

Application of the Extended Kalman filter to fuzzy modeling: Algorithms and practical implementation

A. Javier Barragán Piña¹ José M. Andújar Márquez¹ Mariano J. Aznar Torres²
Agustín Jiménez Avello³ Basil M. Al-Hadithi³

¹DIESIA - Universidad de Huelva, Spain, {antonio.barragan, andujar}@diesia.uhu.es

²MICSEII - Universidad de Huelva, Spain, marianojose.aznar@alu.uhu.es

³DISAM - Universidad Politécnica de Madrid, Spain, {agustin.jimenez, basil.alhadithi}@upm.es

Abstract

Modeling phase is fundamental both in the analysis process of a dynamic system and the design of a control system. If this phase is in-line is even more critical and the only information of the system comes from input/output data. Some adaptation algorithms for fuzzy system based on extended Kalman filter are presented in this paper, which allows obtaining accurate models without renounce the computational efficiency that characterizes the Kalman filter, and allows its implementation in-line with the process.

Keywords: Algorithm, Kalman filter, estimation, fuzzy system, modeling.

1. Introduction

The use of Kalman filter in fuzzy logic has been researched in several applications, such as the extraction of rules from a given rule base [1], parameters optimization of defuzzification mechanisms [2] or in optimization of Takagi-Sugeno models ([3, 4]. However, this proposal has not been generalized yet to adapt antecedents and consequents of Takagi-Sugeno general models.

Some general algorithms for estimation of adaptive parameters of a fuzzy model using the extended Kalman filter (EKF) are presented in this paper. These algorithms are general because they do not limit the size of input/output vectors, neither the type nor distribution of the membership functions used in the definition of fuzzy sets of the model. Algorithms developed in this paper are based on the theoretical development presented in [5], however, in order to make the paper self-contained, the equations necessary for the proposed algorithms are repeated in this work. Authors try to use the excellent features of Kalman filter to obtain fuzzy models of unknown systems from input/output data, and also to allow its application in real-time [6, 5].

This paper is organized as follows: in section 2 is presented the fuzzy modeling problem in a completely general form with the notation that will be used along the article. This section is also devoted

to formal presentation of the extended Kalman filter and its use for modeling fuzzy systems. From here, in section 3 we propose three algorithms for modeling fuzzy systems using the extended Kalman filter. In section 4, we study the performance of the proposed algorithms to build fuzzy models in several examples, and finally, in section 5, some conclusions and future works are presented.

2. Problem Formulation

The extended Kalman filter [7] allows to model a nonlinear systems in presence of white noise both in model and in measures if the system supports linearized models around any working point. Let $\mathbf{p}(k)$ be the set of adaptive parameters of a fuzzy system, and $\mathbf{y}(k)$ the set of outputs of this fuzzy system, the system represented in (1) allows to obtain these parameters using the extended Kalman filter.

$$\begin{aligned}\mathbf{p}(k+1) &= \mathbf{p}(k) \\ \mathbf{y}(k) &= \mathbf{h}(\mathbf{x}(k), \mathbf{p}(k)) + \mathbf{e}(k)\end{aligned}\quad (1)$$

The Extended Kalman filter can be solved by iterative application of following set of equations [8]:

$$\mathbf{P}(k|k) = \Phi(k)\mathbf{P}(k|k-1)\Phi^T(k) + \mathbf{R}_v \quad (2)$$

$$\begin{aligned}\mathbf{K}(k) &= (\Phi(k)\mathbf{P}(k|k)\mathbf{C}^T(k) + \mathbf{R}_{ve}) \\ &\quad (\mathbf{C}(k)\mathbf{P}(k|k)\mathbf{C}^T(k) + \mathbf{R}_e)^{-1}\end{aligned}\quad (3)$$

$$\begin{aligned}\hat{\mathbf{p}}(k+1|k) &= \Phi(k)\hat{\mathbf{p}}(k|k-1) + \Gamma(k)\mathbf{u}(k) \\ &\quad + \mathbf{K}(k)(\mathbf{y}(k) - \mathbf{C}(k)\hat{\mathbf{p}}(k|k-1))\end{aligned}\quad (4)$$

$$\begin{aligned}\mathbf{P}(k+1|k) &= \Phi(k)\mathbf{P}(k|k)\Phi^T(k) + \mathbf{R}_v \\ &\quad - \mathbf{K}(k)(\mathbf{C}(k)\mathbf{P}(k|k)\Phi^T(k) + \mathbf{R}_{ve}^T),\end{aligned}\quad (5)$$

where $\hat{\mathbf{p}}(\cdot)$ is the estimation of the parameters vector, and $\mathbf{R}_v$, $\mathbf{R}_{ve}$ and $\mathbf{R}_e$ are the noise covariance matrices, estimated from the hope operator.

If $\mathbf{h}(\mathbf{x}(k), \mathbf{p}(k))$ is a completely general Takagi-Sugeno discrete fuzzy model, it can be represented by the following set of rules [9, 10, 11, 12]:

$$\begin{aligned}R^{(l,i)}: &\text{ If } x_1(k) \text{ is } A_{1i}^l \text{ and } \dots \text{ and } x_n(k) \text{ is } A_{ni}^l \\ &\text{ Then } y_i^l(k) = a_{0i}^l + \sum_{j=1}^n a_{ji}^l x_j(k),\end{aligned}$$

where n is the number of input variables, m is the number of output variables, $l = 1..M$ is the index of the rule, and M_i the number of rules that model the evolution of the i -th system output, $y_i(k)$.

The extended Kalman filter can be used to adjust the parameters of a fuzzy model, where the jacobian matrices of the system are:

$$\Phi(\mathbf{p}(k)) = \mathbf{I}, \quad (6)$$

$$\Gamma(\mathbf{p}(k)) = \mathbf{0}, \quad (7)$$

and

$$\mathbf{C}(\mathbf{p}(k)) = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{p}} \right|_{\mathbf{p}=\hat{\mathbf{p}}(k)} \quad (8)$$

Matrix $\mathbf{C}$ can be obtained by [5]:

$$\frac{\partial h_i}{\partial a_{jI}^L} = \begin{cases} \frac{w_I^L \tilde{x}_J}{\sum_{l=1}^{M_I} w_I^L} & \text{if } i = I \\ 0 & \text{if } i \neq I, \end{cases} \quad (9)$$

and

$$\frac{\partial h_i}{\partial \sigma_{jI}^L} = \begin{cases} \frac{\partial w_I^L}{\partial \sigma_{jI}^L} \sum_{j=0}^n \left(\frac{\sum_{l=1}^{M_I} w_I^L (a_{jI}^L - a_{jI}^l)}{\left(\sum_{l=1}^{M_I} w_I^L \right)^2} \right) \tilde{x}_j & \text{if } i = I \\ 0 & \text{if } i \neq I \end{cases} \quad (10)$$

where $\tilde{\mathbf{x}}$ is extended input vector [13, 14],

$$\tilde{\mathbf{x}} = (\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_n)^T = (1, x_1, \dots, x_n)^T,$$

$w_i^l(\mathbf{x})$ is the matching degree of the rules:

$$w_i^l(\mathbf{x}) = \prod_{j=1}^n \mu_{ji}^l(x_j(k), \sigma_{ji}^l),$$

and its derivative with respect to each of the parameters of the antecedents is given by:

$$\frac{\partial w_I^L}{\partial \sigma_{jI}^L} = \frac{\partial \mu_{jI}^L(x_J(k), \sigma_{jI}^L)}{\partial \sigma_{jI}^L} \prod_{q=1, q \neq J}^n \mu_{qI}^L(x_q(k), \sigma_{qI}^L). \quad (11)$$

Given the formulation exposed above, the estimation problem is to determine the values of adaptive parameters of both antecedents, σ_{ji}^l , and consequents of the rules, a_{ji}^l .

3. Application of the Extended Kalman Filter to fuzzy modeling

In this section we present three algorithms that allow the application of the extended Kalman filter to model a fuzzy system. Algorithm 1 [4, 15], EKF(c), adjust only the consequents of a fuzzy model, keeping the antecedents in its initial value. This algorithm is the fastest because needs to do fewer calculations, but it doesn't adjust antecedents and this

limits their ability to model complex systems. This algorithm is presented in Figure 1. We have considered the set of the initial parameters to 0, but it could be initialized by other procedures. α is a positive integer of a high value which represents the little certainty that the algorithm should give to the initial parameters, and $\mathbf{I}$ is the identity matrix.

```

1:  $\hat{\mathbf{p}}(0|-1) = \mathbf{0}$ 
2:  $\mathbf{P}(0|-1) = \mathbf{I}\alpha$ 
3: for  $k = 0..k_{end}$  do
4:   Calculate  $\mathbf{P}(k|k)$  by (2)
5:   Estimate  $\hat{\mathbf{y}}(k)$  using the fuzzy model
6:   Calculate  $\mathbf{C}(k)$  by (9)
7:   Get  $\mathbf{K}(k)$  by (3)
8:   Update  $\hat{\mathbf{p}}(k+1|k)$  by (4)
9:   Update  $\mathbf{P}(k+1|k)$  by (5)
10: end for k
```

Figure 1: EKF(c) algorithm. Adaptation of consequents of a fuzzy model.

Algorithms 2 and 3 adjust both antecedents and consequents of the fuzzy model, but they confront with modeling process using two different strategies. Algorithm 2, EKF(ac), represented in Figure 2, performs the simultaneous adjustment of antecedents and consequents. As in the previous algorithm, α and β are positive integers of a high value which represents the little certainty that the algorithm should give to the initial parameters. In this case, α is used for consequents and β for antecedents. The distinction of these two parameters can refine the algorithm and decrease response, both the modeling error, as the potential for instability of it.

```

1:  $\mathbf{p}_a$  = initial antecedents parameters
2:  $\mathbf{p}_c = \mathbf{0}$ 
3:  $\hat{\mathbf{p}}(0|-1) = [\mathbf{p}_a, \mathbf{p}_c]$ 
4:  $\mathbf{P}(0|-1) = \begin{pmatrix} \mathbf{I}\beta & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\alpha \end{pmatrix}$ 
5: for  $k = 0..k_{end}$  do
6:   Calculate  $\mathbf{P}(k|k)$  by (2)
7:   Estimate  $\hat{\mathbf{y}}(k)$  using the fuzzy model
8:   Calculate  $\partial \mathbf{w} / \partial \sigma$  by (11)
9:   Calculate  $\mathbf{C}_a(k)$  by (10)
10:  Calculate  $\mathbf{C}_c(k)$  by (9)
11:   $\mathbf{C}(k) = [\mathbf{C}_a(k), \mathbf{C}_c(k)]$ 
12:  Get  $\mathbf{K}(k)$  by (3)
13:  Update  $\hat{\mathbf{p}}(k+1|k)$  by (4)
14:  Update  $\mathbf{P}(k+1|k)$  by (5)
15: end for k
```

Figure 2: EKF(ac) algorithm. Simultaneous adaptation of antecedents and consequents of a fuzzy model.

Algorithm 3, EKF(c+a), is really the sequential execution of two filters. First it perform the setting of consequents and later the setting of antecedents (we check that the adjustment in the reverse order is worse, so it will not be considered). This algorithm is shown in Figure 3, where α and β have the same meaning as in the previous algorithm.

```

1:  $\mathbf{p}_c = \mathbf{0}$ 
2:  $\mathbf{p}_a$  = initial antecedents parameters
3:  $\mathbf{P}_c(0|-1) = \mathbf{I}\alpha$ 
4:  $\mathbf{P}_a(0|-1) = \mathbf{I}\beta$ 
5: for  $k = 0..k_{end}$  do
6:   Calculate  $\mathbf{P}_c(k|k)$  by (2)
7:   Estimate  $\hat{\mathbf{y}}(k)$  using the fuzzy model
8:   Calculate  $\mathbf{C}_c(k)$  by (9)
9:   Get  $\mathbf{K}_c(k)$  by (3)
10:  Update  $\hat{\mathbf{p}}_c(k+1|k)$  by (4)
11:  Update  $\mathbf{P}_c(k+1|k)$  by (5)
12:  .....
13:  Calculate  $\mathbf{P}_a(k|k)$  by (2)
14:  Estimate  $\hat{\mathbf{y}}(k)$  using the fuzzy model
15:  Calculate  $\partial \mathbf{w} / \partial \boldsymbol{\sigma}$  by (11)
16:  Calculate  $\mathbf{C}_a(k)$  by (10)
17:  Get  $\mathbf{K}_a(k)$  by (3)
18:  Update  $\hat{\mathbf{p}}_a(k+1|k)$  by (4)
19:  Update  $\mathbf{P}_a(k+1|k)$  by (5)
20: end for  $k$ 

```

Figure 3: EKF(c+a) algorithm. Separate adaptation of antecedents and consequents of a fuzzy model.

4. Examples

To demonstrate the practical application of the extended Kalman filter to fuzzy modeling, several examples will be shown. For each case we run the three algorithms presented in the previous section to evaluate their performance: EKF(c), EKF(ac) and EKF(c+a). These examples represent a wide range of possibilities: static and dynamic systems nonlinear, with only one type of membership function and with mixed functions. Example 3 makes a comparison with a recently published method [17], in order to assess the goodness of the proposed algorithms.

4.1. Example 1

Let be the system:

$$y(x) = e^{-0.03x} \sin(0.07x) \text{ with } x \in [-150, 150],$$

affected by a white noise of covariance $R_e = 0.5$. We are going to model this system with two different configurations. In the first case, we will use an initial model consisting only of Gaussian membership functions uniformly distributed. In the second case, we will start with a more complex initial model, which consists of several membership functions of different types, mixed. In order to verify the performance of algorithms, we will use the same noise in all cases. The initial covariance matrix for each of the algorithms will be initialized with the value that obtain best performance in each case, i.e., $\alpha = 10^2$ for EKF(c) algorithm, $\alpha = 10^3$ and $\beta = 10^{-2}$ for EKF(ac) algorithm, and $\alpha = 10^{10}$ and $\beta = 10^4$ for EKF(c+a).

4.1.1. Case I - Gaussian membership functions

Starting from an initial model whose antecedents are Gaussian membership functions uniformly distributed, and with all consequents set to zero, we have executed the 3 adjustment algorithms. EKF(ac) algorithm has modified antecedents slightly, but antecedents resulting from EKF(c+a) algorithm are shown in Figure 4.

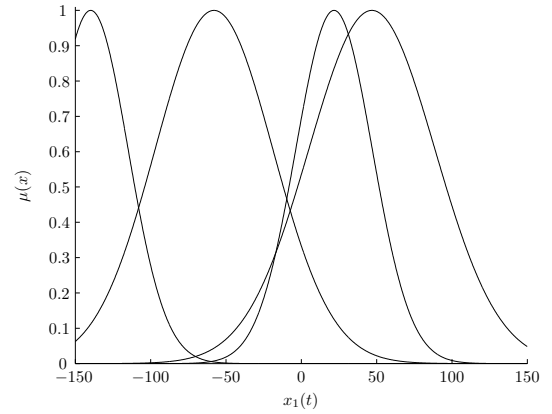


Figure 4: Membership functions resulting from the EKF(c+a) algorithm.

As shown in Figures 5 and 6, algorithm EKF(c) obtains a good model, but their in-line response is not very good, while algorithm EKF(ac) obtains a better performance during in-line modeling, but the final model obtained is worse than the rest. The best result both for in-line modeling and for the final model validation, is EKF(c+a) algorithm, which gets a much smaller error in both cases.

The mean final error of EKF(c) algorithm is 4.50899, it raises to 6.41349 for EKF(ac) algorithm and drops to 2.21189 for EKF(c+a) algorithm.

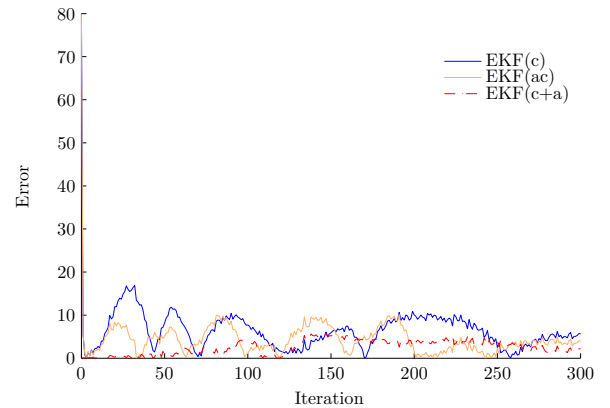


Figure 5: Modeling errors during execution.

The fuzzy model obtained by EKF(c+a) is:

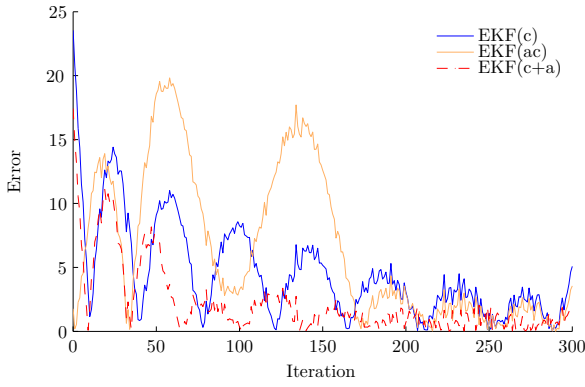


Figure 6: Errors of final models.

R(1,1): IF x is GAUSSMF(-139.773; 35.180)
 THEN $y = -535.075 - 3.977x$
 R(2,1): IF x is GAUSSMF(-57.928; 55.290)
 THEN $y = -36.671 - 0.746x$
 R(3,1): IF x is GAUSSMF(46.857; 59.213)
 THEN $y = 9.172 - 0.072x$
 R(4,1): IF x is GAUSSMF(-21.750; 35.923)
 THEN $y = 13.973 - 0.492x$

The response, both in-line models as the final models, can be seen in Figures 7 and 8 respectively.

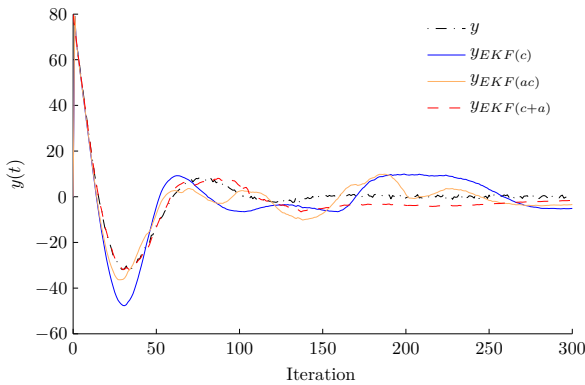


Figure 7: Run-time response of the models.

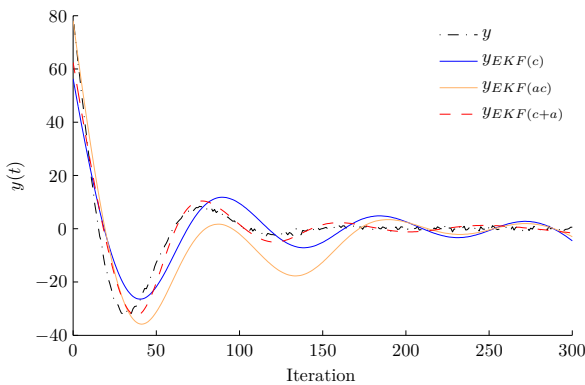


Figure 8: Response from the final models.

4.1.2. Case II - Membership functions of different type, mixed

In this case the antecedents of the initial model consist of a S function, a Z function, a trapezoidal function and a bell function, distributed as shown in Figure 9, and all the consequents initialized to zero.

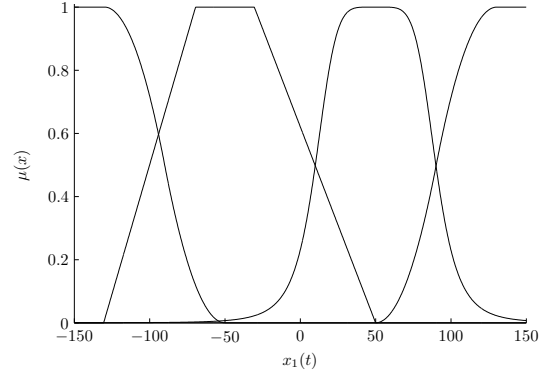


Figure 9: Initial membership functions.

As shown graphically in Figures from 10 to 13, the best results for both in-line and final model, again correspond to algorithm EKF(c+a).

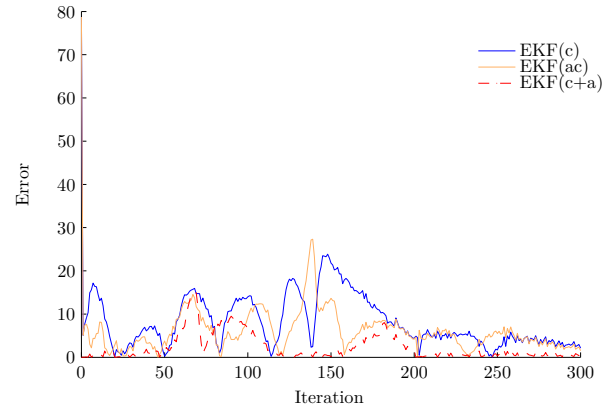


Figure 10: Modeling run-time errors.

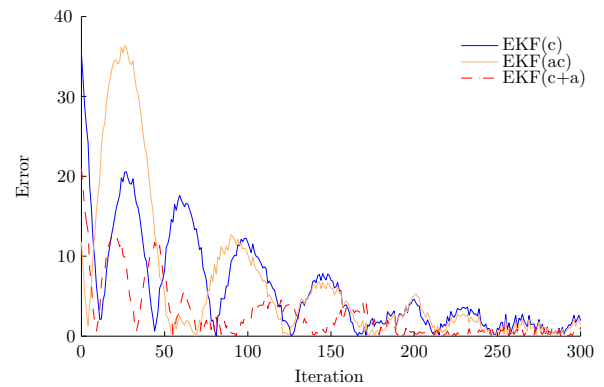


Figure 11: Errors of final models.

The mean final error of EKF(c) algorithm is 5.9141, worse compared to the previous case.

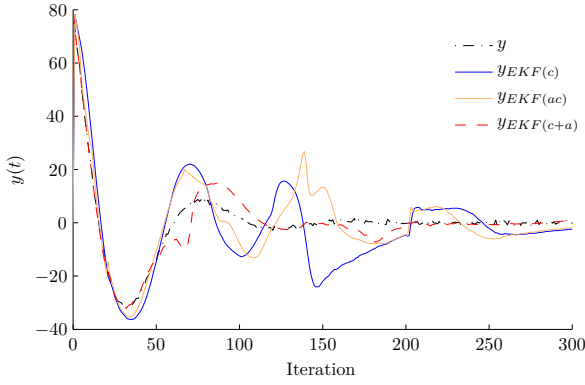


Figure 12: Run-time response of the models.

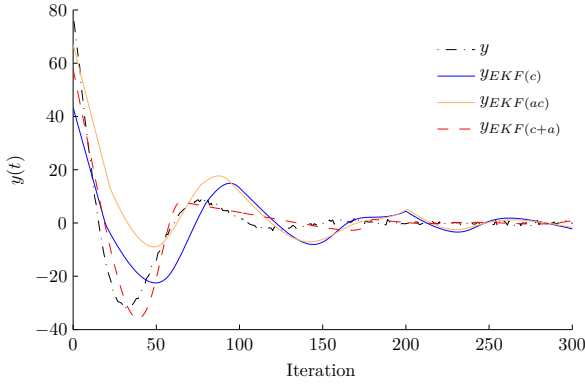


Figure 13: Response from the final models.

EKF(ac) algorithm obtain a slightly worse model in this case, with an error of 6.4323, while the mean final error of the EKF(c+a) algorithm remains the best, with an mean error of 2.67202.

Antecedents resulting from EKF(c+a) algorithm are shown in Figure 14, and the final fuzzy model obtained by this algorithm is:

- R(1,1): IF x is ZMF(-147.53; -85.03)
THEN $y = -404.844 - 3.084x$
- R(2,1): IF x is TRAPMF(-135.01; -12.63; -1.53; 29.62)
THEN $y = -1.733 - 0.114x$
- R(3,1): IF x is GBELLMF(50.01; 2.72; 96.02)
THEN $y = 3.190 - 0.060x$
- R(4,1): IF x is SMF(50.70; 104.76)
THEN $y = 2.665 + 0.005x$

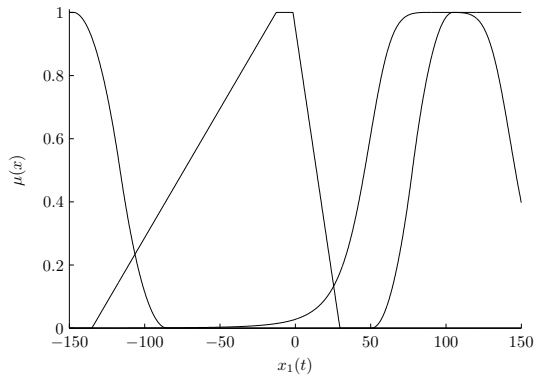


Figure 14: Membership functions resulting from the EKF(c+a) algorithm.

4.2. Example 2

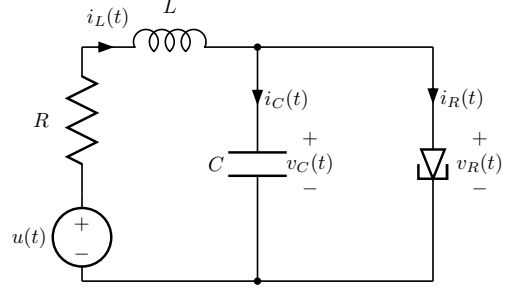


Figure 15: Tunnel diode circuit.

Given the tunnel-diode circuit shown in the Figure 15 [16], where $R = 1.5K\Omega$, $C = 2pF$ and $L = 5\mu H$, and whose dynamics can be represented by:

$$\begin{aligned}\dot{x}_1(t) &= \frac{x_2(t) - h(x_1(t))}{C} \\ \dot{x}_2(t) &= \frac{u(t) - x_1(t) - Rx_2(t)}{L},\end{aligned}$$

where $x_1(t) = v_C(t)$, $x_2(t) = i_L(t)$, $h(v)$ is the v_R - i_R characteristic of the tunnel-diode, and $u(t)$ is shown in the Figure 16. We assume that the system is affected by a white noise of covariance:

$$\mathbf{R}_e = \begin{pmatrix} 0.001 & 0 \\ 0 & 0.01 \end{pmatrix}.$$

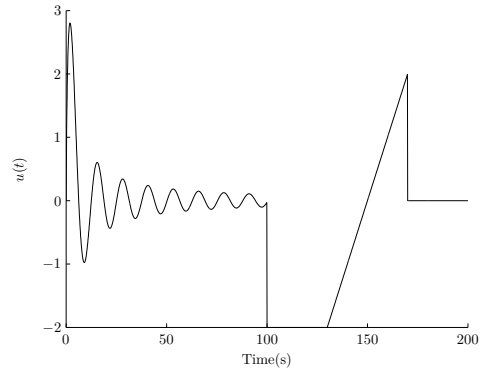


Figure 16: Input voltage applied to the circuit $u(t)$.

The initial covariance matrix is set again in order to get the best performance, i.e., $\alpha = 10^{10}$ for EKF(c) algorithm, $\alpha = 10^2$ and $\beta = 10^{-2}$ for EKF(ac) algorithm, and $\alpha = 10^{10}$ and $\beta = 10^4$ for EKF(c+a).

After making 100 runs for each of the algorithms the results are shown in Table 1. The graphical representation of mean errors in each of the runs can be seen in Figures 17 and 18.

	EKF(c)	EKF(ac)	EKF(c+a)
Mean Time (ms)	0.67781	0.68085	1.20500
In-line Error of $x_1 \times 10^{-3}$	0.83199	0.83599	0.78549
In-line Error of $x_2 \times 10^{-3}$	7.84798	7.83592	7.65570
Final Error of $x_1 \times 10^{-3}$	0.80531	0.80612	0.80523
Final Error of $x_2 \times 10^{-3}$	7.73933	7.73932	7.73933

Table 1: Mean values for 100 runs.

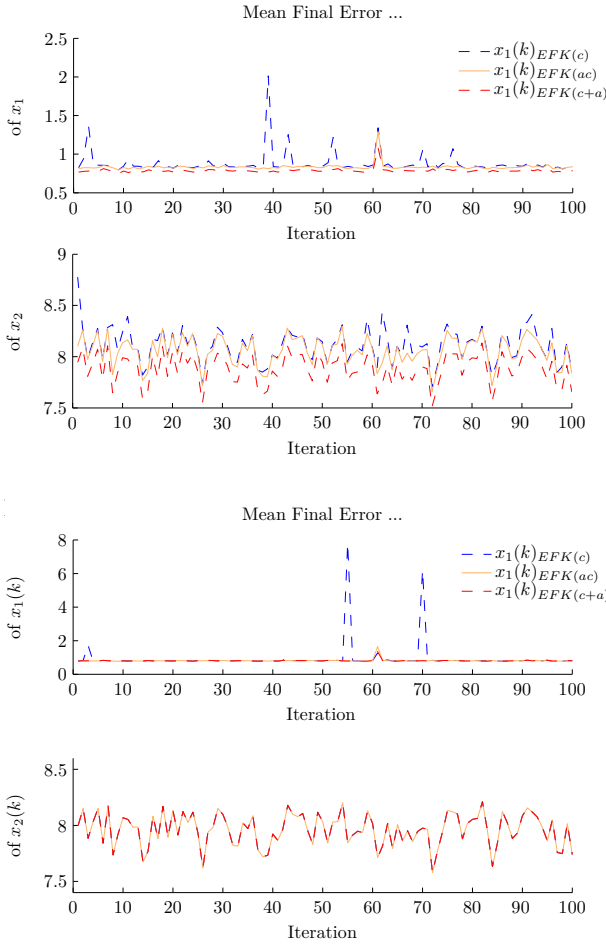


Figure 18: Errors of final models.

4.3. Example 3

We have taken a practical example from [17] in order to compare these methods with a recent one that is known has a good performance.

Be the nonlinear system:

$$f(x, y) = 5x^2 + 3xy - y^2, \text{ with } x, y \in [-4, 4],$$

affected by a white noise of covariance $R_e = 3$.

The initial model consists of Gaussian membership functions uniformly distributed, and the initial covariance matrix is initialized with the value giving the best performance in this case, i.e., $\alpha = 10^4$ for EKF(c) algorithm, $\alpha = 10^2$ and $\beta = 10^{-2}$ for EKF(ac) algorithm, and $\alpha = 10^5$ and $\beta = 10^{-2}$ for EKF(c+a). After run 100 times for each of the algorithms, the modeling errors are shown in Figures 19 and 20, the mean of this values are shown in the Table 2, and the time taken for each run is shown in Figure 21.

	EKF(c)	EKF(ac)	EKF(c+a)
Mean Time (ms)	0.17369	0.72062	0.52163
In-line Error	15.70668	8.04251	0.16824
Final Error	1.630987	2.03821	1.62997

Table 2: Mean values for 100 runs.

The modeling error of this system obtained in [17] is 0.6086 ± 0.0437 . If we compare this result with

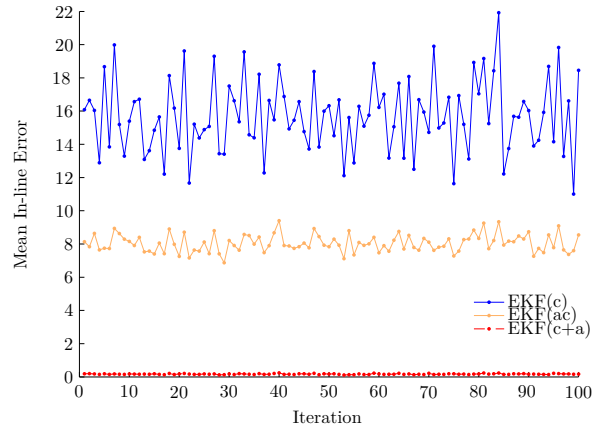


Figure 19: Run-time modeling mean errors.

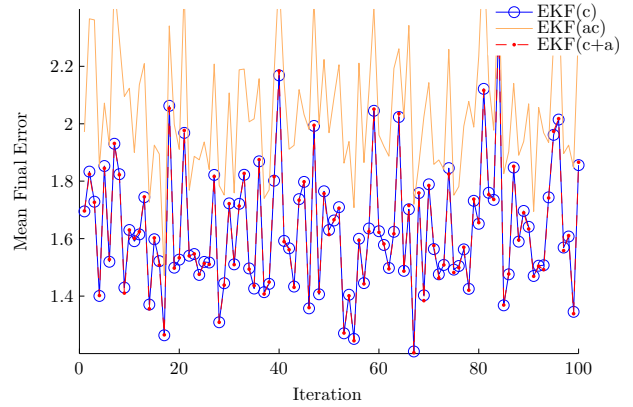


Figure 20: Modeling mean errors of final models.

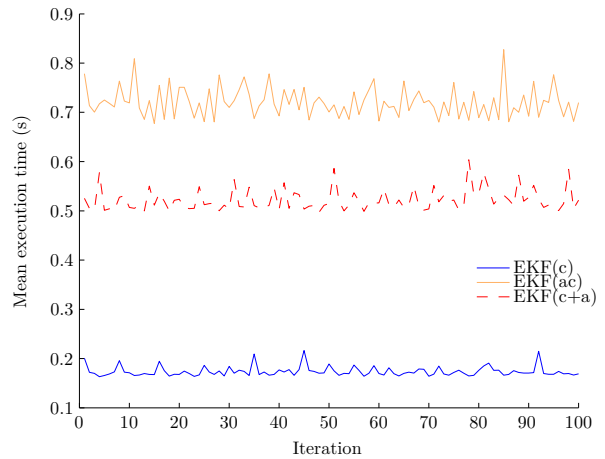


Figure 21: Mean running time.

the final mean error obtained in EKF(c+a), 1.62997 (see Table 2), we can see the first is slightly better, but note that EKF(c+a) is an in-line algorithm, and its performance in run-time is really good.

With the idea of showing the response of the different models obtained has been extracted one of the simulations, obtaining the results shown in Figures 22 and 23.

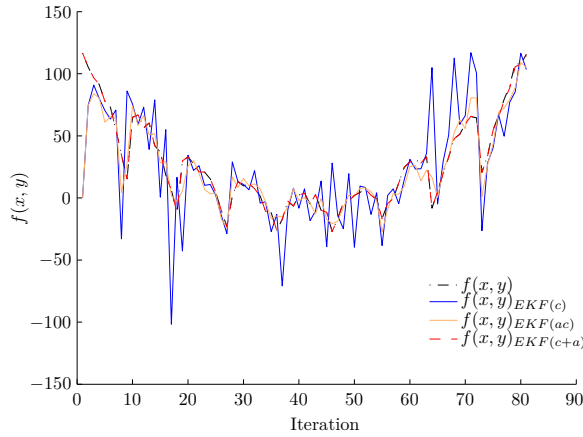


Figure 22: Run-time response of the models.

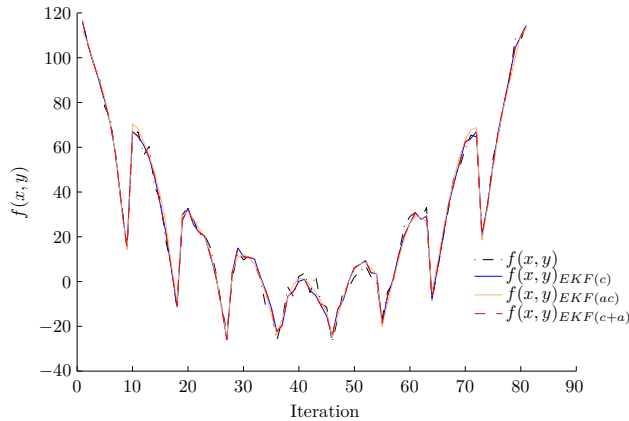


Figure 23: Response from the final models.

5. Conclusions

Two new algorithms based on extended Kalman filter for parametric adaptation of a fuzzy system completely general, i.e., without restrictions on the size of the input or output vectors, or the type or distribution of membership functions used in the definition of fuzzy sets of the model has been presented in this paper.

In order to show the generality of the methodology, several possible alternatives for fit both antecedents and consequents of the fuzzy model have been presented, and we have obtained several comparatives of accuracy and efficiency performed on three examples of nonlinear systems. In view of the results, it seems evident that the fitting algorithm in

two stages obtains better results both in the evolution of the instantaneous error while running in-line filter, as in the final models obtained. In addition, with respect to runtime, the EKF(c+a) algorithm has proven to be faster in most cases.

Tests have shown that these algorithms occasionally may diverge, especially the EKF(ac) algorithm, preventing a good in-line behavior. It has also been observed in all cases a great dependence on the values assigned to parameters α and β , which recommends a previous study of systems to model.

In future works we aim to improve the overall response of the algorithms through an in-line study of the evolution of the modeling process, so as to obtain a more robust performance of these filters. In addition, with respect to the initialization of the covariance matrices, authors are working on several alternatives to limit the current dependence of the parameters α and β .

Acknowledgment

The present work is a contribution of the DPI2010-17123 project, funded by the Spanish Ministry for Education and Science and by the European Regional Development Union.

References

- [1] Wang Liang and Yen John. Extracting fuzzy rules for system modeling using a hybrid of genetic algorithms and Kalman filter. *Fuzzy Sets and Systems*, 101(3):353–362, February 1999.
- [2] Tao Jiang and Yao Tang Li. Generalized defuzzification strategies and their parameter learning procedures. *IEEE Transactions on Fuzzy Systems*, 4(1):64–71, August 1996.
- [3] Pramath Ramaswamy, Martin Riese, Robert M. Edwards, and Kwang Youl Lee. Two approaches for automating the tuning process of fuzzy logic controllers. In *32nd IEEE Conference on Decision and Control. Part 2 (of 4)*, San Antonio, TX, USA, December 1993.
- [4] Dan J. Simon. Training fuzzy systems with the extended Kalman filter. *Fuzzy Sets and Systems*, 132(2):189–199, 2002.
- [5] Antonio Javier Barragán Piña, José Manuel Andújar Márquez, Mariano Aznar Torres, and Agustín Jiménez Avello. Methodology for adapting the parameters of a fuzzy system using the extended Kalman filter. In *7th conference of the European Society for Fuzzy Logic and Technology, EUSFLAT (LFA-2011)*, Aix-les-Bains, France, July 2011.
- [6] Agustín Jiménez, Basil M. Al-Hadithi, and Fernando Matía. An optimal T-S model for the estimation and identification of nonlinear functions. *WSEAS Trans. Sys. Ctrl.*, 3(10):897–906, January 2008.

- [7] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. Academic Press, 1979.
- [8] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. John Wiley & Sons, Inc., 2nd edition, 2001.
- [9] Robert Babuška. Fuzzy modeling – a control engineering perspective. In *Proceedings of FUZZ-IEEE/IFES'95*, volume 4, pages 1897–1902, Yokohama, Japan, March 1995.
- [10] Robert Babuška, Magne Setnes, Uzay Kaymak, and Hans R. van Nauta Lemke. Rule base simplification with similarity measures. In *Proceedings of the 5th IEEE International Conference on Fuzzy Systems*, volume 3, pages 1642–1647, New Orleans, LA, September 1996.
- [11] Hung T. Nguyen, Michio Sugeno, Richard M. Tong, and Ronald R. Yager. *Theoretical aspects of fuzzy control*. John Wiley Sons, 1995.
- [12] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
- [13] José Manuel Andújar and Antonio Javier Barragán. A methodology to design stable nonlinear fuzzy control systems. *Fuzzy Sets and Systems*, 154(2):157–181, September 2005.
- [14] José Manuel Andújar, Antonio Javier Barragán, and Manuel Emilio Gegúndez. A general and formal methodology for designing stable nonlinear fuzzy control systems. *IEEE Transactions on Fuzzy Systems*, 17(5):1081–1091, October 2009.
- [15] Fernando Matía, Agustín Jiménez, Basil M. Al-Hadithi, Diego Rodríguez-Losada, and Ramón Galán. The fuzzy Kalman filter: State estimation using possibilistic techniques. *Fuzzy Sets and Systems*, 157(16):2145–2170, August 2006.
- [16] H. K. Khalil. *Nonlinear systems*. Prentice-Hall, NJ, 2000.
- [17] Seok Beom Roh, Tae Chon Ahn, and Witold Pedrycz. The refinement of models with the aid of the fuzzy k-nearest neighbors approach. *IEEE Transactions on Instrumentation and Measurement*, 59(3):604–615, March 2010.