
ELSEVIER International Journal of Approximate Reasoning 25 (2000) 145–167

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Design and rule base reduction of a fuzzy filter for the estimation of motor currents

Dan Simon

Department of Electrical and Computer Engineering, Cleveland State University, Stilwell Hall Room 332, 1960 E. 24th Street, Cleveland, OH 44115-2425, USA

Received 1 September 1999; accepted 1 June 2000

Abstract

Fuzzy systems have been used extensively and successfully in control systems over the past few decades, but have been applied much less often to filtering problems. This is somewhat surprising in view of the dual relationship between control and estimation. This paper discusses and demonstrates the application of fuzzy filtering to motor winding current estimation in permanent magnet synchronous motors. Motor winding current estimation is an important problem because in order to implement effective closed-loop control, a good estimation of the current is needed. Motor winding currents are notoriously noisy because of electrical noise in the motor drive. We use a fuzzy system with correlation-product inference and centroid defuzzification for motor winding current estimation. With the assumption that the membership functions are triangular (but not necessarily symmetric), we then optimize the membership functions using gradient descent. Next we use singular value decomposition to reduce the rule base for the fuzzy filter. Rule base reduction can be important for fuzzy systems in those cases where the fuzzy system needs to be implemented in real time. This is especially true with regard to fuzzy filtering in a real time motor controller. The methods discussed in this paper are demonstrated on real motor winding currents that were collected with a digital oscilloscope. It is demonstrated that fuzzy techniques provide a feasible approach to motor current estimation, that gradient descent optimization improves the performance of the filter, and that rule base reduction results in a relatively small degradation of filter performance.

E-mail address: d.simon@ieee.org (D. Simon).

0888-613X/00/$ - see front matter
PII: S0888-613X(00)00054-2

Keywords: Fuzzy logic; Filtering; Estimation; Optimization; Gradient descent; Rule base reduction; Singular value decomposition; Motor

1. Introduction

The electrical windings of a permanent magnet synchronous motor are spaced on the stator (the fixed part of the motor) at regular angles. When excited with current, the windings produce magnetic fluxes that add vectorially to produce the stator flux. The controlling variables are the proportions of currents in the motor windings, which determine flux magnitude and orientation. Rotating rotor magnets produce the rotor flux and interact with the stator flux to produce torque. When the stator and rotor fluxes are aligned, the magnetic fields are in equilibrium at the minimum energy position and no torque is produced. When the stator and rotor fluxes are not aligned, the rotor magnets are pulled toward the stator electromagnets. This torque is maximum when the rotor flux is 90° behind the stator flux in the direction of motion. At this point the flux vectors are said to be *field-oriented* for maximum torque at a given current. This is also the most efficient operating region of the motor, because in this mode the power input to the mechanical side of the motor is maximized. For continuous rotation at the highest torque and efficiency, the stator flux is rotated in the desired direction of motion, keeping 90° ahead of the rotor flux. The stator flux is produced by controlling the current in the stator windings. Krause and Wasynczuk [1] provide a good overview of permanent magnet synchronous motors.

In order to implement an effective closed-loop current controller we need an accurate estimate of the current [2]. Current estimation is thus an important problem. It is also a challenging problem because the measured winding currents are strongly affected by electrical noise in the motor drive.

The motor's winding currents are generally shaped like sinusoids. Knowing this, we can formulate common sense fuzzy membership functions for use in a predictor–corrector type of estimator. The fuzzy winding current estimator is recursive and non-linear. Its inputs comprises past estimates, and present and past measurements. The use of fuzzy logic for motor winding current estimation was first explored by Simon [3].

We begin the fuzzy filter design process by gathering noisy experimental motor winding current data from a motor. Next we construct initial membership functions for a fuzzy current estimator on the basis of common sense and experience. We then use human expertise to guess the true motor currents underlying the experimental data. Finally we use these "true" motor currents as the basis with which to fine-tune the membership functions of the fuzzy current estimator. The membership functions are fine-tuned (i.e., optimized) using an iterative gradient-descent method.

After the membership functions are optimized, we can use singular value decomposition (SVD) to reduce the rule set of the fuzzy estimator. Rule base reduction is important in view of the challenge of real time implementation in a digital signal processor. The SVD method of rule reduction generates appropriate linear combinations of membership functions in order to obtain new membership functions for a reduced rule base.

The fuzzy estimator is applied to real motor winding currents in this paper. The results presented establish fuzzy estimation as a viable option for stator winding current estimation.

Section 2 gives a general algorithm for estimating a signal in the presence of noise using a fuzzy filter. Section 3 presents a technique for optimizing fuzzy membership functions using gradient descent, and Section 4 summarizes an algorithm that can be used to reduce a fuzzy rule base. Section 5 contains experimental results, and Section 6 contains some concluding remarks and suggestions for further research.

2. Fuzzy estimation

We begin with a standard discrete, time-invariant system given by

$$x_{k+1} = f(x_k) + v_k, \tag{1}$$

$$z_k = h(x_k) + w_k, \tag{2}$$

where k is the time index, x_k the state vector, z_k the measurement, and v_k and w_k are the noise processes. The problem of finding an estimate $\hat{x}_k$ for the state vector based on past and present measurements is known as the a posteriori filtering problem. One commonly used estimator architecture is the recursive predictor–corrector, given by

$$\hat{x}_k = \hat{f}(\hat{x}_{k-1}) + g(z_k, \hat{x}_{k-1}), \tag{3}$$

where $\hat{f}(\cdot)$ is an estimate of $f(\cdot)$, and $g(\cdot)$ is the correction function. The process model $f(\cdot)$ is often known, or it can be found using system identification methods. If $\hat{f}(\cdot)$ is available, only the correction mapping $g(\cdot)$ needs to be determined. Various analytic methods have been used for obtaining the correction mapping [4]. As an alternative to analytic methods, the correction mapping could be implemented as a fuzzy function [5].

2.1. Current estimation

Consider the problem of estimating a discrete-time signal $\{x\}$ corrupted by noise. The fuzzy estimator structure that we use to obtain an estimate of the signal is given by

$$\hat{x}_k^- = \hat{x}_{k-1}^+ + T\hat{v}_{k-1}, \tag{4}$$

$$\hat{x}_k^+ = \hat{x}_k^- + g(z_k, \hat{x}_k^-), \tag{5}$$

where $\hat{x}_k^-$ denotes the estimate of x at time k *before* the measurement at time k is processed, and $\hat{x}_k^+$ denotes the estimate of x at time k *after* the measurement at time k is processed. T is the update period of the estimator, z_k the noisy measurement of the winding current, and $\hat{v}$ is the estimate of current rate. (The determination of the rate estimate is discussed in Section 2.2.) The fuzzy correction mapping $g(\cdot)$ has two inputs

$$(\text{input } 1)_k = z_k - \hat{x}_k^-, \tag{6}$$

$$(\text{input } 2)_k = (\text{input } 1)_k - (\text{input } 1)_{k-1}. \tag{7}$$

So the correction mapping depends on the difference between the measurement and the a priori estimate, and the amount by which that difference has changed since the last time step. The output of the correction mapping is a fuzzy variable which is determined by correlation-product inference. The fuzzy rule base for the mapping $g(\cdot)$ was chosen as shown in Table 1. In this paper, triangular input and output membership functions are used.

The initial rule base and triangular membership functions were constructed on the imprecise basis of experience, and trial and error. An appropriate initial knowledge base is critical, because without an initial knowledge we cannot proceed further with any optimization schemes. In spite of its importance, the generation of initial knowledge remains a difficult and ill-defined task in the construction of fuzzy logic systems.

In general, we denote the centroid and the two half-widths of the ith fuzzy membership function of the jth input by c_{ij}, b_{ij}^-, and b_{ij}^+. The membership function attains a value of 1 when the input is c_{ij}. As the input decreases from c_{ij}, the membership function reaches a value of 0 at $c_{ij} - b_{ij}^-$. As the input

Table 1
Rule base for fuzzy filter[a]

Input 1	Input 2						
	NL	NM	NS	Z	PS	PM	PL
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

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

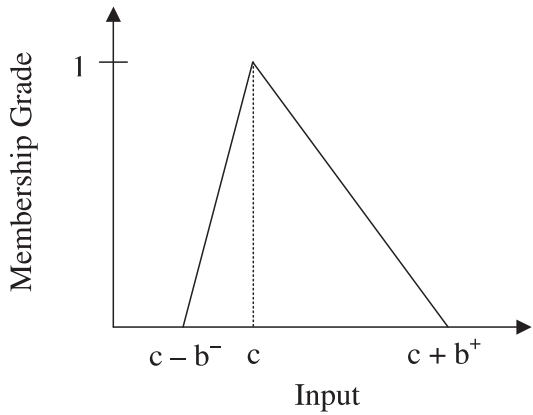

Fig. 1. Triangular membership function.

increases from c_{ij}, the membership function reaches a value of 0 at $c_{ij} + b_{ij}^{+}$. A generic triangular membership function is shown in Fig. 1. The degree of membership of a crisp input x in the ith category of the jth input is given by

$$f_{ij}(x) = \begin{cases} 1 + (x - c_{ij})/b_{ij}^{-} & \text{if } -b_{ij}^{-} \leqslant (x - c_{ij}) \leqslant 0, \\ 1 - (x - c_{ij})/b_{ij}^{+} & \text{if } 0 \leqslant (x - c_{ij}) \leqslant b_{ij}^{+}, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The fuzzy output is mapped into a crisp numerical value using centroid defuzzification [6].

$$g(z_k, \hat{x}_k^{-}) = \frac{\sum_{j=1}^{n} m(y_j) y_j J_j}{\sum_{j=1}^{n} m(y_j) J_j}, \tag{9}$$

where y_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 one-half of the sum of the two half-widths of the jth output fuzzy membership function.) The fuzzy output function $m(y)$ is computed as

$$m(y) = \text{fuzzy output function} = \sum_{i,k} m_{ik}(y), \tag{10}$$

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

$$m_{ik}(y) = w_{ik} m_{oik}(y), \tag{11}$$

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

$$w_{ik} = \min[f_{i1}(\text{input } 1), f_{k2}(\text{input } 2)]. \tag{12}$$

See Eq. (8) for the definition of the $f(\cdot)$ functions.

2.2. Current rate estimation

One of the inputs to the fuzzy estimator discussed above is the current rate estimate $\hat{v}$ (see Eq. (4)). This estimate must be computed causally from motor winding current estimates using numerical differentiation, which is in itself a challenging task. We will somewhat arbitrarily assume that we have the present and past three current estimates available. With this in mind, we use the method of undetermined coefficients [7] to obtain an expression for the current rate. The method of undetermined coefficients is a simple but elegant approach to deriving formulas for numerical differentiation. It consists of expanding $x(t)$ about available values of t using Taylor series expansions. For instance, if we have $x(t)$ available at times $t-\tau$, $t-2\tau$, and $t-3\tau$, then we write Taylor series expansions for $x(t-\tau)$, $x(t-2\tau)$, and $x(t-3\tau)$. We then write $x'(t) = A_1x(t) + A_2x(t-\tau) + A_3x(t-2\tau) + A_4x(t-3\tau)$, where the A_i's are "undetermined coefficients". We can solve for the A_i's by substituting the Taylor series expansions in the $x'(t)$ equation and simply setting the result equal to $x'(t)$. This approach gives us the following expression for the current rate

$$v(t) = \left[-\frac{1}{3}x(t-3\tau) + \frac{3}{2}x(t-2\tau) - 3x(t-\tau) + \frac{11}{6}x(t) \right] \Big/ \tau - \frac{193\tau^3}{72}x^{(4)}(\zeta), \tag{13}$$

where τ is a time step (some multiple of T in Section 2.1) to be determined later, and ζ is an unknown constant in $[t-3\tau, t]$. It is our objective in the remainder of this section to determine an appropriate time step τ. Denoting the error in the current estimate as $\tilde{x}$, we obtain

$$v(t) = \left[-\frac{1}{3}\hat{x}(t-3\tau) + \frac{3}{2}\hat{x}(t-2\tau) - 3\hat{x}(t-\tau) + \frac{11}{6}\hat{x}(t) \right] \Big/ \tau + \left[-\frac{1}{3}\tilde{x}(t-3\tau) + \frac{3}{2}\tilde{x}(t-2\tau) - 3\tilde{x}(t-\tau) + \frac{11}{6}\tilde{x}(t) \right] \Big/ \tau - \frac{193\tau^3}{72}x^{(4)}(\zeta). \tag{14}$$

So if we estimate v as

$$\hat{v}(t) = \left[-\frac{1}{3}\hat{x}(t-3\tau) + \frac{3}{2}\hat{x}(t-2\tau) - 3\hat{x}(t-\tau) + \frac{11}{6}\hat{x}(t) \right] \Big/ \tau \tag{15}$$

we obtain the following expression for the current rate estimation error:

$$\tilde{v}(t) = \frac{-193\tau^3}{72}x^{(4)}(\zeta) + \left[-\frac{1}{3}\tilde{x}(t-3\tau) + \frac{3}{2}\tilde{x}(t-2\tau) - 3\tilde{x}(t-\tau) + \frac{11}{6}\tilde{x}(t)\right]\Big/\tau. \tag{16}$$

Now if we treat the time functions in the above equation as random processes and make the simplifying assumption that $x^{(4)}(t)$ and $\tilde{x}(t)$ are independent, we obtain the following expression for the variance of the current rate estimation error:

$$E(\tilde{v}^2) = \left(\frac{193\tau^3}{72}\right)^2 E([x^{(4)}]^2) + \frac{530}{36\tau^2}E(\tilde{x}^2). \tag{17}$$

This equation shows that there is a tradeoff in using larger or smaller values of τ to estimate the current rate. The first part of Eq. (17) reflects the effect of using current estimates that are separated too much in time to estimate the current rate. As τ increases, the first term in Eq. (17) increases due to the uncorrelatedness between the current estimates that are being used to estimate the current rate. The second part of Eq. (17) reflects the effect of using current estimates that are too noisy to estimate the current rate. As τ decreases, the second term in Eq. (17) increases due to the noise in the current estimates that are being used to estimate the current rate. An appropriate value of τ needs to be used in Eq. (15) based on the relative magnitudes of the current dynamics and the current estimation error.

Based on our knowledge of the current waveforms, we will assume that we have a one-sigma current estimation error that corresponds to about 0.01 V. (The current is measured by an analog-to-digital converter [ADC] on the motor drive, so the acquired voltage is directly proportional to the motor winding current.) Again based on our knowledge of the current waveforms, we will assume that the standard deviation of the fourth derivative of the current is about 80 V/(ms)^4. Our ADC operates at a rate of 200 μs, so τ must be a multiple of 200 μs. We can then compute the rate estimation variance from Eq. (17) as a function of τ. The results are shown in Fig. 2. We see that the variance of the rate estimation error is minimized for $\tau = T$. This shows that the first term in Eq. (17) dominates the variance of the current rate estimation error. In other words, the current rate estimation error is dominated by the high dynamics of the motor current rather than the error in the motor current estimate. The rate estimation error is strongly dependent on τ. So we will use $\tau = T$ in Eq. (15) (where $T = 200$ μs is the ADC period) to estimate the current rate. This finding is critical to the success of the fuzzy estimator.

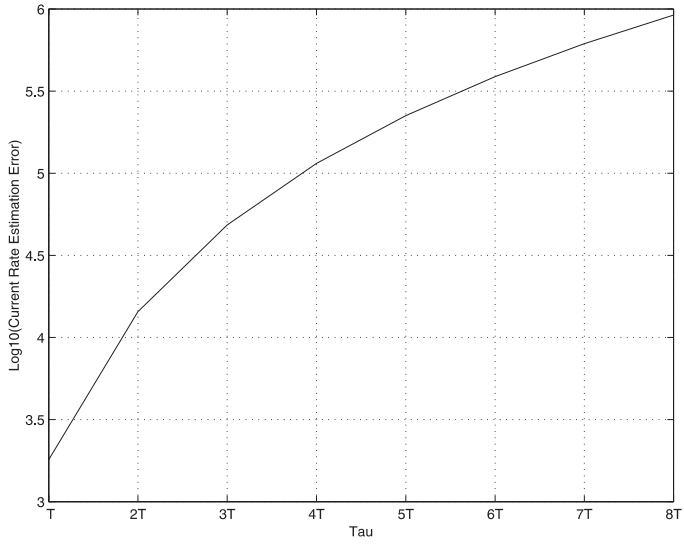

Fig. 2. Standard deviation of current rate estimation error.

3. Optimization

If the fuzzy membership functions are triangular as shown in Fig. 1, gradient descent can be used to optimize the centroids and the widths of the input and output membership functions. The work in this section builds on and extends similar efforts in [5,8]. Consider an error function given by

$$E = \frac{1}{2N} \sum_{q=1}^{N} E_q^2, \tag{18}$$

$$E_q \equiv \hat{x}_q - x_q, \tag{19}$$

where N is the number of training samples. 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.

3.1. Input centroids

Using the relationships of Eq. (8) we obtain

$$\frac{\partial E}{\partial c_{ij}} = \frac{1}{N} \sum_{q=1}^{N} E_q \frac{\partial \hat{x}_q}{\partial c_{ij}}, \tag{20}$$

$$\frac{\partial \hat{x}_q}{\partial c_{ij}} = \sum_{p=1}^{n} \frac{\partial \hat{x}_q}{\partial m_p} \frac{\partial m_p}{\partial c_{ij}} \quad [m_p \equiv m(y_p)], \tag{21}$$

$$\frac{\partial \hat{x}_q}{\partial m_j} = \frac{J_j(y_j - \hat{x}_q)}{\sum_{i=1}^{n} m_i J_i}, \tag{22}$$

$$\frac{\partial m_p}{\partial c_{ij}} = \sum_{k,l} r_{klp} \frac{\partial w_{kl}}{\partial c_{ij}}. \tag{23}$$

In Eq. (21), n is the number of fuzzy output sets. In Eq. (23), k goes from 1 to the number of fuzzy sets for input 1, and l goes from 1 to the number of fuzzy sets for input 2. Eq. (23) is valid for $p = 1, \ldots, n$. In Eq. (23), $r_{klp} = 1$ if $[(\text{input } 1) \in \text{class } k \text{ and } (\text{input } 2) \in \text{class } l] \Rightarrow (\text{output} \in \text{class } p)$, and 0 otherwise. In Eq. (23), $\partial w_{kl}/\partial c_{ij}$ is given as

$$\frac{\partial w_{kl}}{\partial c_{ij}} = \begin{cases} \partial f_{k1}/\partial c_{ij} & \text{if } f_{k1}(\text{input } 1) \leqslant f_{l2}(\text{input } 2), \\ \partial f_{l2}/\partial c_{ij} & \text{otherwise.} \end{cases}$$

The partials of the membership functions $f(\cdot)$ with respect to the input centroids are

$$\frac{\partial f_{k1}(\text{input } 1)}{\partial c_{i2}} = 0, \tag{24}$$

$$\frac{\partial f_{l2}(\text{input } 2)}{\partial c_{i1}} = 0, \tag{25}$$

$$\frac{\partial f_{k1}(\text{input } 1)}{\partial c_{i1}} = \begin{cases} -\delta_{ik}/b_{i1}^{-} & \text{if } c_{i1} - b_{i1}^{-} \leqslant \text{input } 1 \leqslant c_{i1}, \\ \delta_{ik}/b_{i1}^{+} & \text{if } c_{i1} \leqslant \text{input } 1 \leqslant c_{i1} + b_{i1}^{+}, \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

$$\frac{\partial f_{l2}(\text{input } 2)}{\partial c_{i2}} = \begin{cases} -\delta_{il}/b_{i2}^{-} & \text{if } c_{i2} - b_{i2}^{-} \leqslant \text{input } 2 \leqslant c_{i2}, \\ \delta_{il}/b_{i2}^{+} & \text{if } c_{i2} \leqslant \text{input } 2 \leqslant c_{i2} + b_{i2}^{+}, \\ 0 & \text{otherwise,} \end{cases} \tag{27}$$

where δ_{ik} is the Kronecker delta function ($\delta_{ik} = 1$ for $i = k$, 0 otherwise).

3.2. Input half-widths

Again using Eq. (8) it can be shown that

$$\frac{\partial E}{\partial b_{ij}^{-}} = \frac{1}{N} \sum_{q=1}^{N} E_q \frac{\partial \hat{x}_q}{\partial b_{ij}^{-}}, \tag{28}$$

$$\frac{\partial \hat{x}_q}{\partial b_{ij}^{-}} = \sum_{p=1}^{n} \frac{\partial \hat{x}_q}{\partial m_p} \frac{\partial m_p}{\partial b_{ij}^{-}}, \tag{29}$$

$$\frac{\partial m_p}{\partial b_{ij}^{-}} = \sum_{k,l} r_{klp} \frac{\partial w_{kl}}{\partial b_{ij}^{-}}, \tag{30}$$

$$\frac{\partial E}{\partial b_{ij}^{+}} = \frac{1}{N}\sum_{q=1}^{N} E_q \frac{\partial \hat{x}_q}{\partial b_{ij}^{+}}, \tag{31}$$

$$\frac{\partial \hat{x}_q}{\partial b_{ij}^{+}} = \sum_{p=1}^{n} \frac{\partial \hat{x}_q}{\partial m_p}\frac{\partial m_p}{\partial b_{ij}^{+}}, \tag{32}$$

$$\frac{\partial m_p}{\partial b_{ij}^{+}} = \sum_{k,l} r_{klp}\frac{\partial w_{kl}}{\partial b_{ij}^{+}}, \tag{33}$$

$$\frac{\partial \hat{x}_q}{\partial m_j} = \frac{J_j(y_j - \hat{x}_q)}{\sum_{i=1}^{n} m_i J_i}, \tag{34}$$

where $m_p \equiv m(y_p)$ and r_{klp} is given above in Eq. (23). $\partial w_{kl}/\partial b_{ij}^{-}$ and $\partial w_{kl}/\partial b_{ij}^{+}$ are given as

$$\frac{\partial w_{kl}}{\partial b_{ij}^{-}} = \begin{cases} \partial f_{k1}/\partial b_{ij}^{-} & \text{if } f_{k1}(\text{input 1}) \leqslant f_{l2}(\text{input 2}), \\ \partial f_{l2}/\partial b_{ij}^{-} & \text{otherwise,} \end{cases} \tag{35}$$

$$\frac{\partial w_{kl}}{\partial b_{ij}^{+}} = \begin{cases} \partial f_{k1}/\partial b_{ij}^{+} & \text{if } f_{k1}(\text{input 1}) \leqslant f_{l2}(\text{input 2}), \\ \partial f_{l2}/\partial b_{ij}^{+} & \text{otherwise.} \end{cases} \tag{36}$$

The partials of the membership functions $f(\cdot)$ with respect to the half-widths of the input fuzzy membership functions are given as

$$\frac{\partial f_{k1}(\text{input 1})}{\partial b_{i2}^{-}} = \frac{\partial f_{l2}(\text{input 2})}{\partial b_{i1}^{-}} = 0, \tag{37}$$

$$\frac{\partial f_{k1}(\text{input 1})}{\partial b_{i1}^{-}} = \begin{cases} \delta_{ik}[1 - f_{k1}(\text{input 1})]/b_{i1}^{-} & \text{if } c_{i1} - b_{i1}^{-} \leqslant \text{input 1} \leqslant c_{i1}, \\ 0 & \text{otherwise,} \end{cases} \tag{38}$$

$$\frac{\partial f_{l2}(\text{input 2})}{\partial b_{i2}^{-}} = \begin{cases} \delta_{il}[1 - f_{l2}(\text{input 2})]/b_{i2}^{-} & \text{if } c_{i2} - b_{i2}^{-} \leqslant \text{input 2} \leqslant c_{i2}, \\ 0 & \text{otherwise,} \end{cases} \tag{39}$$

$$\frac{\partial f_{k1}(\text{input 1})}{\partial b_{i2}^{+}} = \frac{\partial f_{l2}(\text{input 2})}{\partial b_{i1}^{+}} = 0, \tag{40}$$

$$\frac{\partial f_{k1}(\text{input 1})}{\partial b_{i1}^{+}} = \begin{cases} \delta_{ik}[1 - f_{k1}(\text{input 1})]/b_{i1}^{+} & \text{if } c_{i1} \leqslant \text{input 1} \leqslant c_{i1} + b_{i1}^{+}, \\ 0 & \text{otherwise,} \end{cases} \tag{41}$$

$$\frac{\partial f_{l2}(\text{input 2})}{\partial b_{i2}^{+}} = \begin{cases} \delta_{il}[1 - f_{l2}(\text{input 2})]/b_{i2}^{+} & \text{if } c_{i2} \leqslant \text{input 2} \leqslant c_{i2} + b_{i2}^{+}, \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

3.3. Output centroids

The partials of the objective function E with respect to the centroids of the output fuzzy membership functions are given as

$$\frac{\partial E}{\partial y_j} = \frac{1}{N}\sum_{q=1}^{N} E_q \frac{\partial \hat{x}_q}{\partial y_j}, \tag{43}$$

$$\frac{\partial \hat{x}_q}{\partial y_j} = \frac{m_j J_j}{\sum_{i=1}^{n} m_i J_i}. \tag{44}$$

3.4. Output half-widths

In this section, we will denote the centroids and half-widths of the jth output membership function as y_j, β_j^-, and β_j^+. In other words, an output membership function looks like Fig. 1 with c replaced by y, b^- replaced by β^-, and b^+ replaced by β^+. The partials of the objective function E with respect to the half-widths of the output membership functions are

$$\frac{\partial E}{\partial \beta_j^-} = \frac{1}{N}\sum_{q=1}^{N} E_q \frac{\partial \hat{x}_q}{\partial \beta_j^-}, \tag{45}$$

$$\frac{\partial E}{\partial \beta_j^+} = \frac{1}{N}\sum_{q=1}^{N} E_q \frac{\partial \hat{x}_q}{\partial \beta_j^+}, \tag{46}$$

$$\frac{\partial \hat{x}}{\beta_j^-} = \frac{m_j \sum_{i=1}^{n} m_i \beta_i^- (y_j - y_i)}{\left(\sum_{i=1}^{n} m_i \beta_i^-\right)^2}, \tag{47}$$

$$\frac{\partial \hat{x}}{\beta_j^+} = \frac{m_j \sum_{i=1}^{n} m_i \beta_i^+ (y_j - y_i)}{\left(\sum_{i=1}^{n} m_i \beta_i^+\right)^2}. \tag{48}$$

In the above equations, n is the number of output membership grades. Note from the above equations that if we start with symmetric output membership functions (i.e., $\beta_j^- = \beta_j^+$), then $\partial \hat{x}/\partial \beta_j^- = \partial \hat{x}/\partial \beta_j^+$. Therefore, if we start our optimization with symmetric output membership functions, we will always have symmetric output membership functions because the derivatives of the error function with respect to the lower and upper half-widths will always be equal.

3.5. Gradient descent

After the partial derivatives are computed as described in the above sections, the gradient descent rule is used to update the independent variables from one iteration to the next as follows:

$$c_{ij}(k+1) = c_{ij}(k) - \eta_c \frac{\partial E(k)}{\partial c_{ij}}, \tag{49}$$

$$b_{ij}^-(k+1) = b_{ij}^-(k) - \eta_b \frac{\partial E(k)}{\partial b_{ij}^-}, \tag{50}$$

$$b_{ij}^{+}(k+1) = b_{ij}^{+}(k) - \eta_b \frac{\partial E(k)}{\partial b_{ij}^{+}}, \tag{51}$$

$$y_j(k+1) = y_j(k) - \eta_y \frac{\partial E(k)}{\partial y_j}, \tag{52}$$

$$\beta_{ij}^{-}(k+1) = \beta_{ij}^{-}(k) - \eta_\beta \frac{\partial E(k)}{\partial \beta_{ij}^{-}}, \tag{53}$$

$$\beta_{ij}^{+}(k+1) = \beta_{ij}^{+}(k) - \eta_\beta \frac{\partial E(k)}{\partial \beta_{ij}^{+}}, \tag{54}$$

where η_c, η_b, η_y, and η_β are gradient descent step sizes.

4. Rule base reduction

Wang et al. [9] have recently used an SVD method to reduce the dimension of the input space of a fuzzy system, assuming that the membership functions are B-splines. The present paper, on the other hand, follows the work of Yam et al. [10] in applying SVD directly to the consequents of the rule set. This section briefly describes the rule base reduction algorithm used in this paper.

Consider a fuzzy rule base with two inputs a and b and a single fuzzy consequent r. We have n_a fuzzy sets for the first input and n_b fuzzy sets for the second input. In this paper, we assume that n_a and n_b are odd numbers. The more general case is treated in [10]

$$f_{i1}(a) \quad \text{and} \quad f_{j2}(b) \rightarrow r_{ij}, \tag{55}$$

where r_{ij} is the centroid of the output membership function corresponding to the (i,j)th fuzzy rule. To perform rule base reduction, we form the following $n_a \times n_b$ matrix

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n_b} \\ r_{21} & r_{22} & \cdots & r_{2n_b} \\ \vdots & \vdots & \vdots & \vdots \\ r_{n_a1} & r_{n_a2} & \cdots & r_{n_an_b} \end{bmatrix}. \tag{56}$$

We then perform singular value decomposition on R.

$$R = U\Sigma V^{\mathrm{T}}, \tag{57}$$

where U is $n_a \times n_a$ and V is $n_b \times n_b$. The magnitudes of the singular values in Σ indicate the relative importance of the corresponding columns of U and V in the formation of R. A close approximation to R can be obtained by keeping the n_r largest singular values.

$$R \approx U_r \Sigma_r V_r^{\mathrm{T}}, \tag{58}$$

where U_r is $n_a \times n_r$, Σ_r is $n_r \times n_r$, and V_r is $n_b \times n_r$. We then partition U and V into the parts that are to be retained and the parts that are to be discarded as follows:

$$U = [U_r \mid U_d], \tag{59}$$

$$V = [V_r \mid V_d]. \tag{60}$$

We next form $n_r \times n_r$ matrices Φ_U and Φ_V as follows:

$$\Phi_U = \begin{bmatrix} \sum_{i=1}^{n_a} U_r(i,1) & 0 & \cdots & 0 \\ 0 & \sum_{i=1}^{n_a} U_r(i,2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sum_{i=1}^{n_a} U_r(i,n_r) \end{bmatrix}, \tag{61}$$

$$\Phi_V = \begin{bmatrix} \sum_{i=1}^{n_b} V_r(i,1) & 0 & \cdots & 0 \\ 0 & \sum_{i=1}^{n_b} V_r(i,2) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sum_{i=1}^{n_b} V_r(i,n_r) \end{bmatrix}. \tag{62}$$

Actually, Φ_U and Φ_V can be chosen as any invertible matrices whose row sums are equal to the column sums of U_r and V_r, respectively [10]. We choose the above forms for ease of computation. We define the $(n_a - n_r) \times 1$ vector $\hat{U}_d$ and the $(n_b - n_r) \times 1$ vector $\hat{V}_d$ as

$$\hat{U}_d = \left[\sum_{i=1}^{n_a} U_d(i,1) \quad \cdots \quad \sum_{i=1}^{n_a} U_d(i,n_a - n_r) \right]^{\mathrm{T}}, \tag{63}$$

$$\hat{V}_d = \left[\sum_{i=1}^{n_b} V_d(i,1) \quad \cdots \quad \sum_{i=1}^{n_b} V_d(i,n_b - n_r) \right]^{\mathrm{T}}. \tag{64}$$

We next form the $n_a \times (n_r + 1)$ matrix S_U and the $n_b \times (n_r + 1)$ matrix S_V as follows:

$$S_U = [U_r \mid U_d \times \hat{U}_d] \times \begin{bmatrix} \Phi_U & 0_{n_r \times 1} \\ 0_{1 \times n_r} & 1 \end{bmatrix}, \tag{65}$$

$$S_V = [V_r \mid V_d \times \hat{V}_d] \times \begin{bmatrix} \Phi_V & 0_{n_r \times 1} \\ 0_{1 \times n_r} & 1 \end{bmatrix}, \tag{66}$$

where $0_{i \times j}$ is defined as the $i \times j$ matrix comprised of all zeros. We next form the $(n_r + 1) \times (n_r + 1)$ matrices N_U and N_V as follows:

$$\zeta_U \equiv \begin{cases} 1 & \text{if } \min_{i,j} S_U(i,j) \geqslant -1, \\ |\min_{i,j} S_U(i,j)|^{-1} & \text{otherwise,} \end{cases} \tag{67}$$

$$\zeta_V \equiv \begin{cases} 1 & \text{if } \min_{i,j} S_V(i,j) \geqslant -1, \\ |\min_{i,j} S_V(i,j)|^{-1} & \text{otherwise,} \end{cases} \tag{68}$$

$$N_U = \frac{1}{n_r + 1 + \zeta_U} \times \begin{bmatrix} 1+\zeta_U & 1 & \cdots & 1 \\ 1 & 1+\zeta_U & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1+\zeta_U \end{bmatrix}, \tag{69}$$

$$N_V = \frac{1}{n_r + 1 + \zeta_V} \times \begin{bmatrix} 1+\zeta_V & 1 & \cdots & 1 \\ 1 & 1+\zeta_V & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \cdots & 1+\zeta_V \end{bmatrix}. \tag{70}$$

We next form the $n_a \times (n_r + 1)$ matrix $\tilde{U}$ and the $n_b \times (n_r + 1)$ matrix $\tilde{V}$ as follows:

$$\tilde{U} = S_U N_U, \tag{71}$$

$$\tilde{V} = S_V N_V. \tag{72}$$

We next consider the n_a rows in $\tilde{U}$ and the n_b rows in $\tilde{V}$ as points in an $(n_r + 1)$-dimensional space.

We form an $(n_r + 1) \times (n_r + 1)$ matrix whose $n_r + 1$ rows represent points which comprise a convex hull that approximately bounds the n_a points represented in $\tilde{U}$. We likewise form an $(n_r + 1) \times (n_r + 1)$ matrix whose $n_r + 1$ rows represent points which comprise a convex hull that approximately bounds the n_b points represented in $\tilde{V}$. These two matrices are denoted as Q_U^{-1} and Q_V^{-1}, respectively. Finally we form the $n_a \times (n_r + 1)$ matrix $\bar{U}$, the $(n_r + 1) \times (n_r + 1)$ matrix $\bar{R}$, and the $n_b \times (n_r + 1)$ matrix $\bar{V}$ as follows:

$$\bar{U} = \tilde{U} Q_U, \tag{73}$$

$$\bar{V} = \tilde{V} Q_V, \tag{74}$$

$$\bar{R} = Q_U^{-1} N_U^{-1} \Phi_U^{-1} \Sigma_r \Phi_V^{-\mathrm{T}} N_V^{-\mathrm{T}} Q_V^{-\mathrm{T}}. \tag{75}$$

Now the reduced rule base can be defined. If our initial two-input rule base is

$$f_{i1}(a) \quad \text{and} \quad f_{j2}(b) \to r_{ij}, \tag{76}$$

where $i = 1, \ldots, n_a$ and $j = 1, \ldots, n_b$, then the membership functions for our reduced rule base are defined as

$$\bar{f}_{\bar{i}1}(a) = \sum_{i=1}^{n_a} f_{i1}(a)\bar{U}(i,\bar{i}) \quad (\bar{i} = 1, \ldots, n_r + 1), \tag{77}$$

$$\bar{f}_{\bar{j}2}(b) = \sum_{j=1}^{n_b} f_{j2}(b)\bar{V}(j,\bar{j}) \quad (\bar{j} = 1, \ldots, n_r + 1). \tag{78}$$

The centroids of the consequents for the reduced rule base are defined in the matrix $\bar{R}$. The reduced rule base has $(n_r + 1)^2$ rules instead of the original $n_a n_b$ rules.

5. Experimental results

The fuzzy estimator and optimizer discussed in this paper was implemented in Visual Basic and was used to filter the winding currents of a permanent magnet synchronous motor. The motor winding currents were collected with a digital oscilloscope at a rate of one sample every 200 μs. The gradient descent method was used to optimize the fuzzy membership functions. The training data for the gradient descent optimization consisted of a simple symmetric non-causal 51-point moving average. The fuzzy filter was causal and was implemented as described earlier in this paper. The error function of Eq. (18) consisted of the error between the non-causal moving average and the output of the causal fuzzy filter.

The rule base was defined as shown in Table 1. These are common sense rules, such as, if the error between the measured value and the extrapolated estimate is positive medium, and the change in error is zero, then change the estimate by a positive small amount. There was no attempt in this paper to optimize the rule base. This is an important area of current fuzzy systems research [11], but is not addressed in this paper.

The gradient descent learning parameters η_c, η_b, η_y, and η_β were all initialized to 4. There were seven membership functions for the two inputs and the output. The membership functions were constrained to be non-symmetric triangles, so the error function E in Eq. (18) was optimized with respect to 63 parameters – the centroids of each of the membership functions (21 total), and the two half-widths of each of the membership functions (42 total).

Fig. 3 shows the training data that was used for the gradient descent optimization. Fig. 3 shows 2500 samples of raw current and the output of the 51-point moving average that was used to optimize the fuzzy filter. (The vertical axis of the figures is labelled 'Volts' because the current is acquired with an ADC, which measures the current with a proportional voltage.) Fig. 4 shows the seven initial membership functions for the two inputs and the output. (The two inputs and the output were all initialized with the same seven membership functions.) Fig. 5 shows the decrease of the objective function as training progressed. The algorithm reached a local minimum after about 40 iterations.

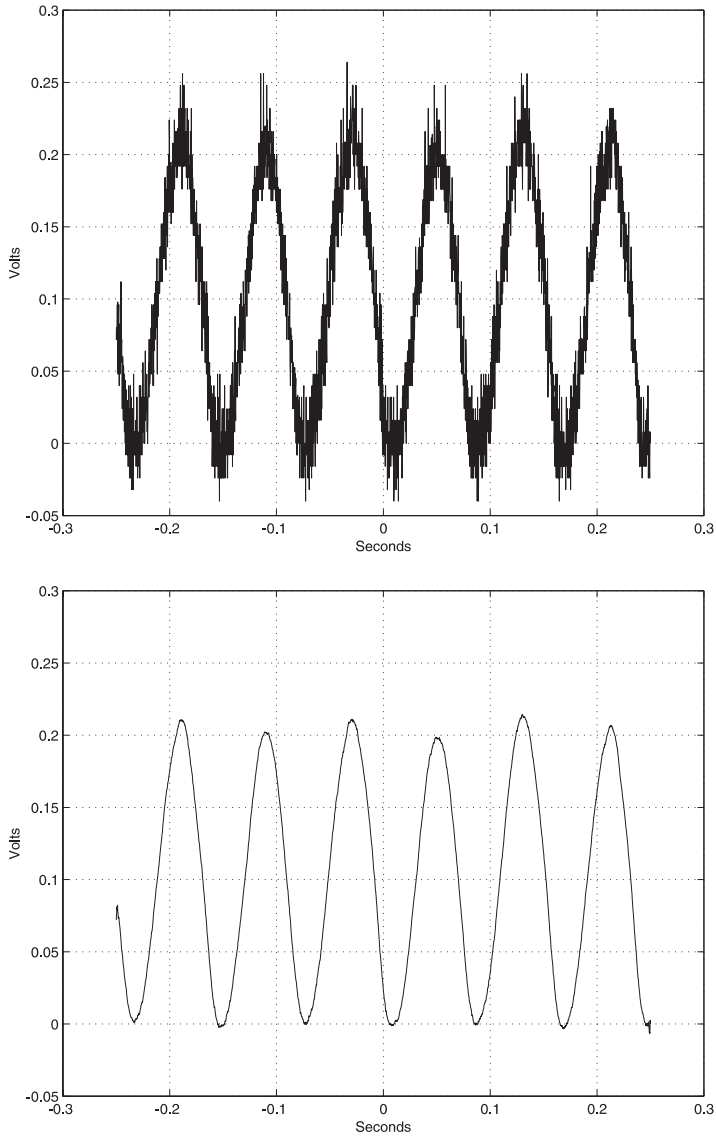

Fig. 3. Training data: (a) unfiltered; (b) 51-point moving average.

The optimization required about two minutes on a Pentium 233 MHz PC running Visual Basic in design mode. Fig. 6 shows the membership functions that resulted from the gradient descent optimization. A comparison with Fig. 4 shows that the membership functions did not change dramatically during the optimization process. But the changes in the membership functions can be

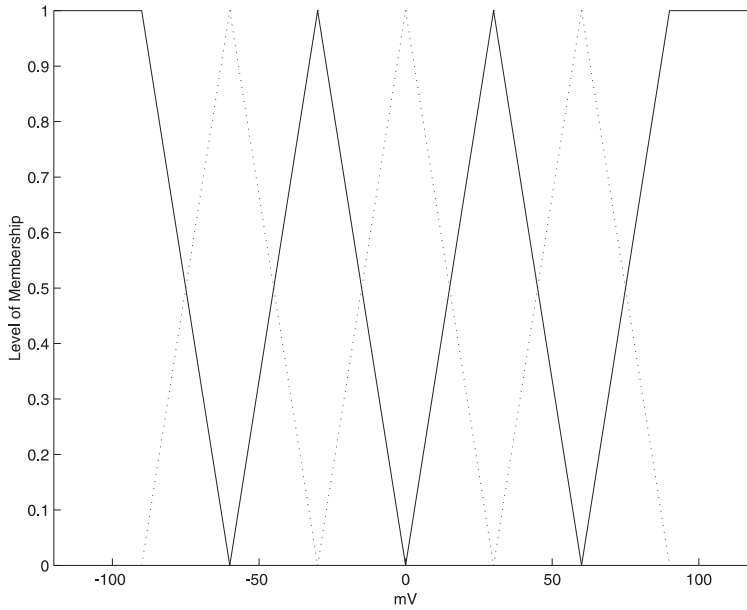

Fig. 4. Default membership functions for input 1, input 2, and output.

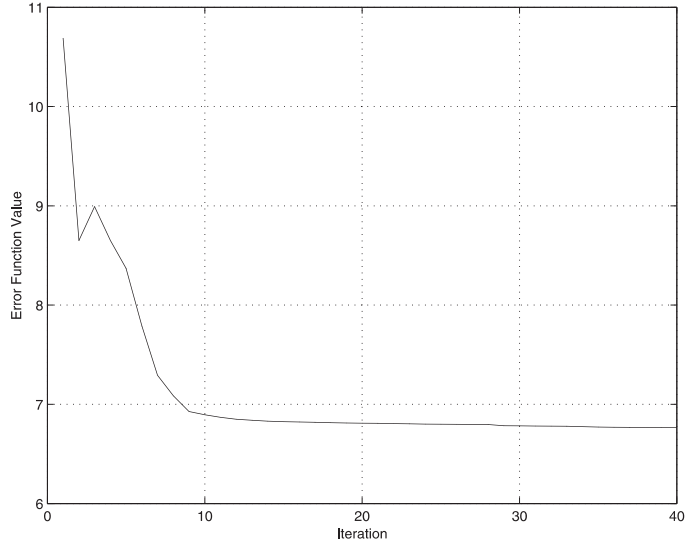

Fig. 5. Gradient descent training of membership functions.

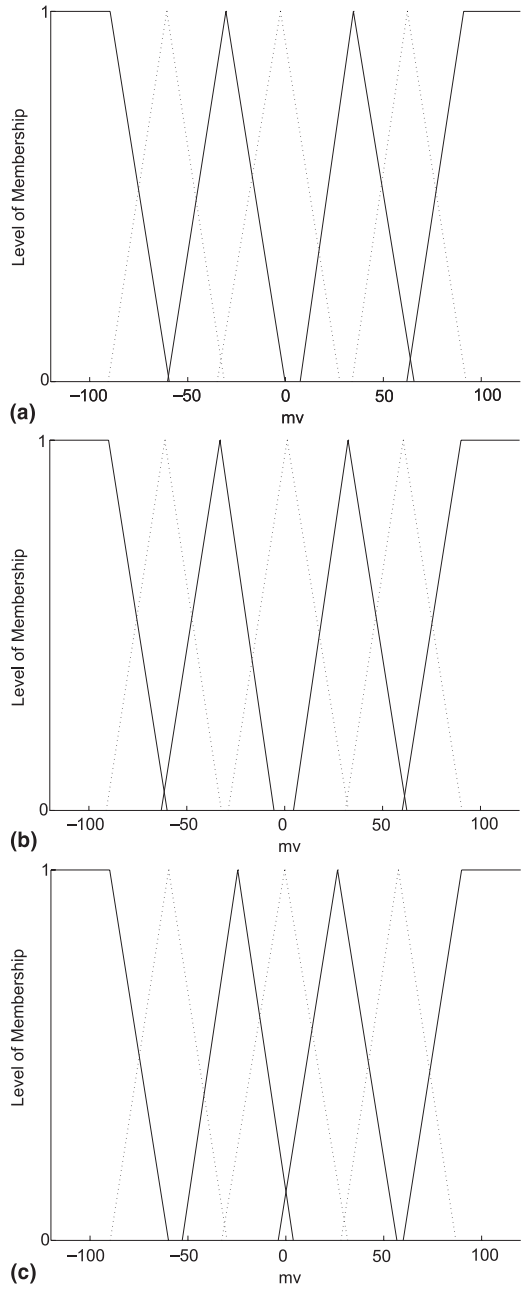

Fig. 6. Optimized membership functions for (a) input 1, (b) input 2, (c) output.

seen, and the membership functions for the two inputs became slightly asymmetric as a result of the optimization.

Fig. 7 shows the test data before and after being filtered with the fuzzy estimator. Comparison with Fig. 3 shows that the resultant curve is not as smooth as the moving average curve; nevertheless, the data that came out of the fuzzy filter is noticeably smoother than the raw data, and there is no visible time delay in the filtered data.

The rule base reduction scheme discussed in Section 4 was implemented in MATLAB and applied to the optimized fuzzy membership functions. The initial membership functions are shown in Fig. 6. We form a 7×7 R matrix based on Eq. (56), Table 1, and the centroids of the output membership functions shown in Fig. 6. The singular values of the resulting R matrix are $\{234, 232, 34, 34, 10, 10, 1\}$ mV. We choose to keep the two largest singular values and go through the algorithm described in Section 4. This results in three fuzzy sets each for the inputs instead of the original seven sets each. The Q^{-1} matrices were chosen using a graphical method described in [12]

$$Q_U^{-1} = \begin{bmatrix} \tilde{U}(1,1) & \tilde{U}(1,2) & \tilde{U}(1,3) \\ \tilde{U}(4,1) & \tilde{U}(4,2) & \tilde{U}(4,3) \\ \tilde{U}(7,1) & \tilde{U}(7,2) & \tilde{U}(7,3) \end{bmatrix}, \tag{79}$$

$$Q_V^{-1} = \begin{bmatrix} \tilde{V}(1,1) & \tilde{V}(1,2) & \tilde{V}(1,3) \\ \tilde{V}(4,1) & \tilde{V}(4,2) & \tilde{V}(4,3) \\ \tilde{V}(7,1) & \tilde{V}(7,2) & \tilde{V}(7,3) \end{bmatrix}. \tag{80}$$

The $\bar{R}$ matrix that results from the rule base reduction algorithm has five distinct values

$$\bar{R} = \begin{bmatrix} -0.09255 & -0.05447 & -0.0004669 \\ -0.05447 & -0.0004669 & 0.05429 \\ -0.0004669 & 0.05429 & 0.09255 \end{bmatrix}. \tag{81}$$

The new reduced membership functions are shown in Fig. 8, and the reduced rule base is shown in Table 2. Instead of the original 49 rules, we now have 9 rules in our rule base. Note from Fig. 8 that some of the membership functions are less than zero for some values of the input. This is somewhat non-intuitive, but the mathematics of fuzzy inference is still valid. If it is important to the user to have membership function values that are always greater than zero, Yam et al. [10] present a way of accomplishing this.

Table 3 summarizes the performance of the fuzzy filter for various sets of membership functions. As expected, the performance gets better as we go from no filtering to nominal filtering to optimal filtering. As expected, the perfor-

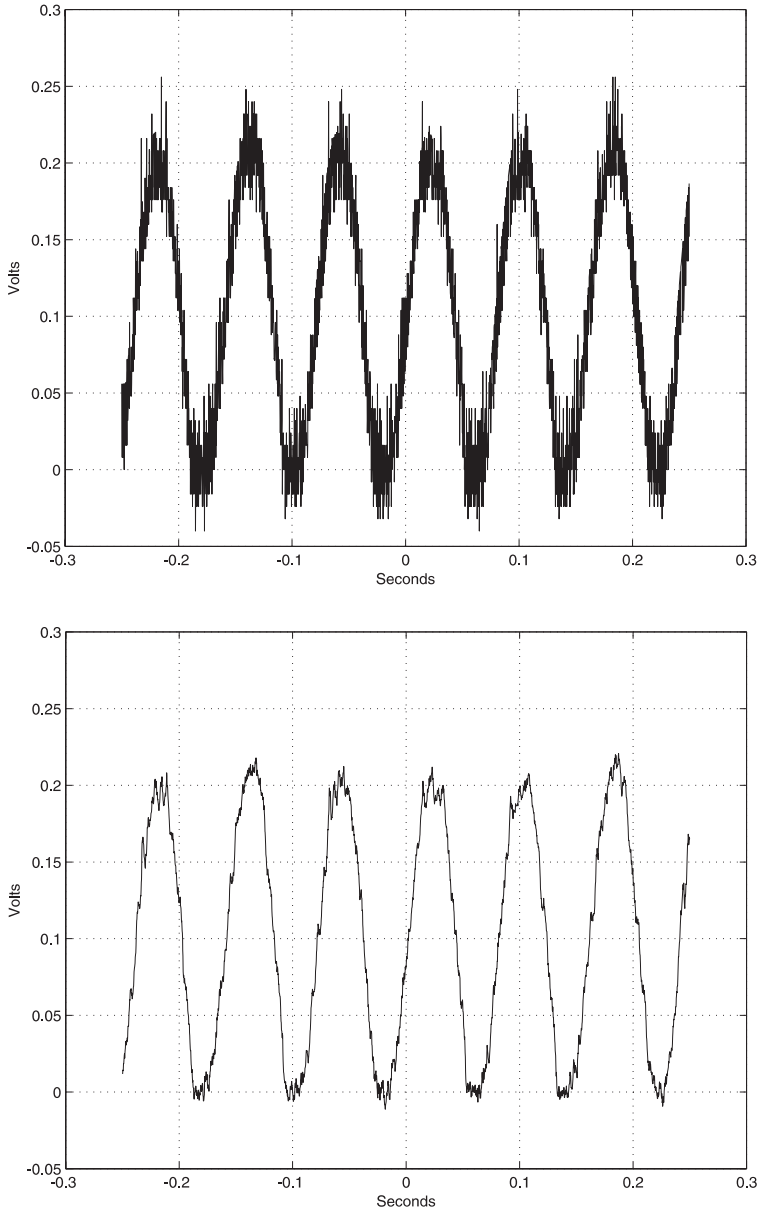

Fig. 7. Test data: (a) unfiltered; (b) filtered with fuzzy estimator.

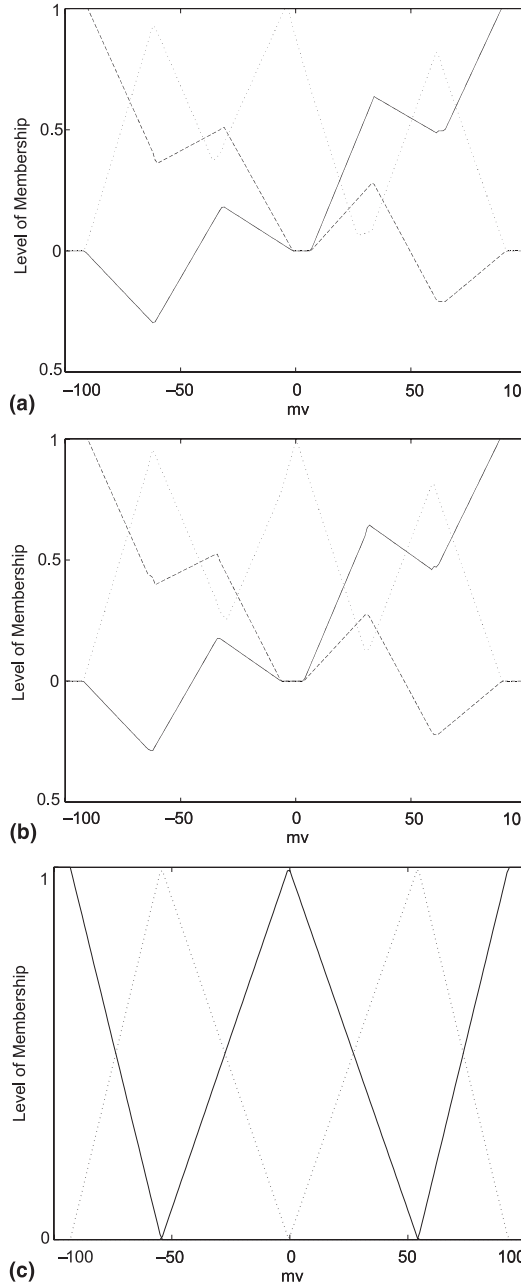

Fig. 8. Reduced membership functions for (a) input 1, (b) input 2, (c) output. For input 1 and input 2, dashed line = negative, dotted line = zero, and solid line = positive.

Table 2
Reduced rule base for fuzzy filter[a]

Input 1	Input 2		
	N	Z	P
N	NL	NS	Z
Z	NS	Z	PS
P	Z	PS	PL

[a] NL = negative large, n = negative, ns = negative small, Z = zero, PS = positive small, P = positive, PL = positive large.

Table 3
Error function values for test data

	Error function value
Raw data	18.0
Nominal membership functions	10.7
Optimized membership functions	7.4
Reduced membership functions	7.7

mance of the reduced rule base is worse than the performance of the optimal filter, but the performance degradation is not as severe as we might expect from reducing a 49-rule rule base to a 9-rule rule base.

6. Conclusion

A fuzzy filter has been applied to the estimation of motor winding currents. The fuzzy estimator offers the possibility of training if a nominal current history is known a priori. The gradient descent optimization discussed in this paper is attractive because of its conceptual straightforwardness, but one of its primary disadvantages is its convergence to a local minimum. Considering the fact that in this paper we optimized with respect to 63 variables, it would be very surprising if we were anywhere close to a global minimum. Further work on the topic of this paper is focusing on optimization methods that do better at finding the global minimum (e.g., genetic algorithms), integration of the filtering scheme with motor control, and real time implementation issues.

The SVD-based rule base reduction was shown to be effective at decreasing the number of rules used in the fuzzy filter. This reduction could be important for real time implementation where cycle time is at a premium. It is not difficult to program a general purpose rule base reduction algorithm if we can make the following assumptions: (1) There are an odd number of membership functions for the two inputs and the output; (2) the membership functions are symmetric

triangles; and (3) we desire to keep the two largest singular values in the R matrix of Eq. (56). A MATLAB *m*-file for rule base reduction (based on the algorithms presented in [10] and summarized here) of a general two-input, one-output fuzzy logic system can be downloaded from `http://csaxp.csuohio.edu/~simon/reduce/`.

Acknowledgements

The author wishes to thank Dennis Feucht of Innovatia Laboratories for his valuable assistance in this work.

References

[1] P. Krause, O. Wasynczuk, Electromechanical Motion Devices, McGraw-Hill, New York, 1989.
[2] D. Feucht, Sensorless start-up positioning of brushless DC motors, PCIM Magazine 19 (3) (1993) 24–27.
[3] D. Simon, Fuzzy estimation of DC motor winding currents, in: North American Fuzzy Information Processing Society Conference, New York, NY, 1999, pp. 859–863.
[4] D. Simon, H. El-Sherief, Hybrid Kalman/Minimax filtering in phase-locked loops, Control Engrg. Practice 4 (5) (1996) 615–623.
[5] D. Simon, H. El-Sherief, Fuzzy logic for digital phase-locked loop filter design, IEEE Trans. Fuzzy Systems 3 (2) (1995) 211–218.
[6] B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[7] K. Atkinson, An Introduction to Numerical Analysis, Wiley, New York, 1989.
[8] F. Guély, P. Siarry, Gradient descent method for optimizing various fuzzy rule bases, in: IEEE International Conference on Fuzzy Systems, San Francisco, CA, 1993, pp. 1241–1246.
[9] L. Wang, R. Langari, J. Yen, Principal components, B-Splines, and fuzzy system reduction, in: W. Chiang, J. Lee (Eds.), Fuzzy Logic for the Applications to Complex Systems, World Scientific, Singapore, 1996, pp. 255–259.
[10] Y. Yam, P. Baranyi, C. Yang, Reduction of fuzzy rule base via singular value decomposition, IEEE Trans. Fuzzy Systems 7 (2) (1999) 120–132.
[11] Y. Shi, R. Eberhart, Y. Chen, Implementation of evolutionary fuzzy systems, IEEE Trans. Fuzzy Systems 7 (2) (1999) 109–119.
[12] Y. Yam, Fuzzy approximation via grid point sampling and singular value decomposition, IEEE Trans. Syst., Man, Cybernet. – Part B: Cybernetics 27 (6) (1997) 933–951.