[
q(\pi, \alpha, \phi) \approx q(\pi) q(\alpha) q(\phi)
\]
\[
= q(\pi) \prod_{s_i=1}^{S} q(\alpha^{s_i}) q(\phi).
\]

This results in
\[
\log p(Y | \{B(\theta^{s_i})\}_{s_i=1}^{S}, \{m^{s_i}\}_{s_i=1}^{S})
\geq \int d\pi q(\pi) \log \frac{p(\pi | c_0 d_0)}{q(\pi)} + \int d\alpha q(\alpha) \log \frac{p(\alpha | m_0, \Lambda_0)}{q(\alpha)}
+ \int d\phi q(\phi) \log \frac{p(\phi | a_0, b_0)}{q(\phi)}
+ \int d\pi d\alpha d\phi q(\pi) q(\alpha) q(\phi) \log \left( \sum_{s_i=1}^{S} p(s_i | \pi) p(Y | s_i, \ldots) \right).
\]

Again, a discrete distribution \(q(s_i)\), with \(\sum_{s_i=1}^{S} q(s_i) = 1\), is introduced to further lower bound the data evidence
\[
\log p(Y | \{B(\theta^{s_i})\}_{s_i=1}^{S}, \{m^{s_i}\}_{s_i=1}^{S})
\geq \int d\pi q(\pi) \log \frac{p(\pi | c_0 d_0)}{q(\pi)} + \int d\alpha q(\alpha) \log \frac{p(\alpha | m_0, \Lambda_0)}{q(\alpha)}
+ \int d\phi q(\phi) \log \frac{p(\phi | a_0, b_0)}{q(\phi)}
+ \int d\pi d\alpha d\phi q(\pi) q(\alpha) q(\phi) \log \left( \sum_{s_i=1}^{S} q(s_i | \pi) p(Y | s_i, \ldots) \right)
\times \log \frac{p(s_i | \pi) p(Y | s_i, \ldots)}{q(s_i)}
\]

i.e.,
\[
\log p(Y | \{B(\theta^{s_i})\}_{s_i=1}^{S}, \{m^{s_i}\}_{s_i=1}^{S})
\geq \int d\pi q(\pi) \log \frac{p(\pi | c_0 d_0)}{q(\pi)} + \int d\alpha q(\alpha) \log \frac{p(\alpha | m_0, \Lambda_0)}{q(\alpha)}
+ \int d\phi q(\phi) \log \frac{p(\phi | a_0, b_0)}{q(\phi)}
+ \sum_{s_i=1}^{S} q(s_i) \int d\alpha d\phi q(\alpha) q(\phi) \log p(Y | s_i, \ldots)
+ \sum_{s_i=1}^{S} q(s_i) \int d\pi q(\pi) \log \frac{p(s_i | \pi)}{q(s_i)}.
\]

The lower bound is defined as a functional of the variational posterior distributions as follows:
\[
\mathcal{F}(q(\pi), q(\alpha), q(\phi), \{q(s_i)\}_{s_i=1}^{S})
= \int d\pi q(\pi) \log \frac{p(\pi | c_0 d_0)}{q(\pi)} + \int d\alpha q(\alpha) \log \frac{p(\alpha | m_0, \Lambda_0)}{q(\alpha)}
+ \int d\phi q(\phi) \log \frac{p(\phi | a_0, b_0)}{q(\phi)}
+ \sum_{s_i=1}^{S} q(s_i) \int d\alpha d\phi q(\alpha) q(\phi) \log p(Y | s_i, \ldots)
+ \sum_{s_i=1}^{S} q(s_i) \int d\pi q(\pi) \log \frac{p(s_i | \pi)}{q(s_i)}.
\]

Now, we are in a position to present our method to infer the parameters of the fuzzy-filters combination. Algorithm 1 lists the various steps involved in our method.

III. Optimization of a Lower Bound on the Data Evidence

A. Optimization w.r.t. \(q(\pi)\)

\(\mathcal{F}\) will be stationary w.r.t. distribution \(q(\pi)\), if
\[
\log(p(\pi | c_0 d_0)) - \log(q(\pi))
+ \sum_{s_i=1}^{S} q(s_i) \log(p(s_i | \pi)) + \text{cons}(q(\pi)) = 0, \ i.e.,
\log \left( \prod_{s_i=1}^{S} \pi_{s_i}^{\left(c_0/d_0 \right)^{-1}} \right) - \log(q(\pi))
+ \sum_{s_i=1}^{S} \log \left( \pi_{s_i}^{q(s_i)} \right) + \text{cons}(q(\pi)) = 0, \ i.e.,
\log \left( \prod_{s_i=1}^{S} \pi_{s_i}^{(c_0/S)^{-1} + q(s_i) - 1} \right)
- \log(q(\pi)) + \text{cons}(q(\pi)) = 0.
\]

This implies that
\[
q^*(\pi) = \text{Dir}(\pi | cd), \ d = [d_1 \ldots d_S]^T \in R^S, \ \sum_{s_i=1}^{S} d_{s_i} = 1
\]
such that
\[
cd_{s_i} = \frac{c_0}{S} + q(s_i).
\]

\[
\log p(Y | \{B(\theta^{s_i})\}_{s_i=1}^{S}, \{m^{s_i}\}_{s_i=1}^{S}) \geq \int d\pi d\alpha d\phi q(\pi, \alpha, \phi)
\log \frac{p(\pi | c_0 d_0)p(\alpha | m_0, \Lambda_0)p(\phi | a_0, b_0)\sum_{s_i=1}^{S} p(s_i | \pi)p(Y | s_i, \ldots)}{q(\pi, \alpha, \phi)}.
\]
**Algorithm 1** VB inference of the fuzzy filters combination

**Require:** Data pairs \( \{x(j), y(j)\}_{j=1}^{N} \).

1. Choose a total of \( S \) fuzzy filters’ structures \( \{m_s^{\nu}\}_{s=1}^{S} \); hyper-parameters \( c_0, m_0, \Lambda_0, a_0, b_0 \) which define the regularizing priors; parameters \( \{c_s^{\nu}, h_s^{\nu}\}_{s=1}^{S} \) such that the interpretability constraints on the membership functions of the \( s_i \)-th filter can be formulated as \( c_s^{\nu} h_s^{\nu} \geq h_s^{\nu} \).

2. Define

\[
\mathcal{F}(q(\pi), q(\alpha), q(\phi), \{q(s_i)\}_{s_i=1}^{S}, \{B(\theta^{\nu_s})\}_{s_i=1}^{S}, c_0, m_0, \Lambda_0, a_0, b_0, \{m_s^{\nu}\}_{s=1}^{S} ) = \\
\int d\pi q(\pi) \log \frac{p(\pi|o \theta_0)}{q(\pi)} + \int d\alpha q(\alpha) \log \frac{p(\alpha|m_0, \Lambda_0)}{q(\alpha)} \\
+ \int d\phi q(\phi) \log \frac{p(\phi|a_0, b_0)}{q(\phi)} \\
+ \sum_{s_i=1}^{S} q(s_i) \int d\alpha d\phi q(\alpha)(q(\phi) \log p(Y|s_i, B(\theta^{\nu_s}), \alpha^{\nu_s}, \phi, m^{\nu_s}) \\
+ \sum_{s_i=1}^{S} q(s_i) \int d\pi q(\pi) \log \frac{p(s_i | \pi)}{q(s_i)}.
\]

3. Derive analytically the correct expressions for distributions \( q(\pi), q(\alpha), q(\phi), q(s_i) \) by maximizing \( \mathcal{F} \) over \( \{q(\pi), q(\alpha), q(\phi), q(s_i)\} \) (i.e. by setting the functional derivatives of \( \mathcal{F} \) with respect to each free distribution equal to zero). Let \( q^{*}(\pi), q^{*}(\alpha), q^{*}(\phi), q^{*}(s_i) \) denote the obtained expressions for the variational distributions.

4. Estimate the optimal values of antecedents of fuzzy filters, i.e. \( \theta^{\nu_s}, \ldots, \theta^{S \nu_s} \), via maximizing \( \mathcal{F} \) further over \( \{\theta^{\nu_s}, \ldots, \theta^{S \nu_s}\} \). That is, solve the following constrained nonlinear optimization problem:

\[
(\theta^{1 \nu_s}, \ldots, \theta^{S \nu_s}) = \text{arg} \max_{(\theta^{1 \nu_s}, \ldots, \theta^{S \nu_s})} \left[ \mathcal{F}_0(\theta) ; c_s^{\nu_s} h_s^{\nu_s} \geq h_s^{\nu_s}\right]
\]

where

\[
\mathcal{F}_0(\theta) = \mathcal{F}(q^{*}(\pi), q^{*}(\alpha), q^{*}(\phi), q^{*}(s_i))_{s_i=1}^{S}, \{B(\theta^{\nu_s})\}_{s_i=1}^{S}, c_0, m_0, \Lambda_0, a_0, b_0, \{m_s^{\nu}\}_{s=1}^{S}.
\]

5. return \( \{q^{*}(\pi), q^{*}(\alpha), q^{*}(\phi), q^{*}(s_i)\}_{s_i=1}^{S} \)

**Using** \( \sum_{s_i=1}^{S} d_{s_i} = 1 \), and \( \sum_{s_i=1}^{S} q(s_i) = 1 \), we have

\[
c = c_0 + 1.
\]

Thus

\[
d_{s_i} = \frac{1}{c_0 + 1} \left( \frac{c_0}{S} + q(s_i) \right).
\]

**B. Optimization w.r.t.** \( q(\alpha^{1}) \), \ldots, \( q(\alpha^{S}) \)

\( \mathcal{F} \) will be stationary w.r.t. distribution \( q(\alpha^{s_i}) \), where \( s_i = 1, \ldots, S \), if

\[
\log(p(\alpha^{s_i}|m_0^{s_i}, -m_0^{s_i})) - \log(q(\alpha^{s_i})) \\
+ q(s_i) \int d\phi q(\phi) \log(p(Y|s_i, B(\theta^{\nu_s}), \alpha^{s_i}, \phi, m^{s_i})) \\
+ \text{cons}\{q(\alpha^{s_i})\} = 0.
\]

The substitutions

\[
p(\alpha^{s_i}|m_0^{s_i}, \Lambda_0^{s_i}) \propto \exp \left( -\frac{1}{2} (\alpha^{s_i} - m_0^{s_i})^T \Lambda_0^{s_i} (\alpha^{s_i} - m_0^{s_i})\right) \\
p(Y|s_i, B(\theta^{s_i}), \alpha^{s_i}, \phi, m^{s_i}) \\
\propto \exp \left( -\frac{\phi}{2} (Y - B(\theta^{s_i})\alpha^{s_i})^T (Y - B(\theta^{s_i})\alpha^{s_i}) \right)
\]

result in

\[
-\frac{1}{2} (\alpha^{s_i} - m_0^{s_i})^T \Lambda_0^{s_i} (\alpha^{s_i} - m_0^{s_i}) - \log(q(\alpha^{s_i})) \\
- \frac{q(s_i)}{2} \int d\phi q(\phi) \left( Y - B(\theta^{s_i})\alpha^{s_i} \right)^T (Y - B(\theta^{s_i})\alpha^{s_i}) + \text{cons}\{q(\alpha^{s_i})\} = 0.
\]

This implies that

\[
q^{*}(\alpha^{s_i}) = N(\alpha^{s_i}|m_0^{s_i}, (\Lambda_0^{s_i})^{-1}), \text{ such that} \\
\Lambda_0^{s_i} = \Lambda_0^{s_i} + q(s_i) \left( \int d\phi q(\phi) \phi (B(\theta^{s_i}))^T B(\theta^{s_i}) \right)
\]

\[
\Lambda_0^{s_i} = \Lambda_0^{s_i} + q(s_i) \left( \int d\phi q(\phi) \phi (B(\theta^{s_i}))^T B(\theta^{s_i}) \right) Y.
\]

Thus

\[
m^{s_i} = \left( \Lambda_0^{s_i} \right)^{-1} \left[ \Lambda_0^{s_i} m_0^{s_i} + q(s_i) \left( \int d\phi q(\phi) \phi (B(\theta^{s_i}))^T Y \right) \right].
\]

The term \( \int d\phi q(\phi) \phi \), appearing in the expressions for \( \Lambda_0^{s_i} \) and \( m^{s_i} \), will be evaluated after obtaining the correct expression for \( q(\phi) \) in the coming part of the text.

**C. Optimization w.r.t.** \( q(\phi) \)

Before equating the derivative of \( \mathcal{F} \) w.r.t. \( q(\phi) \) equal to zero, note that

\[
\log(p(Y|s_i, B(\theta^{s_i}), \alpha^{s_i}, \phi, m^{s_i})) \\
= \frac{N}{2} \log(\phi) - \frac{\phi}{2} (Y - B(\theta^{s_i})\alpha^{s_i})^T (Y - B(\theta^{s_i})\alpha^{s_i}) \\
- \frac{N}{2} \log(2\pi).
\]

\( \mathcal{F} \) will be stationary w.r.t. distribution \( q(\phi) \), if

\[
\log(p(\phi|a_0, b_0)) - \log(q(\phi)) + \text{cons}\{\phi\} \\
+ \sum_{s_i=1}^{S} q(s_i) \int d\alpha^{s_i} q(\alpha^{s_i}) \left[ \frac{N}{2} \log(\phi) \\
- \frac{\phi}{2} (Y - B(\theta^{s_i})\alpha^{s_i})^T (Y - B(\theta^{s_i})\alpha^{s_i}) \right] = 0, \text{i.e.,}
\]

\[
\log(p(\phi|a_0, b_0)) - \log(q(\phi)) + \text{cons}\{\phi\} \\
- \frac{\phi}{2} \sum_{s_i=1}^{S} q(s_i) \int d\alpha^{s_i} q(\alpha^{s_i})(Y - B(\theta^{s_i})\alpha^{s_i})^T \times (Y - B(\theta^{s_i})\alpha^{s_i}) \\
+ \frac{N}{2} \log(\phi) = 0.
\]
Using the facts
\[
\log(p(\phi|a_0, b_0)) = (b_0 - 1) \log(\phi) - \frac{\phi}{a_0} + \text{cons}\{\phi\}
\]
\[
\int d\alpha^s \ q(\alpha^s) (Y - B(\theta^s) \alpha^s)^T (Y - B(\theta^s) \alpha^s) = (Y - B(\theta^s) m^s)^T (Y - B(\theta^s) m^s)
+ \text{Tr} \left( (\Lambda^s)^{-1} (B(\theta^s))^T B(\theta^s) \right)
\]
we have
\[
(b_0 - 1) \log(\phi) - \frac{\phi}{a_0} - \log(q(\phi)) + \text{cons}\{\phi\} + \frac{N}{2} \log(\phi)
- \frac{\phi}{2} \sum_{s_i=1}^S q(s_i) [(Y - B(\theta^s) m^s)^T (Y - B(\theta^s) m^s)]
+ \text{Tr} \left( (\Lambda^s)^{-1} (B(\theta^s))^T B(\theta^s) \right)
\]
This implies that
\[
q^*(\phi) = \text{Ga}(\phi|a, b), \text{ such that }
\frac{1}{a} = \frac{1}{a_0} + \frac{1}{2} \sum_{s_i=1}^S q(s_i) [(Y - B(\theta^s) m^s)^T (Y - B(\theta^s) m^s)]
+ \text{Tr} \left( (\Lambda^s)^{-1} (B(\theta^s))^T B(\theta^s) \right)
\]
\[
b = b_0 + \frac{N}{2}.
\]

D. Optimization w.r.t. \( q(s_i) \)

\[ F \] will be stationary w.r.t. distribution \( q(s_i) \), if
\[
\int d\pi q(\pi) \log(p(s_i|\pi)) - \log(q(s_i))
+ \int d\alpha^s d\phi q(\alpha^s) \log p(Y|s_i, B(\theta^s), \alpha^s, \phi, m^s)
= 0
\]
Per a result of Dirichlet distributions, we have
\[
\int d\pi q(\pi) \log(p(s_i|\pi)) = \Psi(cd_{s_i}) - \Psi(c)
\]
where \( \Psi(\cdot) \) is the digamma function. Now, consider the term
\[
\int d\alpha^s d\phi q(\alpha^s) \log p(Y|s_i, B(\theta^s), \alpha^s, \phi, m^s)
= \int d\phi q(\phi)
\times \left( \int d\alpha^s q(\alpha^s) \log p(Y|s_i, B(\theta^s), \alpha^s, \phi, m^s) \right)
\]
\[
= \frac{N}{2} \int d\phi q(\phi) \log(\phi) - \frac{N}{2} \log(2\pi)
- \frac{1}{2} \int d\phi q(\phi) (\phi) [\phi (Y - B(\theta^s) m^s)^T (Y - B(\theta^s) m^s)]
+ \text{Tr} \left( (\Lambda^s)^{-1} (B(\theta^s))^T B(\theta^s) \right)
\]
\[
= \frac{N}{2} (\Psi(b) + \log(a)) - \frac{N}{2} \log(2\pi)
- \frac{ab}{2} [\phi (Y - B(\theta^s) m^s)^T (Y - B(\theta^s) m^s)]
+ \text{Tr} \left( (\Lambda^s)^{-1} (B(\theta^s))^T B(\theta^s) \right)
\]
Here, we have used some results of Gamma distribution. Finally, the equilibrium equation becomes
\[
\Psi(cd_{s_i}) - \Psi(c) - \log(q(s_i)) + \frac{N}{2} (\Psi(b) + \log(a))
- \frac{N}{2} \log(2\pi) - \frac{ab}{2} [\phi (Y - B(\theta^s) m^s)^T (Y - B(\theta^s) m^s)]
+ \text{Tr} \left( (\Lambda^s)^{-1} (B(\theta^s))^T B(\theta^s) \right) = 0.
\]
This implies that
\[
q^*(s_i) = \frac{1}{Z} \exp \left( \Psi(cd_{s_i}) - \Psi(c) + \frac{N}{2} (\Psi(b) + \log(a)) \right)
- \frac{N}{2} \log(2\pi) - \frac{ab}{2} r(m^s, \Lambda^s, \theta^s)
\]
where \( Z \) is the normalization constant such that \( \sum_{s_i=1}^S q(s_i) = 1 \), and
\[
r(m^s, \Lambda^s, \theta^s) = (Y - B(\theta^s) m^s)^T (Y - B(\theta^s) m^s)
+ \text{Tr} \left( (\Lambda^s)^{-1} (B(\theta^s))^T B(\theta^s) \right)
\]
The constant terms in the expression of \( q^*(s_i) \), which do not vary as \( s_i \) varies from 1 to \( S \), can be included in the normalization constant. This simplifies the expression for \( q^*(s_i) \) as
\[
q^*(s_i) = \frac{1}{Z} \exp \left( \Psi(cd_{s_i}) - \frac{ab}{2} r(m^s, \Lambda^s, \theta^s) \right)
\]
E. Optimization w.r.t. (\( \theta^1, \ldots, \theta^S \))

The optimal values of antecedents of fuzzy filters are obtained via maximizing
\[
F(q^*(\pi), q^*(\alpha), q^*(\phi), \{q^*(s_i)\}_{s_i=1}^S, \{B(\theta^s)\}_{s_i=1}^S, c_0, m_0, \
\Lambda_0, a_0, b_0, \{m^s\}_{s_i=1}^S)
\]
over (\( \theta^1, \ldots, \theta^S \)). The lower bound on the logarithmic evidence for the data can be expressed as
\[
F(q(\pi), q(\alpha), q(\phi), \{q(s_i)\}_{s_i=1}^S, \{B(\theta^s)\}_{s_i=1}^S, c_0, m_0, \
\Lambda_0, a_0, b_0, \{m^s\}_{s_i=1}^S)
\]
\[
= \sum_{s_i=1}^S q(s_i) \int d\alpha^s d\phi q(\alpha) q(\phi) \log p(Y|s_i, B(\theta^s), \alpha^s, \phi, m^s) + \sum_{s_i=1}^S q(s_i) \int d\pi q(\pi) \log \frac{p(s_i|\pi)}{q(s_i)}
- \int d\pi q(\pi) \log \frac{q(\pi)}{p(\pi|c_0, m_0)} - \int d\alpha q(\alpha) \log \frac{q(\alpha)}{p(\alpha|m_0, \Lambda_0)}
- \int d\phi q(\phi) \log \frac{q(\phi)}{p(\phi|a_0, b_0)}
\]
The value of $\mathcal{F}$ at obtained optimal distributions (i.e., at $q(\pi) = q'(\pi), q(\alpha) = q'(\alpha), q(\phi) = q'(\phi), q(s) = q'(s)$) is given as:

$$
\mathcal{F}(q'(\pi), q'(\alpha), q'(\phi), \{q'(s_i)\})_{s_i=1}^S = \mathcal{B}(\Theta^{s_i})_{s_i=1}^S \Lambda_0, a_0, b_0, \{m^{s_i}\}_{s_i=1}^S
$$

$$
= \frac{N}{2}(\Psi(b) + \log(a) - \log(2\pi)) - \frac{ab}{2} \sum_{s_i=1}^S q'(s_i)r(m^{s_i}, \Lambda^{s_i}, \Theta^{s_i})
$$

$$
+ \sum_{s_i=1}^S q'(s_i)[\Psi(cd_{s_i}) - \Psi(c) - \log(q'(s_i))]
$$

$$
- \log \left( \frac{\Gamma(c)}{\Gamma(c_0)} \right)
$$

$$
- \sum_{s_i=1}^S \log \left( \frac{\Gamma(c_0 / S)}{\Gamma(cd_{s_i})} \right)
$$

$$
- \sum_{s_i=1}^S \left[ cd_{s_i} - \frac{c_0}{S} \right] [\Psi(cd_{s_i}) - \Psi(c)]
$$

$$
- \sum_{s_i=1}^S \left[ 1 \frac{1}{2} \log \left( \frac{(\Lambda^{s_i}_0)^{-1}}{(\Lambda^{s_i})^{-1}} \right) + \frac{1}{2} Tr (\Lambda^{s_i}_0 (\Lambda^{s_i})^{-1}) \right]
$$

$$
+ \frac{1}{2} (m^{s_i} - m^{s_i}_0)^T \Lambda^{s_i}_0 (m^{s_i} - m^{s_i}_0) - \frac{K}{2} - \log(\Gamma(b_0))
$$

$$
- b_0 \log(a_0) + \log(\Gamma(b)) + b_0 \log(a) - b \Psi(b) + b_0 \Psi(b)
$$

$$
- \frac{b}{a} \frac{a}{a_0} + b.
$$

At this point, we observe that for the given values of probability mass functions $\{q'(s_i)\}_{s_i=1}^S$ and parameters $(c, \{d_{s_i}\}_{s_i=1}^S, \{\Lambda^{s_i}\}_{s_i=1}^S, \{m^{s_i}\}_{s_i=1}^S, a, b)$, an increase in the value of $\mathcal{F}$ will occur if parameters $\{\theta^{s_i}\}_{s_i=1}^S$ are optimized based on the following optimization problem:

$$
\theta^{s_i,*} = \arg \min_{\theta^{s_i}} [r(m^{s_i}, \Lambda^{s_i}, \Theta^{s_i})] \geq h^{s_i}.
$$

### IV. Algorithms

The rules to update the parameters of the distributions $(q(\pi), q(\alpha), q(\phi), q(s))$ and antecedents of fuzzy filters are summarized as follows:

$$
\theta^{s_i,*} = \arg \min_{\theta^{s_i}} [r(m^{s_i}, \Lambda^{s_i}, \Theta^{s_i})] \geq h^{s_i}]
$$

$$
\Lambda^{s_i} = \Lambda^{s_i}_0 + q'(s_i)ab(B(\Theta^{s_i,*})^T B(\Theta^{s_i,*})^{-1})
$$

$$
m^{s_i} = [\Lambda^{s_i}_0 + q'(s_i)ab(B(\Theta^{s_i,*})^T B(\Theta^{s_i,*})^{-1})]^{-1} \Lambda^{s_i}_0 m^{s_i}_0 + q'(s_i)ab(B(\Theta^{s_i,*})^T)\Psi(s_i)
$$

$$
q'(s_i) = \frac{1}{2} e^{-\frac{b}{2} r(m^{s_i}, \Lambda^{s_i}, \Theta^{s_i,*})}
$$

where $\mathcal{Z}$ is s.t. $\sum_{s_i=1}^S q'(s_i) = 1$

$$
b = b_0 + \frac{N}{2}
$$

$$
\frac{1}{a} = \frac{1}{a_0} + \frac{1}{2} \sum_{s_i=1}^S q'(s_i)r(m^{s_i}, \Lambda^{s_i}, \Theta^{s_i,*})
$$

$$
c = c_0 + 1
$$

$$
d_{s_i} = \frac{1}{c_0 + 1} \left( c_0 + q'(s_i) \right).
$$

Here, in the expressions for $\Lambda^{s_i}$ and $m^{s_i}$, the term $\int d\phi q(\phi) \phi$ has been substituted as $ab$.

Each of these update rules is guaranteed to monotonically increase the objective function $\mathcal{F}$. Therefore, several iterations of update rules can be performed to increase $\mathcal{F}$ until a consistent solution is reached. The iterative optimization process can be terminated if the increase in $\mathcal{F}$ from an iteration to the next is less than a tolerance limit. Algorithm 2 summarizes our method to infer the parameters of the fuzzy-filters combination via maximizing $\mathcal{F}$.

**Remark 2:** Problem 2 is simpler and a particular case of the Problem 3, where there is only one fuzzy filter whose parameters need to be inferred. The update rules, in this case, are summarized in Algorithm 3.

**Remark 3:** Algorithm 3 can be started with an initial guess of $m|0|_0 = m_0, \Lambda|0|_0 = \Lambda_0, a|0| = a_0$, and $b|0| = b_0$, and $\theta^{s_i}|0$, in the case of grid partitioning, can be set for the uniformly distributed membership functions in the corresponding inputs’ ranges. Regarding the starting point of Algorithm 2, one can perform the following.

1) Infer independently the parameters of each of the $S$ filters using Algorithm 3.

2) Let $(\theta^{s_i,*}_{\text{single}}, \Lambda^{s_i}_{\text{single}}, m^{s_i}_{\text{single}}, a^{s_i}_{\text{single}}, b^{s_i}_{\text{single}})$ denote the parameters of the $s_i$th filter returned by Algorithm 3 and $\mathcal{F}^{s_i}_{\text{single}}$ be the corresponding value of the optimized objective function. Now, the initial guess can be chosen as:

$$
q'(s_i)|0 = \exp(\mathcal{F}^{s_i}_{\text{single}}), \text{ where } \sum_{s_i=1}^S q'(s_i)|0 = 1
$$

$$
\theta^{s_i,*}|0 = \theta^{s_i,*}_{\text{single}}, \text{ } m^{s_i}|0 = m^{s_i}_{\text{single}}, \text{ } \Lambda^{s_i}|0 = \Lambda^{s_i}_{\text{single}}
$$

$$
\frac{1}{a|0} = \frac{1}{a} + \frac{1}{2} \sum_{s_i=1}^S q'(s_i)|0 r(m^{s_i}_{\text{single}}, \Lambda^{s_i}_{\text{single}}, \theta^{s_i,*}_{\text{single}})
$$

$$
b|0 = b_0 + \frac{N}{2}, \text{ } c|0 = c_0 + 1
$$

$$
d_{s_i}|0 = \frac{1}{c_0 + 1} \left( c_0 + q'(s_i)|0 \right).
$$

Here, the posterior distribution of the indicator variable is initiated proportional to the exponential of the lower bound on logarithmic evidence. The reason for this is that Algorithm 2, like EM algorithm, has the chances of being trapped in a local maxima. Therefore, a good starting point (which is obtained by the inference of component models using Algorithm 3) can be
Algorithm 2 An algorithm for VB inference of the fuzzy filters combination

Require: Data pairs \( \{x(j), y(j)\}_{j=1,\ldots,N} \).

1: Choose a total of \( S \) fuzzy filters' structures \( \{m_i\}_{i=1}^S \) hyper-parameters \( \alpha_0, \alpha_1, \alpha_2, b_0 \) which define the regularizing priors; parameters \( \{c_i, d_i\}_{i=1}^S \) such that the interpretability constraints on the membership functions of the \( s_i \)-th filter can be formulated as \( c_i \theta_i^* \geq h^* \).

2: Set iteration count \( t = 0 \) and choose a tolerance limit (say, equal to 0.01%).

3: if \( \max_{s_i} \{q^*(s_i)|\theta_i|+1 \leq q^*(s_i)|\theta_i| < 0.0001\} \) & \( (t > 0) \) then

4: return \( (\theta_i^*|\theta_i|+1, \cdots, \theta_i^*|\theta_i|+1) \) and \( (c_i|\theta_i|+1, d_i|\theta_i|+1, q^*(s_i)|\theta_i|+1) \) \( s_i = \{1, \ldots, S\} \).

5: else

6: Update the parameters as follows

\[
\theta_i^*|\theta_i|+1 = \arg \min_{\theta_i^*|\theta_i|} \left[ r(m_i|\theta_i^*|\theta_i), c_i \theta_i^* \geq h^* \right]
\]

\[
\Lambda_i|\theta_i|+1 = \Lambda_0 + \{q^*(s_i)|\theta_i|d_i|\theta_i| B(B(\theta_i^*|\theta_i|+1)) \}
\]

\[
m_i|\theta_i|+1 = \left[ \Lambda_0 + \{q^*(s_i)|\theta_i|d_i|\theta_i| B(B(\theta_i^*|\theta_i|+1)) \} \right]^{-1}
\]

\[
\{q^*(s_i)|\theta_i|+1 = \frac{1}{Z} \exp \left( \Psi(c_i|\theta_i|+1) \right)
\]

where \( Z \) is s.t. \( \sum_{s_i=1}^S q^*(s_i)|\theta_i|+1 = \lambda \).

\[
\frac{1}{a_i|\theta_i|+1} = \frac{1}{a_0} + \frac{1}{2} \sum_{s_i=1}^S q^*(s_i)|\theta_i|+1 \left[ m_i|\theta_i|+1 + \theta_i^*|\theta_i|+1 \right]
\]

\[
c_i|\theta_i|+1 = c_0 + 1
\]

\[
d_i|\theta_i|+1 = \frac{1}{c_0 + 1} \left( c_0 + q^*(s_i)|\theta_i|+1 \right)
\]

Here, \( \{m_i|\theta_i|+1, \Lambda_i|\theta_i|+1, q^*(s_i)|\theta_i|+1, \} \) denote the initial guess.

7: end if

taken to reduce the chances of algorithm convergence to a local maxima.

Remark 4: After optimizing \( F \) via Algorithm 2, one finds for a model \( m^* \)

\[
q^*(s_i = i^*) \approx 1 \quad \text{and} \quad q^*(s_i) \approx 0, \quad \text{for} \quad s_i = 1, \ldots, i^* - 1, i^* + 1, \ldots, S
\]
i.e., the \( i^* \)-th fuzzy filter is the winner model that takes the most responsibility of the data. Therefore, Algorithm 2 is capable of automatically selecting the most-suitable fuzzy filter out of the set and inferring its parameters.

Remark 5: Algorithms 2 and 3 were implemented in MATLAB 6.5 [56]. The first update of the step 6 in both algorithms involves a nonlinear constrained optimization problem. The parameters estimation, which is based on the nonlinear optimization problem, was performed by running a single iteration of the algorithm “fmincon” that is available in MATLAB Optimization Toolbox [57].

V. SIMULATION STUDIES

A. Study I

The first example, which is taken from [45], deals with the identification of a noisy time series

\[
x_j = 1.5x_{j-1} \exp \left( -\frac{x_{j-1}^2}{4} \right) + \epsilon_j, \quad \epsilon_j \sim N(0, 1)
\]

\[
y_j = x_j + v_j.
\]

Here, \( v_j \) is generated by a gross-error model, which is defined as

\[
F = (1 - \epsilon)G + \epsilon H
\]

where \( F \) is the noise distribution, and \( G \) and \( H \) are the probability distributions that occur with probabilities \( 1 - \epsilon \) and \( \epsilon \), respectively. As given in [45], the gross-error model with \( \epsilon = 0.05 \)
TABLE I
SIMULATIONS RESULTS OF STUDY 1

<table>
<thead>
<tr>
<th>method</th>
<th>number of fuzzy rules</th>
<th>number of model parameters</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithm 2</td>
<td>3.38 ± 0.7253</td>
<td>10.14 ± 2.176</td>
<td>0.3433 ± 0.1221</td>
</tr>
<tr>
<td>Best known result from [45]</td>
<td>5</td>
<td>20</td>
<td>0.4200</td>
</tr>
</tbody>
</table>

G ∼ N(0, 0.05), and H ∼ N(0, 3) was taken for the simulation study. It is required to train a fuzzy model with noisy input–output pairs \{y_{j-1}, y_j\} to approximate the true function (i.e., \(f(x) = 1.5x \exp(-x^2/4)\)). The training dataset consists of 200 input–output pairs and the testing time series has 400 input–output pairs generated from the true function.

A set of four different fuzzy filters (say, \(m_1, m_2, m_3\), and \(m_4\)), which is of the type discussed in Appendix A, which define 1-D-clustering-based membership functions on the input variable, was considered. The \(m_1, m_2, m_3\), and \(m_4\) define 3, 4, 5, and 6 number of membership functions, respectively. Algorithm 2, which is initialized per the suggestions of Remark 3, was used to infer the parameters of fuzzy-filters combination. The priors were taken as follows.

1) \(m_{0i}\) equal to the zero vector.
2) \(\Lambda_{0i}\) equal to the unity matrix.
3) \(a_0 = 10^6\) and \(b_0 = 10^{-6}\) (i.e., relatively noninformative priors for the uncertainty).
4) \(c_0\) equal to \(S\).

The matrix \(c_{0i}\) and vector \(h_{0i}\) were designed to incorporate the following constrains on membership functions.

1) Any two consecutive knots must be separated at least by a distance of 0.1.
2) The extreme left knot must be greater than the minimum value of input variable in training set and the extreme right one less than the maximum value of input variable in the training set.

The experiment was repeated 50 times on the independently generated datasets. As stated in Remark 4, each time the algorithm converged to choose the winner model. The minimum and maximum iterations of the algorithm during the experiments were 2 and 6, respectively. Fig. 1 plots the counts that a model was chosen as the winner one. The performance of the winner fuzzy model was evaluated by calculating the root-mean-square error (RMSE) on the testing dataset. The first row of Table I reports the mean and standard deviation of the results during 50 runs of the experiment.

B. Study 2

The second example has been taken from [46], where the aim is to approximate the following nonlinear function:

\[
f(x) = \left[ a_1 + a_2 \frac{x}{b} + a_3 \left( \frac{2x^2}{b^2} - 1 \right) \right] \exp \left( -\frac{x^2}{2b^2} \right)
\]

where \(a_1 = 1, a_2 = 2, a_3 = -2,\) and \(b = 1.8\). The training dataset consists of 100 data pairs \(\{x(j), y(j)\}_{j=1}^{100}\). Here, \(x(j)\) is a random number generated from a uniform distribution on \([-10, 10]\) and \(y(j) = f(x(j)) + n_j\), where \(n_j\) is a uniform random number on \([-2.5, 2.5]\). The approximation quality was measured in terms of RMSE on uniform grid of 201 points on \([-10, 10]\).

A set of four different fuzzy filters (say, \(m_1, m_2, m_3,\) and \(m_4\)) of the type discussed in Appendix B, was considered. Following the approach of [46], fuzzy \(c\)-means clustering was used to initialize the membership functions. The \(m_1, m_2, m_3,\) and \(m_4\) define 4, 5, 6, and 7 membership functions, respectively. Algorithm 2 was used to infer the parameters of fuzzy-filter combination. The priors were same as in study 1, except
that \( m_0^{s_i} = m_{\text{single}}^{s_i} \), and \( \Lambda_0^{s_i} = \Lambda_{\text{single}}^{s_i} \), where the parameters \((\Lambda_{\text{single}}^{s_i}, m_{\text{single}}^{s_i})\) were returned by Algorithm 3 taking \( m_0^{s_i} \) equal to the zero vector and \( \Lambda_0^{s_i} \) equal to the unity matrix. The tuning of membership functions was constrained as follows:

1) The variances of Gaussian membership functions should take their values within 50\%–150\% of their initial values obtained by clustering.

2) The centers of Gaussian membership functions should take their values within \( \pm \sqrt{\text{variance of their initial values}} \) obtained by clustering.

As in [46], the experiment was repeated independently 50 times. The number of iterations of Algorithm 2 was observed to be varying from 3 to 164 with an average equal to 13.66. Fig. 2 and Table II show the obtained results.

C. Study 3

A real-world example to model an ECG signal was taken from [46]. The ECG signal data belong to the MIT-BIH database. The aim is to identify a model that predicts the current signal value \( s(j) \) using the previous four values \( s(j - 1), s(j - 2), s(j - 3), \) and \( s(j - 4) \). The model inputs and output are defined as

\[
x(j) = [s(j - 1)s(j - 2)s(j - 3)s(j - 4)]^T \in \mathbb{R}^4
\]

\[
y(j) = s(j).
\]

As in [46], the fuzzy model was trained with 450 data pairs and tested on another 5000 pairs.

Algorithm 3 was used to infer the parameters of a fuzzy filter (of the type given in Appendix B) that defines three different clusters on the input data. Following the method given in [46], the initial parameters of the membership functions were obtained from fuzzy clustering of the input training data. The tuning of membership functions was constrained in the same way as in study 2. The priors were taken as follows.

1) \( m_0 \) is equal to the zero vector.

2) \( \Lambda_0 \) is equal to the unity matrix.

3) \( a_0 = 10^6 \) and \( b_0 = 10^{-6} \).

The algorithm converged after 25 iterations. The results of the experiment are shown in Table III.

D. Study 4

Let us consider a process model

\[
y = f(x_1, x_2) + n
\]

\[
f(x_1, x_2) = (1 - x_1 x_2)e^{-(x_1 + x_2)^2} - \cos(4x_1 x_2) + \log(1 + x_1 x_2)
\]

where \( x_1 \) and \( x_2 \) are chosen from a uniform distribution over \([-0.9, 0.9]\), and \( n \) is a random variable normally distributed with zero mean and some fixed variance. The aim is to filter the uncertainty \( n \) from \( y \) using a fuzzy model. A total of 500 pairs of input–output data \([x(j) = [x_1(j) x_2(j)]^T, y(j)]\) were extracted to infer the parameters of a fuzzy filter. Table IV lists the details of the considered fuzzy models for the purpose of filtering. The performance of a filter was measured by calculating the energy of filtering errors defined as

\[
\text{FE} = \sum_{j=1}^{500} |f(x_1(j), x_2(j)) - G^T(x(j), \theta^*)m|^2
\]

where \((m, \theta^*)\) are returned by Algorithm 3. Algorithm 3 was used to infer the parameters of each of the six filters with the same priors and constraints as in study 1. The nonlinear optimization problem was solved using the MATLAB algorithm “fmincon” with its default settings regarding the maximum number of iterations and other parameters.

For a fixed variance of \( n \), Algorithm 3 was run several times on the different independently generated 500 pairs of inputs–output data. The results of 40 independent runs of Algorithm 3 are shown in Figs. 3 and 4. Fig. 3 shows, for each model, the scatter plots between filtering errors energy and negative free energy. Following inferences can be made from Figs. 3 and 4.

1) The higher values of filtering errors energy correspond to the lower values of negative free energy and vice versa. This demonstrates the negative-free-energy-based models comparison capability of the approach.

2) \( m^1 \), as seen from Fig. 4(a), is the most-suitable model in the case of lower magnitude uncertainties, while \( m^4 \),

### Table II

<table>
<thead>
<tr>
<th>method</th>
<th>number of fuzzy rules</th>
<th>number of model parameters</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithm 2</td>
<td>6.34 ± 0.7453</td>
<td>25.36 ± 2.9813</td>
<td>0.4737 ± 0.1022</td>
</tr>
<tr>
<td>Best known result from [46]</td>
<td>8</td>
<td>32</td>
<td>0.4836 ± 0.0820</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>method</th>
<th>number of fuzzy rules</th>
<th>number of model parameters</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithm 2</td>
<td>3</td>
<td>39</td>
<td>0.0150</td>
</tr>
<tr>
<td>Best known result from [46]</td>
<td>8</td>
<td>104</td>
<td>0.0214</td>
</tr>
</tbody>
</table>

### Table IV

<table>
<thead>
<tr>
<th>model</th>
<th>number of membership functions for each input</th>
<th>membership functions type (grid partitioning)</th>
<th>model order</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m^1 )</td>
<td>5</td>
<td>triangular</td>
<td>zero</td>
</tr>
<tr>
<td>( m^2 )</td>
<td>4</td>
<td>gaussian</td>
<td>zero</td>
</tr>
<tr>
<td>( m^3 )</td>
<td>3</td>
<td>clustering</td>
<td>zero</td>
</tr>
<tr>
<td>( m^4 )</td>
<td>2</td>
<td>triangular</td>
<td>zero</td>
</tr>
<tr>
<td>( m^5 )</td>
<td>2</td>
<td>gaussian</td>
<td>zero</td>
</tr>
<tr>
<td>( m^6 )</td>
<td>2</td>
<td>clustering</td>
<td>zero</td>
</tr>
</tbody>
</table>
Fig. 3. Scatter plots between the values of filtering errors energy and negative free energy. (a) (Low uncertainty) $n \sim N(0, 0.05)$. (b) (High uncertainty) $n \sim N(0, 0.5)$.

Fig. 4. Bar plots of filtering-errors energy values obtained during 40 independent experiments. (a) (Low uncertainty) $n \sim N(0, 0.05)$. (b) (High uncertainty) $n \sim N(0, 0.5)$.

Fig. 5. Histogram of winner-model index for study 4. (a) (Low uncertainty) $n \sim N(0, 0.05)$. (b) (High uncertainty) $n \sim N(0, 0.5)$. 
due to its slightly better performance in terms of filtering errors energy, should be preferred in the case of higher magnitude uncertainties.

Algorithm 2 was run on each of the 40 independently generated datasets. Fig. 5 plots the counts that a model was chosen as the winner one.

Following remarks can be made on the aforementioned simulation studies.

1) A comparison between first and second row of Tables I–III demonstrates the effectiveness of our VB-based approach in terms of performance and model complexity.
2) Fig. 5 shows the effectiveness of the Algorithm 2 in choosing the right structure of the fuzzy model, i.e., m1 in the case of n ~ N(0, 0.05) and m2 in the case of n ~ N(0, 0.5).

VI. CONCLUDING REMARKS

This study has introduced a mixed Takagi–Sugeno filter whose antecedents are deterministic, while the consequents are random variables. The parameters of the fuzzy filter are inferred under VB framework. The desired features (i.e., “automated regularization,” “model comparisons,” and “handling uncertainty via incorporating statistical noise models”) of the VB framework have been exploited to suggest an algorithm that selects the most suitable fuzzy filter out of the set and infers its parameters. The simulation studies verify the feasibility of the method.

The limitations of our method are the following.

1) The algorithms, like EM algorithm, might have the chances of being trapped in a local maxima.
2) A closed-form expression to update nonlinear antecedent, unlike other parameters updates, is not available. This is computationally the most expensive step of the algorithms.

These two issues will be addressed in our future study. The scope for future research includes the following:

1) A deterministic robustness analysis of the Algorithm 3;
2) the mean-squared-error analysis of the Algorithm 3;
3) the generalization of the proposed algorithms to non-Gaussian noise models;
4) an interpretation of the distribution q(θ) as a multivariate membership function for consequents, and thus, constructing a functionally equivalent deterministic fuzzy filter.

The main motivation behind our study is derived from the idea of combining the uncertainties-handling capabilities of fuzzy-membership functions with that of statistical models. The authors are optimistic that this integrated framework (i.e., fuzzy modeling + statistical modeling) has much to offer in the field of computational intelligence and machine learning.

APPENDIX

TAKAGI–SUGENO FUZZY FILTER

Let us consider a Takagi–Sugeno fuzzy model (F_\text{S} : X \rightarrow Y) that maps n-dimensional real-input space (X = X_1 \times X_2 \times \cdots \times X_n) to a 1-D real line.

A. Grid Partitioning of Input Space

A rule of the model is represented as

\[
y_f = s_0 + \sum_{j=1}^{n} s_j x_j.
\]

Here, (x_1, \ldots, x_n) are the model input variables, y_f is the filtered output variable, (A_1, \ldots, A_n) are the linguistic terms that are represented by fuzzy sets, and (s_0, s_1, \ldots, s_n) are real scalars. Given a universe of discourse X_j, a fuzzy subset A_j of X_j is characterized by the following mapping:

\[
\mu_{A_j} : X_j \rightarrow [0, 1]
\]

where, for x_j \in X_j, \mu_{A_j} (x_j) can be interpreted as the degree or grade to which x_j belongs to A_j. This mapping is called as membership function of the fuzzy set. Let us define, for the jth input, P_j nonempty fuzzy subsets of X_j (which are represented by A_{j1}, A_{j2}, \ldots, A_{jP_j}) Let the ith rule of the rule base is represented as

\[
\text{IF } x_1 \text{ IS } A_{1i} \text{ AND } \cdots \text{ AND } x_n \text{ IS } A_{ni},
\]

\[
\text{THEN } y_f = s_{i0} + s_{i1} x_1 + \cdots + s_{in} x_n
\]

where A_{1i} \in \{A_{11}, \ldots, A_{P_{1i}}\}, A_{2i} \in \{A_{12}, \ldots, A_{P_{2i}}\}, etc. Now, the different choices of A_{1i}, A_{2i}, \ldots, A_{ni} leads to the K = \prod_{j=1}^{P_j} number of fuzzy rules. For a given input vector x = [x_1 \cdots x_n]^T \in \mathbb{R}^n, the degree of fulfillment of the ith rule, by modeling the logic operator “AND” using product, is given by

\[
g_i (x) = \prod_{j=1}^{n} \mu_{A_{ji}} (x_j).
\]

The output of the fuzzy model to input vector x is computed by taking the weighted average of the output provided by each rule

\[
y_f = \sum_{i=1}^{K} (s_{i0} + s_{i1} x_1 + \cdots + s_{in} x_n) g_i (x) \]

\[
\sum_{i=1}^{K} g_i (x)
\]

\[
= \sum_{i=1}^{K} (s_{i0} + s_{i1} x_1 + \cdots + s_{in} x_n) \prod_{j=1}^{P_j} \mu_{A_{ji}} (x_j) / \sum_{i=1}^{K} \prod_{j=1}^{P_j} \mu_{A_{ji}} (x_j)
\]

(6)

Let us define a real vector \theta such that the membership functions of any type (e.g., trapezoidal, triangular, etc.) 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) Triangular Membership Functions: Let

\[
\theta = \{t_{01}^i, t_{11}^i, \ldots, t_{(i-1)}^{P_i-2}, t_{i1}^{P_i-1}, \ldots, t_{n1}^{P_n}, \ldots, t_{1n}^{P_{n-2}}, t_{nn}^{P_n-1}\}
\]

such that for the ith input, t_{0i}^i < t_{i1}^i < \cdots < t_{(i-1)}^{P_i-2} < t_{i1}^{P_i-1} holds for all i = 1, \ldots, n. Now, P_i triangular membership functions for ith input (\mu_{A_{1i}}, \mu_{A_{2i}}, \ldots, \mu_{A_{Pi}}) can be defined as

\[
\mu_{A_{1i}} (x_i, \theta) = \max \left( 0, \min \left( 1, \frac{t_{i1}^i - x_i}{t_{i1}^i - t_{01}^i} \right) \right)
\]

\[
\mu_{A_{Pi}} (x_i, \theta) = \max \left( 0, \min \left( 1, \frac{x_i - t_{i1}^{P_i-2}}{t_{i1}^{P_i-2} - t_{i1}^{P_i-1}} \right) \right)
\]
for all $j = 2, \ldots, P_i - 1$

$$
\mu_{A_{P_i,j}}(x_i) = \max \left( 0, \min \left( \frac{x_i - t_i^{P_i-2}}{t_i^{P_i-1} - t_i^{P_i-2}}, 1 \right) \right).
$$

2) 1-D-Clustering-Criterion-Based Membership Functions:

Let

$$
\theta = (t_1^0, t_1^1, \ldots, t_n^{P_n-2}, t_n^{P_n-1}, \ldots, t_1^0, t_1^1, \ldots, t_n^{P_n-2}, t_n^{P_n-1})
$$

such that for $i$th input, $t_i^0 < t_i^1 < \cdots < t_i^{P_i-2} < t_i^{P_i-1}$ holds for all $i = 1, \ldots, n$. Let us consider the problem of assigning two different memberships (say $\mu_{A_{1,i}}$ and $\mu_{A_{2,i}}$) to a point $x_i$ such that $t_i^0 < x_i < t_i^1$, based on following clustering criterion:

$$
[\mu_{A_{1,i}}, (x_i), \mu_{A_{2,i}}(x_i)] = \arg \min_{[u_1, u_2]} \left[ u_1^2(x_i - t_i^0)^2 + u_2^2(x_i - t_i^1)^2, u_1 + u_2 = 1 \right].
$$

This results in

$$
\mu_{A_{1,i}}(x_i) = \frac{(x_i - t_i^1)^2}{(x_i - t_i^0)^2 + (x_i - t_i^1)^2}, \quad \text{and}
$$

$$
\mu_{A_{2,i}}(x_i) = \frac{(x_i - t_i^0)^2}{(x_i - t_i^0)^2 + (x_i - t_i^1)^2}.
$$

Thus, for the $i$th input, $P_i$ membership functions can be defined as

$$
\begin{align*}
\mu_{A_{1,i}}(x_i) &= \begin{cases} 1, & x_i \leq t_i^0 \\ \frac{(x_i - t_i^1)^2}{(x_i - t_i^0)^2 + (x_i - t_i^1)^2}, & t_i^0 \leq x_i \leq t_i^1 \\ 0, & \text{otherwise} \end{cases} \\
\mu_{A_{2,i}}(x_i) &= \begin{cases} 1, & x_i \geq t_i^{P_i-1} \\ \frac{(x_i - t_i^{P_i-2})^2}{(x_i - t_i^{P_i-2})^2 + (x_i - t_i^{P_i-1})^2}, & t_i^{P_i-2} \leq x_i \leq t_i^{P_i-1} \\ \frac{(x_i - t_i^{P_i-1})^2}{(x_i - t_i^{P_i-2})^2 + (x_i - t_i^{P_i-1})^2}, & t_i^{P_i-1} \leq x_i \leq t_i^1 \\ 0, & \text{otherwise} \end{cases}
\end{align*}
$$

for $j = 2, \ldots, P_i - 1$.

3) Gaussian Membership Functions: Let

$$
\theta = (t_1^0, t_1^1, \ldots, t_n^{P_n-2}, t_n^{P_n-1}, \ldots, t_1^0, t_1^1, \ldots, t_n^{P_n-2}, t_n^{P_n-1})
$$

such that for $i$th input, $t_i^0 < t_i^1 < \cdots < t_i^{P_i-2} < t_i^{P_i-1}$ holds for all $i = 1, \ldots, n$. Now, $P_i$ Gaussian membership functions for $i$th input can be defined as

$$
\mu_{A_{ij}}(x_i, \theta) = e^{-(x_i - t_i^{j-1})^2}, \quad j = 1, \ldots, P_i.
$$

For any choice of membership functions (which can be constructed from a vector $\theta$), (6) can be rewritten as function of $\theta$

$$
y_f = \sum_{i=1}^{K} (s_{i0} + s_{i1}x_1 + \cdots + s_{in}x_n)\tilde{G}_i(x, \theta)
$$

$$
\tilde{G}_i(x, \theta) = \frac{\prod_{j=1}^{P_i} \mu_{A_{ij}}(x_j, \theta)}{\sum_{i=1}^{K} \prod_{j=1}^{P_i} \mu_{A_{ij}}(x_j, \theta)}.
$$

Let us introduce the following notation:

$$
\alpha = \begin{bmatrix} s_{10} \\ s_{11} \\ \vdots \\ s_{1n} \\ s_{K0} \\ s_{K1} \\ \vdots \\ s_{Kn} \end{bmatrix}, \quad G(x, \theta) = \begin{bmatrix} \tilde{G}_1(x, \theta) \\ x_1\tilde{G}_1(x, \theta) \\ \vdots \\ x_n\tilde{G}_1(x, \theta) \\ \tilde{G}_K(x, \theta) \\ x_1\tilde{G}_K(x, \theta) \\ \vdots \\ x_n\tilde{G}_K(x, \theta) \end{bmatrix}.
$$

Now, we have

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

In this expression, $\theta$ is not allowed to be any arbitrary vector, since the elements of $\theta$ must, $\forall i = 1, \ldots, n$, ensure

$$
a_i \leq t_i^0 < t_i^1 < \cdots < t_i^{P_i-2} < t_i^{P_i-1} \leq b_i
$$

where $x_i \in [a_i, b_i]$.

These inequalities and any other membership-functions-related constraints (designed to incorporate a priori knowledge) can be written in the form of a matrix inequality $c\theta \geq h$. Hence, a Takagi–Sugeno-type fuzzy filter can be represented as

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

B. Fuzzy-Clustering-Based Partitioning of Input Space

Several studies have used fuzzy $c$-means (or its robust alternatives) to find clusters in the input space and, thus, obtaining the parameters of the membership functions. Such methods define multivariate membership functions, and corresponding to each cluster, there exists a fuzzy rule of the Takagi–Sugeno form given by

$$
R_i : \text{IF} \ x \ \text{IS} \ A_{i}, \ \text{THEN} \ y_f = s_{i0} + s_{i1}x_1 + \cdots + s_{in}x_n
$$

$i = 1, 2, \ldots, K$. The fuzzy set $A_i$ (with a membership function $A_i(x) : R^n \rightarrow [0, 1]$) is typically defined with the Gaussian function

$$
\mu_{A_{i}}(x) = \prod_{j=1}^{n} \exp \left( -\frac{|x_j - t_j^{i,0}|^2}{2\sigma_j^{i,1}^2} \right)
$$

where $t_j^{i,0}$ is the center, and $\sigma_j^{i,1}$ is the dispersion of the membership function on $x_j$ defined by the $i$th cluster. The parameters of the membership functions (i.e., $t_j^{i,0}, \sigma_j^{i,1}; j = 1, \ldots, n; i = 1, \ldots, K$) can be obtained from fuzzy clustering of the input.
data [46]. Let the parameters of the membership functions be collected in a vector $\theta$, which is defined as

$$\theta = \{t_{11}^{1,0}, t_{11}^{1,1}, \ldots, t_{n1}^{1,0}, t_{n1}^{1,1}, \ldots, t_{1n}^{K,0}, t_{1n}^{K,1}, \ldots, t_{kn}^{K,0}, t_{kn}^{K,1} \}.$$ 

It is easy now to see that the fuzzy filter, like the grid-partitioning case, can be still functionally represented as

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

where

$$G(x, \theta) = \begin{bmatrix}
    s_{10} \\
    s_{11} \\
    \vdots \\
    s_{1n} \\
    s_{k0} \\
    s_{k1} \\
    \vdots \\
    s_{kn}
\end{bmatrix},$$

$$\alpha = \begin{bmatrix}
    x_1 \tilde{G}_1(x, \theta) \\
    x_1 \tilde{G}_1(x, \theta) \\
    \vdots \\
    x_n \tilde{G}_1(x, \theta) \\
    x_1 \tilde{G}_k(x, \theta) \\
    x_1 \tilde{G}_k(x, \theta) \\
    \vdots \\
    x_n \tilde{G}_k(x, \theta)
\end{bmatrix}.$$ 

Although the parameters of membership functions (i.e., elements of vector $\theta$) are obtained by fuzzy clustering, one may prefer a fine tuning of the elements of vector $\theta$ under a filtering-performance criterion. In this case, the tuning process can be constrained. A necessary constraint is that the variances of Gaussian membership functions must be greater than zero. Any type of constraint on the parameters of membership functions can be formulated as a matrix inequality $cb \geq h$.

**Acknowledgment**

The authors would like to thank S. Singh, Department of Computer Science, Himachal Pradesh University, for his support in the development of software for the algorithms.

**References**


Mohit Kumar 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, Delhi, India, in 2001, and the Ph.D. degree (summa cum laude) in electrical engineering and the “Dr.-Ing. Habil.” degree (with a venia legendi) in automation engineering from Rostock University, Rostock, Germany, in 2004 and 2009, respectively.

From 2001 to 2004, he was a Research Scientist with the Institute of Occupational and Social Medicine, Rostock. He is currently the Head of the research group “Life Science Automation—Technologies” with the Center for Life Science Automation, Rostock. His research interests include the modeling of complex and uncertain processes with applications to life science. He initiated intelligent fuzzy computing to build a mathematical bridge between artificial intelligence and real-world applications (for more details, see www.fuzzymodeling.com).

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

Until 1991, he was the head of the Section Analytical Chemistry, Academy of Sciences of the German Democratic Republic (GDR), Central Institute for Organic Chemistry. 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, Rostock University. From 1994 to 2000, he directed the Institution of Automation, Rostock University. Since 2003, he has also been the Vice President with the Center for Life Science Automation, Rostock. His current research interests include medical process measurement, lab automation, and smart systems and devices.

Regina Chappell received the Dipl.-Med. degree, the “Dr.Med.” degree in occupational medicine, and the “Dr.Med.Habil.” degree in occupational and sports medicine from Rostock University, Rostock, Germany, in 1980, 1984, and 2002, respectively.

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