FSS 3806 pp: 1--11 (col.fig.: nil) PROD. TYPE: COM ED: DK PAGN: Vish – SCAN: Mallikarjun

ARTICLE IN PRESS

ELSEVIER

Fuzzy Sets and Systems ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

FUZZY sets and systems

www.elsevier.com/locate/fss

Training fuzzy systems with the extended Kalman filter

Dan Simon *

Department of Electrical Engineering, Cleveland State University, Stilwell Hall Room 332, 1960 East 24th Street, Cleveland, OH 44115, USA

Received 14 June 2000; accepted 7 November 2001

Abstract

The generation of membership functions for fuzzy systems is a challenging problem. We show that for Mamdani-type fuzzy systems with correlation-product inference, centroid defuzzification, and triangular membership functions, optimizing the membership functions can be viewed as an identification problem for a nonlinear dynamic system. This identification problem can be solved with an extended Kalman filter. We describe the algorithm and compare it with gradient descent and with adaptive neuro-fuzzy inference system (ANFIS) based optimization of fuzzy membership functions. The methods discussed in this paper are illustrated on a fuzzy filter for motor winding current estimation, and are compared with Butterworth filtering. We demonstrate that the Kalman filter can be an effective tool for improving the performance of a fuzzy system.

Keywords: Learning; Filtering; Estimation; Training; Optimization; Gradient descent; Kalman filtering

1. Introduction

Various methods have been proposed over the years for the optimization of fuzzy membership functions. For example, many results have been presented in the literature of the use of genetic algorithms for fuzzy membership function optimization [23,31,38]. Other derivative-free methods that have been used include neural networks [5,10], evolutionary programming [12], cell mapping (a geometric method) [35], fuzzy equivalence relations [43], and heuristic methods [39]. Gradient descent is one of the derivative-based methods that has been used for shaping a fuzzy system's membership functions [7,31]. Other derivative-based methods that have been used to optimize fuzzy membership functions include the simplex method [6,15], least squares [36,41], backpropagation [40], and other numerical techniques [25].

Derivative-free methods have the advantage that they do not require the derivative of the objective function with respect to the membership function parameters. They are more robust than derivative-based methods with respect to finding a global minimum and with respect to their applicability to a wide range of objective functions and membership function forms. However, they typically tend to converge more slowly than derivative-based methods. Derivative-based methods have the advantage of fast convergence, but they tend to converge to local minima. In addition, due to their dependence on analytical derivatives, they are limited to specific objective functions, specific types of inference, and specific types of membership functions.

In this paper, we formulate a new derivative-based optimizer for fuzzy membership functions. Our new

* Tel.: +1-216-687-5407; fax: +1-216-687-5405.

E-mail addresses: simon@csvax.csuohio.edu, d.j.simon@csuohio.edu (D. Simon).

0165-0114/02/$ - see front matter
PII: S0165-0114(01)00241-X

formulation consists of an extended Kalman filter. Typically gradient descent has been the method of choice for derivative-based membership function optimization. However, some practical difficulties associated with gradient descent are slow convergence and ineffectiveness at finding a good solution [27,29,34,37]. This can be attributed to the first-order characteristic of gradient descent and the resultant neglect of correlations between outputs at successive iterations. This difficulty can be addressed by using second-order optimization algorithms that process and use additional information about the shape of the surface of the objective function. The particular second-order method that we explore here is Kalman filtering.

Kalman filters have been used with fuzzy logic in various ways. For instance, Kalman filters have been used to extract fuzzy rules from a given rule base [42]. They have been used to optimize the parameters of the Gaussian distribution transformation-based defuzzification (GTD) and polynomial transformation-based defuzzification (PTD) strategies [17]. Kalman filters have also been used to optimize the output function parameters of Takagi–Sugeno–Kang fuzzy systems [28]. Fuzzy logic has been used to compute the gains of a bank of parallel Kalman filters in order to combine their outputs [14], and fuzzy logic has been used to combine least mean square filtering with Kalman filtering for GPS-based navigation [24]. Fuzzy logic has also been used to tune the parameters of a Kalman filter [1,18,24]. However, this present paper is the first known use of the Kalman filter for the optimization of the membership functions of a fuzzy system.

For linear dynamic systems with white process and measurement noise, the Kalman filter is known to be an optimal estimator. For nonlinear dynamic systems with colored noise, the Kalman filter can be extended by linearizing the system around the current parameter estimates. This algorithm updates parameters in a way that is consistent with all previously measured data and generally converges in a few iterations. In the following sections, we describe how the extended Kalman filter can be applied to fuzzy system optimization. We demonstrate its performance on a fuzzy motor current estimator, and compare it with fuzzy system optimization using gradient descent and ANFIS. We further compare the resultant fuzzy filter with a Butterworth filter.

Section 2 discusses how gradient-based methods in general, and Kalman filters in particular, can optimize the membership functions of a fuzzy system. Section 3 contains experimental results and a comparison of the Kalman filter method with gradient descent and ANFIS, and Section 4 contains some concluding remarks and suggestions for further research.

2. Derivative-based optimization of fuzzy systems

Consider a fuzzy system that uses correlation-product inference. Assume that the input and output membership functions are symmetric triangles. The initial rule base and membership functions are constructed on the imprecise basis of experience, and trial and error. In spite of its importance, the generation of rule bases and membership functions remains a difficult and ill-defined task in the construction of fuzzy logic systems.

In general, we denote the centroid and half-width of the ith fuzzy membership function of the jth input by c_{ij} and b_{ij}, respectively. The membership function attains a value of 1 when the input is c_{ij}. As the input decreases from c_{ij}, the membership function value decreases linearly to 0 at $c_{ij} - b_{ij}$, and remains at 0 for all inputs less than $c_{ij} - b_{ij}$. As the input increases from c_{ij}, the membership function value decreases linearly to 0 at $c_{ij} + b_{ij}$, and remains at 0 for all inputs greater than $c_{ij} + b_{ij}$. The degree of membership of a crisp input x in the ith category of the jth input is therefore given by

$$f_{ij}(x) = \begin{cases} 1 + (x - c_{ij})/b_{ij} & \text{if } -b_{ij} \leqslant (x - c_{ij}) \leqslant 0, \\ 1 - (x - c_{ij})/b_{ij} & \text{if } 0 \leqslant (x - c_{ij}) \leqslant b_{ij}, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Similarly, for a single-output fuzzy system, we denote the centroid and half-width of the jth fuzzy membership function of the output by γ_j and β_j, respectively. For the special case where there are two inputs and one output, centroid defuzzification can be expressed as

$$\text{crisp output} = \frac{\sum_{j=1}^{n} m(\gamma_j)\gamma_j J_j}{\sum_{j=1}^{n} m(\gamma_j) J_j}, \tag{2}$$

where γ_j and J_j are the centroid and area of the jth output fuzzy membership function, and n is the number of fuzzy output sets. (Note that for the triangular membership functions that we are using, J_j is equal to the half-width (β_j) of the jth output fuzzy membership function.) The fuzzy output function $m(\gamma)$ is computed as

$$m(\gamma) = \text{fuzzy output function} = \sum_{i,k} m_{ik}(\gamma), \tag{3}$$

where $m_{ik}(\gamma)$ is defined as the consequent fuzzy output function when input 1 is in class i and input 2 is in class k.

$$m_{ik}(\gamma) = w_{ik} m_{oik}(\gamma), \tag{4}$$

$m_{oik}(\gamma)$ is the fuzzy function of the consequent that is activated when input 1 is in class i and input 2 is in class k, and w_{ik} is the activation level of that consequent.

$$w_{ik} = \min[f_{i1}(\text{input 1}), f_{k2}(\text{input 2})]. \tag{5}$$

Centroid defuzzification can easily be extended for the case where there are more than two inputs and one output, but the notation becomes (even more) cumbersome.

If the fuzzy membership functions are symmetric triangles as assumed in this paper, derivative-based methods can be used to optimize the centroids and the widths of the input and output membership functions. Consider an error function given by

$$E = \frac{1}{2N} \sum_{q=1}^{N} E_q^2, \tag{6}$$

$$E_q = \hat{y}_q - y_q, \tag{7}$$

where N is the number of training samples, y_q is the target value of the fuzzy system, and $\hat{y}_q$ is the output of the fuzzy system. We can optimize E by using the partial derivatives of E with respect to the centroids and half-widths of the input and output fuzzy membership functions. We can obtain expressions for these derivatives using (1) and following. Then, using the differentiation chain rule on (6), we can obtain expressions for the derivative of the error function with respect to the half-widths and centroids. We can then use those derivatives in an optimization scheme to minimize the error function with respect to the fuzzy membership function parameters. This idea was first suggested in [13] and was extended in [31,32]. See those references for detailed derivations and formulas for the derivatives.

2.1. Gradient descent

After the partial derivatives are computed as described above, the gradient descent rule can be used to update the independent variables from the kth iteration to the $(k+1)$ st iteration as follows:

$$\begin{aligned}
c_{ij}(k+1) &= c_{ij}(k) - \eta_c \frac{\partial E(k)}{\partial c_{ij}},\\
b_{ij}(k+1) &= b_{ij}(k) - \eta_b \frac{\partial E(k)}{\partial b_{ij}},\\
\gamma_i(k+1) &= \gamma_i(k) - \eta_\gamma \frac{\partial E(k)}{\partial \gamma_i},\\
\beta_i(k+1) &= \beta_i(k) - \eta_\beta \frac{\partial E(k)}{\partial \beta_i},
\end{aligned} \tag{8}$$

where η_c, η_b, η_γ, and η_β are gradient descent step sizes. Usually some method is used with the gradient descent algorithm to try to avoid convergence to a local minimum. For instance, once a local minimum is found, the solution can be randomly perturbed and the gradient descent algorithm can be restarted in an attempt to find a better local minimum.

2.2. The extended Kalman filter

Derivations of the extended Kalman filter are widely available in the literature [3,11]. In this section, we briefly outline the algorithm and show how it can be applied to fuzzy membership function optimization. Consider a nonlinear finite dimensional discrete time system of the form:

$$\begin{aligned}
x_{n+1} &= f(x_n) + w_n,\\
d_n &= h(x_n) + v_n,
\end{aligned} \tag{9}$$

where the vector x_n is the state of the system at time n, w_n is the process noise, d_n is the observation vector, v_n is the observation noise, and $f(\cdot)$ and $h(\cdot)$ are nonlinear vector functions of the state. Assume that the initial state x_0 and sequences $\{v_n\}$ and $\{w_n\}$ are Gaussian and independent from each other with

$$\mathsf{E}(x_0) = \bar{x}_0, \tag{10}$$

$$\mathsf{E}[(x_0 - \bar{x}_0)\,(x_0 - \bar{x}_0)^{\mathrm{T}}] = P_0, \tag{11}$$

$$\mathsf{E}(w_n) = 0, \tag{12}$$

$$\mathsf{E}(w_n w_l^{\mathrm{T}}) = Q\delta_{nl}, \tag{13}$$

$$\mathsf{E}(v_n) = 0, \tag{14}$$

$$\mathsf{E}(v_n v_l^{\mathrm{T}}) = R\delta_{nl}, \tag{15}$$

where $\mathsf{E}(\cdot)$ is the expectation operator and δ_{nl} is the Kronecker delta. The problem addressed by the extended Kalman filter is to find an estimate $\hat{x}_{n+1}$ of x_{n+1} given d_j $(j=0,\ldots,n)$.

If the nonlinearities in (9) are sufficiently smooth, we can expand them around the state estimate $\hat{x}_n$ using Taylor series to obtain

$$f(x_n) = f(\hat{x}_n) + F_n \times (x_n - \hat{x}_n) + \text{higher order terms},$$

$$h(x_n) = h(\hat{x}_n) + H_n^{\mathrm{T}} \times (x_n - \hat{x}_n) + \text{higher order terms}, \tag{16}$$

where

$$F_n = \left.\frac{\partial f(x)}{\partial x}\right|_{x=\hat{x}_n},$$

$$H_n^{\mathrm{T}} = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}_n}. \tag{17}$$

Neglecting the higher order terms in (16), the system in (9) can be approximated as

$$x_{n+1} = F_n x_n + w_n + \phi_n,$$

$$d_n = H_n^{\mathrm{T}} x_n + v_n + \varphi_n, \tag{18}$$

where

$$\phi_n = f(\hat{x}_n) - F_n \hat{x}_n,$$

$$\varphi_n = h(\hat{x}_n) - H_n^{\mathrm{T}} \hat{x}_n. \tag{19}$$

It can be shown that the desired estimate $\hat{x}_n$ can be obtained by the recursion

$$\hat{x}_n = f(\hat{x}_{n-1}) + K_n[d_n - h(\hat{x}_{n-1})],$$

$$K_n = P_n H_n (R_n + H_n^{\mathrm{T}} P_n H_n)^{-1},$$

$$P_{n+1} = F_n (P_{n-1} - K_n H_n^{\mathrm{T}} P_{n-1}) F_n^{\mathrm{T}} + Q_n, \tag{20}$$

K_n is known as the Kalman gain. In the case of a linear system, it can be shown that P_n is the covariance matrix of the state estimation error, and the state estimate $\hat{x}_{n+1}$ is optimal in the sense that it approaches the conditional mean $E[x_{n+1}|(d_0, d_1, \ldots, d_n)]$ for large n. For nonlinear systems the filter is not optimal and the estimates are only approximately conditional means.

2.3. Application to fuzzy systems

Inspired by the successful use of the Kalman filter for training neural networks [27] and for defuzzification strategies [17], we can apply a similar technique to the training of fuzzy systems. In general, we can view the optimization of fuzzy membership functions as a weighted least-squares minimization problem, where the error vector is the difference between the fuzzy system outputs and the target values for those outputs. Consider a fuzzy system that has L outputs. We use d to denote the target vector for the fuzzy system outputs, and $h(k)$ to denote the actual outputs at the kth iteration of the optimization algorithm.

$$d = [d_1 \quad \cdots \quad d_L]^{\mathrm{T}}$$

$$h(k) = [h_1(k) \quad \cdots \quad h_L(k)]^{\mathrm{T}}. \tag{21}$$

In order to cast the membership function optimization problem in a form suitable for Kalman filtering, we let the membership function parameters constitute the state of a nonlinear system, and we let the output of the fuzzy system constitute the output of the nonlinear system to which the Kalman filter is applied.

We will consider a two-input, one-output fuzzy system. This restriction is made only for notational convenience, and the results in this paper can be (conceptually) easily extended to an unlimited number of inputs and outputs. Consider a fuzzy system which has μ fuzzy sets for the first input, v fuzzy sets for the second input, and κ fuzzy sets for the output. As above we denote the centroid and half-width of the ith fuzzy membership function of the jth input by c_{ij} and b_{ij}, respectively, and we denote the centroid and half-width of the ith fuzzy membership function of the output by γ_i and β_i, respectively. The state of the nonlinear system can then be represented as

$$x = [b_{11}\ c_{11}\ \cdots\ b_{\mu 1}\ c_{\mu 1}\ b_{12}\ c_{12}\ \cdots \\ b_{v2}\ c_{v2}\ \beta_1\ \gamma_1\ \cdots\ \beta_\kappa\ \gamma_\kappa]^{\mathrm{T}}. \tag{22}$$

The vector x thus consists of all of the fuzzy membership function parameters arranged in a linear array. The nonlinear system model to which the Kalman filter can be applied is

$$x_{n+1} = x_n, \\ d_n = h(x_n), \tag{23}$$

where $h(x_n)$ is the fuzzy system's nonlinear mapping between the membership function parameters and the single output of the fuzzy system. In order to execute a stable Kalman filter algorithm, we need to add some artificial process noise and measurement noise to the system model. This is similar to the approach taken for neural network training using Kalman filters [27]. So we rewrite (23) as

$$x_{n+1} = x_n + w_n, \\ d_n = h(x_n) + v_n, \tag{24}$$

where w_n and v_n are artifically added noise processes. Now we can apply the Kalman recursion (20). In Section 2.2, $f(\cdot)$ is the identity mapping, d_n is the target output of the fuzzy system, and $h(\hat{x}_n)$ is the actual output of the fuzzy system given the current membership function parameters. H_n is the partial derivative of the fuzzy output with respect to the membership function parameters (which can be computed as described and referenced earlier in this paper), and F_n is the identity matrix. The Q_n and R_n matrices are tuning parameters which can be considered as the covariance matrices of the artificial noise processes w_n and v_n, respectively.

3. Experimental results

In this section, we describe and illustrate the use of Kalman filter training for the membership parameters of a fuzzy estimator for motor current windings. In order to implement an effective closed-loop current controller for a permanent magnet synchronous motor, we need an accurate estimate of the current in the motor windings [9]. The motor winding current consists of the current that is commanded by the motor drive and the current that is induced by the rotating stator [21]. Current estimation is thus an important and challenging problem for motor control. In order to implement a control system that responds in a timely manner, the current estimator should be causal. If it is noncausal, then the current controller will exhibit an unacceptable delay, thus resulting in degraded performance. Fuzzy logic was first proposed for motor current estimation in [30,32]. The fuzzy estimator structure that we use to obtain an estimate of the motor current y is given by

$$\hat{y}_k^- = \hat{y}_{k-1}^+ + T\widehat{y'}_{k-1}, \tag{25}$$

$$\hat{y}_k^+ = \hat{y}_k^- + g(z_k, \hat{y}_k^-), \tag{26}$$

where $\hat{y}_k^-$ denotes the estimate of y at time k *before* the measurement at time k is processed (the *a priori* estimate), and $\hat{y}_k^+$ denotes the estimate of y at time k *after* the measurement at time k is processed (the *a posteriori* estimate). T is the update period of the estimator, and z_k is the noisy measurement of the winding current. The estimate of the rate of change of the current ($\widehat{y'}$) is computed using the method of undetermined coefficients [4,32] as

$$\widehat{y'}_k = \frac{1}{T}\left[-\frac{1}{3}\hat{y}_{k-3}^+ + \frac{3}{2}\hat{y}_{k-2}^+ - 3\hat{y}_{k-1}^+ + \frac{11}{6}\hat{y}_k^+\right]. \tag{27}$$

The fuzzy correction mapping $g(\cdot)$ has two arguments:

$$(\text{input } 1)_k = z_k - \hat{y}_k^-, \tag{28}$$

$$(\text{input } 2)_k = (\text{input } 1)_k - (\text{input } 1)_{k-1}. \tag{29}$$

So the correction mapping depends on the difference between the measurement and the *a priori* estimate, and it also depends on the amount by which that difference has changed since the last time step. The fuzzy rule base for the mapping $g(\cdot)$ was chosen as shown in Table 1. The rule base has seven membership functions each for input 1, input 2, and the output (i.e., μ, v, and κ in (22) are each equal to seven). So the

Table 1
Rule base for fuzzy filter

		Input 2						
		NL	NM	NS	Z	PS	PM	PL
Input 1	NL	NL	NL	NM	NM	NS	NS	Z
	NM	NL	NM	NM	NS	NS	Z	PS
	NS	NM	NM	NS	NS	Z	PS	PS
	Z	NM	NS	NS	Z	PS	PS	PM
	PS	NS	NS	Z	PS	PS	PM	PM
	PM	NS	Z	PS	PS	PM	PM	PL
	PL	Z	PS	PS	PM	PM	PL	PL

NL = negative large, NM = negative medium, NS = negative small, Z = zero, PS = positive small, PM = positive medium, PL = positive large.

fuzzy estimator has a total of 21 membership functions. Each membership function is constrained to be a symmetrical triangle, so each membership function has two parameters (a centroid and a half-width). Thus the fuzzy estimator has a total of 42 parameters to be determined. These 42 parameters, arranged in a vector as shown in (22), comprise the state of the Kalman filter.

In order to implement the membership function optimization discussed in this paper, we collected motor winding currents with a digital oscilloscope at a rate of one sample every 200 μs. We then created a training waveform for the data by taking a simple symmetric (noncausal) 51-point moving average. Fig. 1 shows 2500 points of typical raw data and the smoothed training data. (The vertical axis of the figures is labeled “Volts” because the current was acquired with an analog-to-digital converter, which measured the current with a proportional voltage.) The output of the moving average is more than acceptable, but the moving average filter is noncausal and thus cannot be implemented in a real time motor control system.

The fuzzy current estimator was implemented and optimized using Visual Basic. Both the gradient descent and Kalman filter methods were used to optimize the fuzzy membership functions. The error function (6) consisted of the error between the noncausal moving average and the output of the causal fuzzy filter. The optimization schemes were initialized with the default membership functions shown in Fig. 2. (The two inputs and the output were all initialized with the same seven membership functions.)

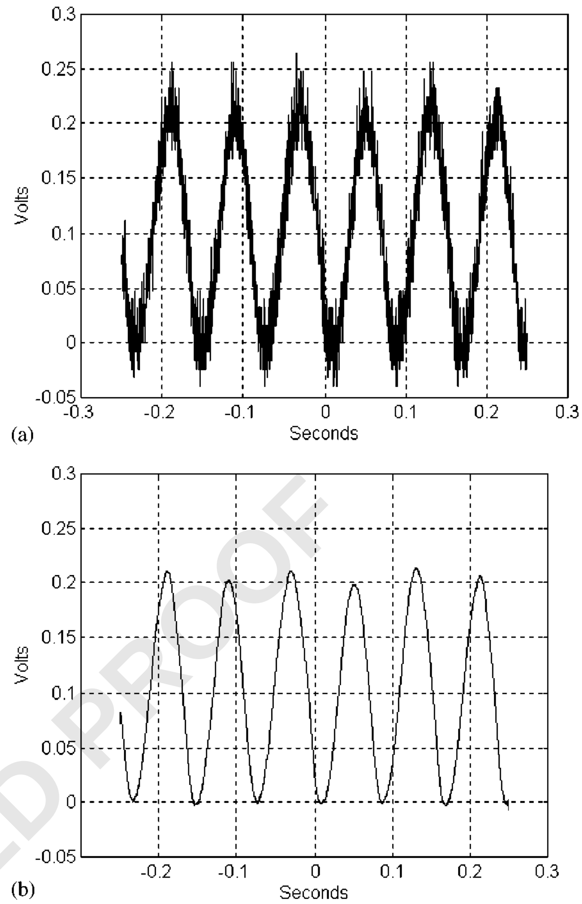

Fig. 1. Training data. (a) Unfiltered. (b) 51-Point moving average.

The gradient descent learning parameters η_c, η_b, η_γ, and η_β used in (8) were all initialized to 4. As gradient descent progressed, if the error function increased from one iteration to the next, the algorithm took a step back to the previous solution and adjusted the value of η_c, η_b, η_γ, or η_β in order to search for a minimum in a new direction. The Kalman filter parameters were set as follows: the matrix Q in (13) was set to $40I_{42}$ (where I_{42} is the 42×42 identity matrix); the matrix R in (15) was set to 50 (a scalar, since there is only one measurement for the Kalman filter); and the matrix P_0 in (11) was set to $200I_{42}$.

Fig. 3 depicts the progress of ANFIS [16] (as implemented in MATLAB’s Fuzzy Logic Toolbox), gradient descent, and Kalman filtering during optimization of the fuzzy filter membership functions. ANFIS was

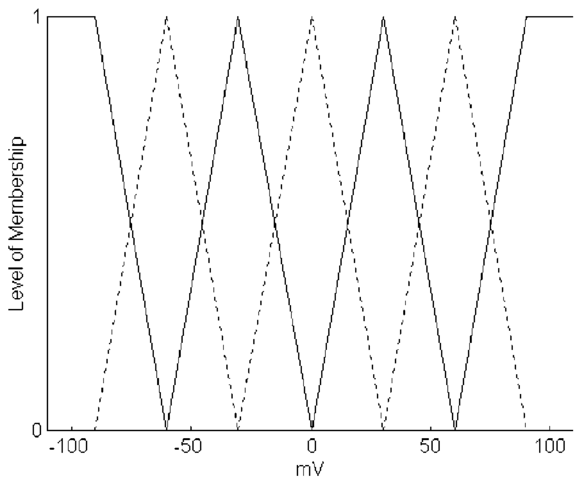

Fig. 2. Default membership functions for input 1, input 2, and output.

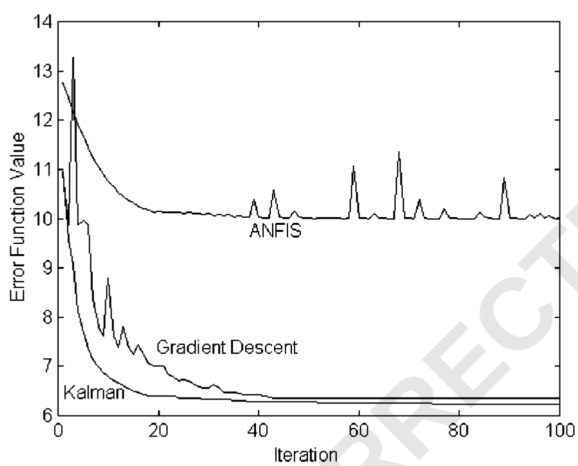

Fig. 3. Fuzzy logic system training progress.

included to provide a comparison with a qualitatively different optimization method. We do not place much emphasis on the ANFIS comparison, because the gradient descent and Kalman methods were applied to Mamdani systems with triangular membership functions, and ANFIS was applied to a Takagi–Sugeno system with Gaussian membership functions. (This is why the initial solution for gradient descent and Kalman filtering can be seen to have the same error, but the initial ANFIS solution has a different error.) Nevertheless, Fig. 3 does show that both gradient descent and Kalman filtering provide better optimization performance than ANFIS for this particular problem. We also ran ANFIS with generalized bell membership functions and triangular membership functions, and obtained results that were nearly indistinguishable from the ANFIS curve shown in Fig. 3. It can be seen from Fig. 3 that gradient descent and Kalman filtering yield comparable results, but the Kalman filter method converges more quickly and it also arrives at a better solution.

The computational requirements of gradient descent and Kalman filtering are about the same. Although the Kalman filter equations (20) are more complex than the gradient descent equations (8), the matrix inversion in (20) involves the inversion of only a 1×1 matrix (since the dynamic system has only one output). The majority of the computational effort for the two methods consists of calculating the derivatives of the objective function with respect to the membership parameters, and this calculation is the same for both optimization methods. The optimization methods were run on a Pentium 233 MHz PC running Visual Basic in design mode. The gradient descent method required about 9 s per iteration, and the Kalman filter method required about 12 s per iteration. It should be noted that in general, however, the computational effort of the Kalman filter is proportional to the square of the number of states, while the computational effort of the gradient descent method is directly proportional to the number of states. So the computational requirements of using a Kalman filter to train a fuzzy system with a large number of inputs and outputs may be an important factor. If so, the Kalman filter method described in this paper could be decoupled as described in [27] in order to ease the computational burden.

Fig. 4 shows the membership functions that resulted from the Kalman filter optimization. A comparision with Fig. 2 shows that the membership functions did not change dramatically during the optimization process, but the changes in the membership functions can be seen clearly, and those changes resulted in the error function decrease depicted in Fig. 3. The resultant membership functions shown in Fig. 4 are not sum normal; that is, they do not add up to one at each point in the domain. Sum normality may be desirable for a variety of reasons (e.g., less computational

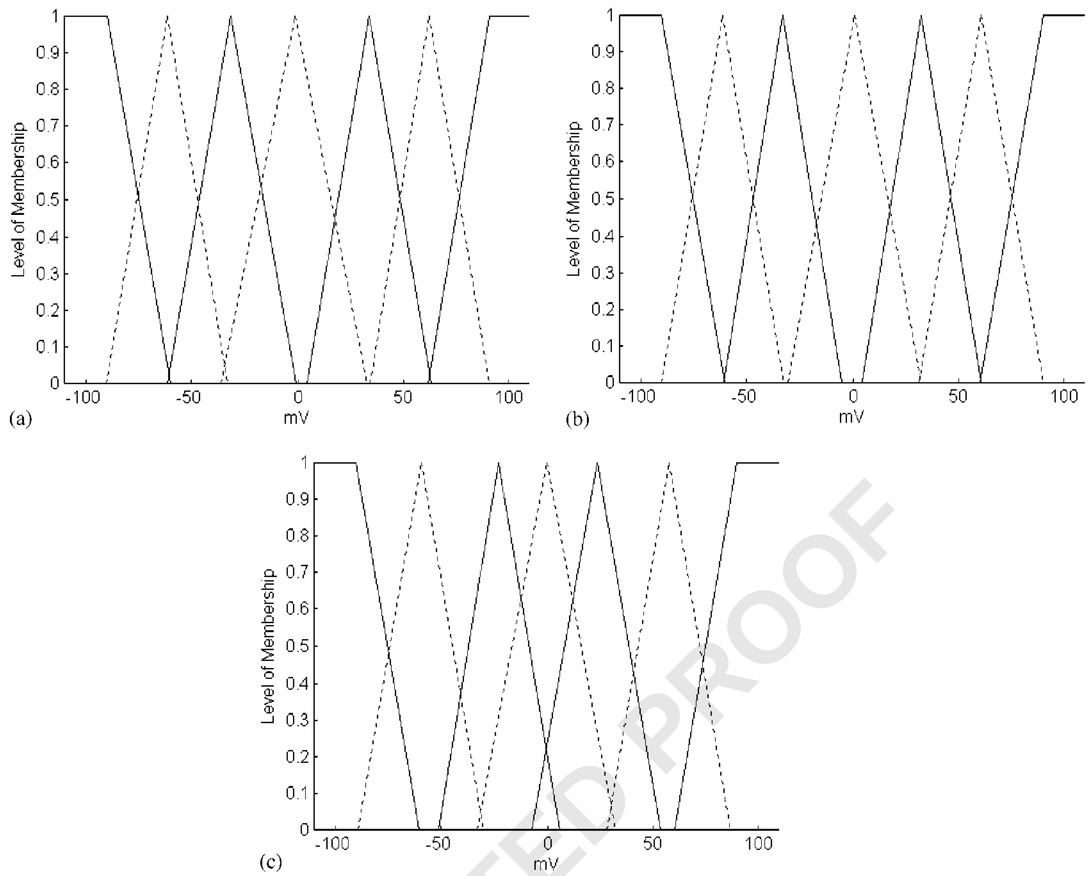

Fig. 4. Optimized membership functions for (a) input 1; (b) input 2; (c) output.

requirements during implementation, and greater amenabilty to rule base reduction). Follow-on work from this paper has incorporated sum normality constraints into the gradient descent and Kalman filtering approaches [33].

Fig. 5 shows some test data before and after being filtered with the fuzzy estimator. The filtered curve still has some high frequency chattering near the minima and maxima of the curve, and it is not nearly as smooth as the training data (Fig. 1); nevertheless, the data that came out of the fuzzy filter is noticably smoother than the raw data, and there is no time delay in the filtered data.

For purposes of comparision, Fig. 5 also shows the test data after being filtered with a Butterworth filter. The Butterworth filter was third order, as was the fuzzy filter (see (25)–(27)). The Butterworth filter has the nice property of being "optimally flat" in the passband [26]. The Butterworth filter actually gives smoother results than the fuzzy filter. However, the fuzzy filter has at least a couple of advantages over the Butterworth filter. First of all, a close look at Fig. 5 shows that the fuzzy filter has less time delay. The Butterworth filter results are delayed by about 16 time steps, or 3.2 ms, relative to the fuzzy filter results. Secondly, the passband of the Butterworth filter needs to be set *a priori*, whereas the parameters of the fuzzy filter depend only on the shape (and not on the frequency) of the waveform to be filtered. In any case, we do not place a lot of emphasis on the comparision between the fuzzy filter and the Butterworth filter, because the contribution of this paper is not to present a better filter, but rather to propose a better way of optimizing fuzzy membership functions.

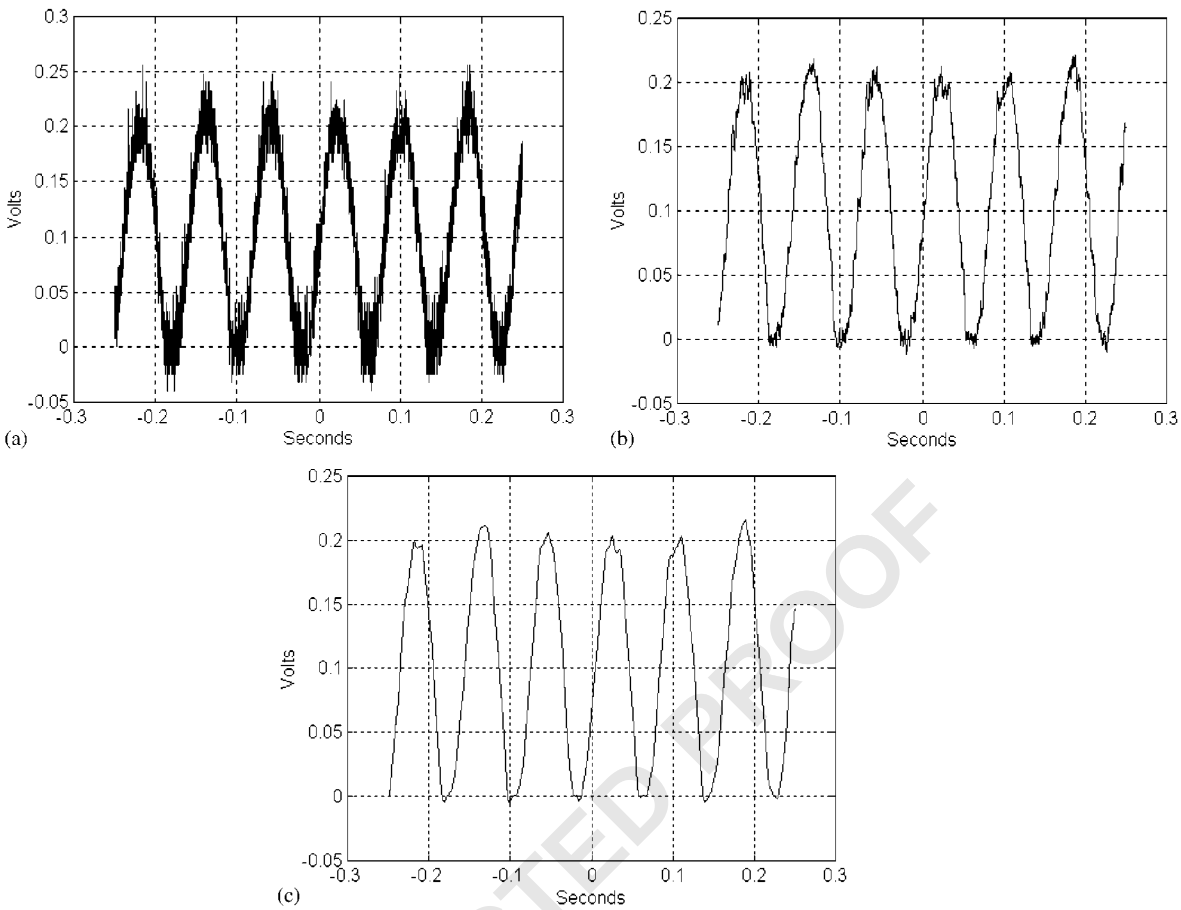

Fig. 5. Test data. (a) Unfiltered; (b) filtered with fuzzy estimator; (c) filtered with Butterworth filter.

4. Conclusion

We have shown that optimizing the membership functions of a fuzzy system can be viewed as a system identification problem for a nonlinear dynamic system. An extended Kalman filter can therefore be used to optimize the membership functions of a fuzzy logic system. The method was applied to a fuzzy filter for estimating winding currents in a permanent magnet synchronous motor. The Kalman filter converges more quickly, finds a better solution, and requires only slightly more computational effort than gradient descent. With this type of high dimension, nonlinear optimization problem it is difficult to say in general what type of optimization scheme works best. The contribution of this work is to show that Kalman filtering is a feasible method for training fuzzy logic systems in general.

Further work could focus on integrating the fuzzy filter discussed in this paper with a motor control scheme, using the Kalman filter for training fuzzy control systems, investigating the effect of the covariance matrices on the convergence of the Kalman filter, training fuzzy systems with membership functions that are other than symmetrically triangular, and constraining the Kalman filter so that the resultant membership functions are sum normal [33,44].

The theoretical strength of the Kalman filter has led to its application in hundreds of technologies, and this paper demonstrates that fuzzy system optimization is yet another fruitful application of Kalman filtering. It is thus recommended that Kalman filtering be given serious consideration as a training method for fuzzy logic systems.

References

[1] G. Abdelnour, S. Chand, S. Chiu, Applying fuzzy logic to the Kalman filter divergence problem, IEEE Internat. Conf. on Systems, Man and Cybernetics, Le Touquet, France, 1993, pp. 630–634.

[2] S. Akbar, R. Parra-Loera, Optimal self-tuning fuzzy controller, SPIE Conf. on Machine Tool, In-Line, and Robot Sensors and Controls, Philadelphia, 1995, pp. 130–137.

[3] B. Anderson, J. Moore, Optimal Filtering, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[4] K. Atkinson, An Introduction to Numerical Analysis, Wiley, New York, 1989.

[5] W. Barada, H. Singh, Generating optimal adaptive fuzzy-neural models of dynamical systems with applications to control, IEEE Trans. Systems Man Cybernet.—Part C 28 (1998) 371–391.

[6] J. Deskur, R. Muszynski, D. Sarnowski, Tuning and investigation of combined fuzzy controller, Internat. Conf. on Power Electronics and Variable Speed Drives, London, 1998, pp. 353–357.

[7] B. Demaya, R. Palm, S. Boverie, A. Titli, Multilevel qualitative and numerical optimization of fuzzy controller, IEEE Internat. Conf. on Fuzzy Systems, Yokohama, Japan, 1995, pp. 1149–1154.

[8] W. Eradus, H. Scholten, A. Udink ten Cate, Optimized fuzzy inference system for oestrus detection in dairy cattle, Internat. Symp. on Neuro-Fuzzy Systems, Lausanne, Switzerland, 1996.

[9] D. Feucht, Sensorless start-up positioning of brushless DC motors, PCIM Mag. 19 (1993) 24–27.

[10] M. Figueiredo, F. Gomide, Design of fuzzy systems using neurofuzzy networks, IEEE Trans. Neural Networks 10 (1999) 815–827.

[11] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, MA, 1974.

[12] J. Goddard, R. Parrazales, I. Lopez, A. de Luca, Rule learning in fuzzy systems using evolutionary programs, IEEE Midwest Symp. on Circuits and Systems, Ames, Iowa, 1996, pp. 703–709.

[13] F. Guély, P. Siarry, Gradient descent method for optimizing various fuzzy rule bases, IEEE Internat. Conf. on Fuzzy Systems, San Francisco, 1993, pp. 1241–1246.

[14] C. Hsiao, Modified gain fuzzy Kalman filtering algorithm, JSME Internat. J. Ser. C 42 (1999) 363–368.

[15] M. Jacomet, A. Stahel, R. Walti, On-line optimization of fuzzy systems, Inform. Sci. 98 (1997) 301–313.

[16] J. Jang, C. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice-Hall, Englewood Cliffs, NJ, 1997.

[17] T. Jiang, Y. Li, Generalized defuzzification strategies and their parameter learning procedures, IEEE Trans. Fuzzy Systems 4 (1996) 64–71.

[18] K. Kobayashi, K. Cheok, K. Watanabe, Estimation of the absolute vehicle speed using fuzzy logic rule-based Kalman filter, American Control Conf., Seattle, 1995, pp. 3086–3090.

[19] B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[20] R. Kovacevic, Y. Zhang, Neurofuzzy model-based weld fusion state estimation, IEEE Control Systems Mag. 17 (1997) 30–42.

[21] P. Krause, O. Wasynczuk, Electromechanical Motion Devices, McGraw-Hill, New York, 1989.

[22] J. Lalk, Intelligent Adaptation of Kalman filters using fuzzy logic, IEEE Conf. Fuzzy Systems (1994) 744–749.

[23] L. Magdalena, F. Monasterio-Huelin, Fuzzy logic controller with learning through the evolution of its knowledge base, Internat. J. Approx. Reasoning 16 (1997) 335–358.

[24] K. Mostov, A. Soloviev, Fuzzy adaptive stabilization of higher order Kalman filters in application to precision kinematic GPS, Institute of Navigation Meeting of the Satellite Division, Vol. 2, 1996, pp. 1451–1456.

[25] E. Nakamura, N. Kehtarnavez, Optimization of fuzzy membership function parameters, IEEE International Conference on Systems Man and Cybernetics, Vancouver, Canada, 1995, pp. 1–6.

[26] A. Oppenheim, A. Willsky, I. Young, Signals and Systems, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[27] G. Puskorius, L. Feldkamp, Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks, IEEE Trans. Neural Networks 5 (1994) 279–297.

[28] P. Ramaswamy, Mr. Riese, R. Edwards, K. Lee, Two approaches for automating the tuning process of fuzzy logic controllers, IEEE Conf. on Decision and Control, San Antonio, TX, 1993, pp. 1753–1758.

[29] S. Shah, F. Palmieri, M. Datum, Optimal filtering algorithms for fast learning in feedforward neural Networks, Neural Networks 5 (1992) 779–787.

[30] D. Simon, Fuzzy estimation of DC motor winding currents, North American Fuzzy Information Processing Society Conf., New York, 1999, pp. 859–863.

[31] D. Simon, H. El-Sherief, Fuzzy logic for digital phase-locked loop filter design, IEEE Trans. Fuzzy Systems 3 (1995) 211–218.

[32] D. Simon, Design and rule base reduction of a fuzzy filter for the estimation of motor currents, Internat. J. Approx. Reasoning 25 (2000) 145–167.

[33] D. Simon, Sum normal optimization of fuzzy membership functions, submitted for publication.

[34] S. Singhal, L. Wu, Training multilayer perceptrons with the extended Kalman algorithm, in: D. Touretzky (Ed.), Advances in Neural Information Processing Systems, Vol. 1, Morgan Kaufmann, San Mateo, CA, 1989, pp. 133–140.

[35] S. Smith, D. Comer, Automated calibration of a fuzzy logic controller using a cell state space algorithm, IEEE Control Systems Mag. 11 (1991) 18–28.

[36] M. Sugeno, K. Tanaka, Successive identification of a fuzzy model and its applications to prediction of a complex system, Fuzzy Sets and Systems 42 (1991) 315–334.

[37] J. Sum, C. Leung, G. Young, W. Kan, On the Kalman filtering method in neural-network training and pruning, IEEE Trans. Neural Networks 10 (1999) 161–166.

[38] H. Surmann, Genetic optimization of a fuzzy system for charging batteries, IEEE Trans. Ind. Electron. 43 (1996) 541–548.

[39] C. Tao, J. Taur, Design of fuzzy controllers with adaptive rule insertion, IEEE Trans. Systems Man Cybernetics—Part B 29 (1999) 389–397.

[40] L. Wang, J. Mendel, Back-propagation of fuzzy systems as nonlinear dynamic system identifiers, IEEE Internat. Conf. on Fuzzy Systems, San Diego, CA, 1992, pp. 1409–1418.

[41] L. Wang, Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1994.

[42] L. Wang, J. Yen, Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter, Fuzzy Sets and Systems 101 (1998) 353–362.

[43] R. Wu, S. Chen, A new method for constructing membership functions and fuzzy rules from training examples, IEEE Trans. Systems Man Cybernet—Part B 29 (1999) 25–40.

[44] Y. Yam, P. Baranyi, C. Yang, Reduction of fuzzy rule base via singular value decomposition, IEEE Trans. Fuzzy Systems 7 (1999) 120–132.