

Received December 6, 2018, accepted December 12, 2018, date of publication December 19, 2018, date of current version January 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2887355

Visualization and Interpretation Tool for Expert Systems Based on Fuzzy Cognitive Maps

ÁLVARO GARZÓN CASADO¹, PABLO CANO MARCHAL¹,
JUAN GÓMEZ ORTEGA, (Member, IEEE),
AND JAVIER GÁMEZ GARCÍA

Robotics, Automation and Computer Vision Group, Agrifood Campus of International Excellence (ceiA3), University of J  n, 23071 J  n, Spain

Corresponding author:   lvaro Garz  n Casado (garzon@ujaen.es)

The work of P. C. Marchal was supported in part by the F.P.U. Grant from the Spanish Ministry of Education. This work was supported in part by the Project DPI2011-27284, Project PI10-AGR-6616, and Project DPI2016-78290-R.

ABSTRACT This paper presents a python library that includes a toolkit with the aim of improving the interpretability of expert systems based on fuzzy cognitive maps through improvements in the visualization and representation of the graphs that can be drawn using the variables of the resulting models. The motivation for the development of the library arises from the need to improve the interpretability of the aforementioned expert systems, given that their multilayer and extracted from experts' knowledge nature can make them very difficult to interpret even for the expert user. Throughout the paper, the reader will be introduced to the basic features of fuzzy logic and fuzzy cognitive maps, and the different developed tools will be defined and exemplified.

INDEX TERMS Expert systems, fuzzy cognitive maps, fuzzy logic, interpretability, visual.

I. INTRODUCTION

In this paper we present *visualFCMlib*, an open-source, python-based toolkit that allows to improve the interpretability of Fuzzy Cognitive Maps (FCM) via a series of tools for representation and simplification of the models.

Fuzzy logic (FL) enables the use of an inference system that simulates the procedures of human reasoning in systems based on knowledge. Fuzzy inference systems (FIS) are probably the most famous and relevant applications of FL and Fuzzy Set theory [1]. The theory of FL provides a mathematical framework that allows modeling the degree of uncertainty inherently associated with human reasoning processes so that it can be interpreted by a computer.

Since the 70's of last century, most of the effort of the researchers working on FL and FIS, as Mamdani [2] or Sugeno [3], [4], has been to improve the computation process of the inference phase. These efforts have crystallized in an enormous advance in the field, with myriads of successful application being reported in the literature.

FL and fuzzy logic systems (FLS) have been extensively used in many different fields of science in general, and industry specifically, with many different purposes, see, for instance, [5]–[11].

Interpretability is one of the most important and interesting features within FIS. However, interpretability is, at the same time, one of the most difficult features to actually attain in a FIS of practical interest. In fact, an appropriate interpretability is not always reached when a ES is developed, more so if we are dealing with large systems where the rules are automatically obtained from data [12].

Many software packages that aid with the design and computations of FIS systems are currently available for practicing engineers, however the interpretability of the developed systems is usually not their main focus, so their utility for interpreting large systems involving many variables is usually limited.

Some of these are Matlab^{  } [13], Juzzy Online [14], [15], Xfuzzy [16] or KBCT [17], among others. These tools are going to be briefly explained in next section. Dealing particularly with FCM, only the developed tool in [18] may be remarked, where learning algorithms for computing the causal weights are included.

In the same way, there are other available softwares for the design of the FCM. On the one hand, as a pioneering software we have *FCM Modeler*, of which there are few references in the literature, but that already had the ability to store various systems based on FCM [19]. On the other hand,

as the latest software we have available *FCM Expert*, in which you can find tools for the simplification of the topology of the system, in addition to supervised and unsupervised learning algorithms [20]. Some software developed between the previous two can be found in [21]. Besides, other softwares have been used, such as [22], for the development of works in FCM field, as [23] and [24]. In comparison, our software, in spite of not including learning algorithms at the moment, presents the potentiality of being written in an open-source language, which leaves the way open to any developer who sees in it an opportunity for improvement.

Arguably, FCM inherently provides a better framework for interpretability as it is easier to grasp the relations between variables when they are organized in a hierarchical graph that relate smaller subsets of variables [25].

In addition, the use of matrices of relationship between each node of the system when using FCM allows a much more precise adjustment of the system relationships than that allowed by other types of techniques.

Some relevant works in the field of FCM are those that can be found in [26], [27], and [28]. FCM is also a good tool in combination with Big Data, being able to use this large-scale data source to serve as knowledge base for development of FIS. Examples of synergies among these two branches of knowledge are those that can be found in [29]–[33].

Therefore, the development and implementation of the system described in this paper is justified under the necessity of a visual, practical and useful system that allows facilitate the interpretation of the inference phase in complex FIS. Sometimes, specially in large systems where experimental data is scarce, the development of expert systems based on expert knowledge is a titanic task. A widespread strategy consists of grouping the variables of the system into small groups of subsystems, so that their debugging is easier [34]. However, the interconnection of these smaller subsystems may provide initially unsatisfactory results, requiring to update and fine tune some of the parameters to obtain a complete system that behaves consistently with the prescriptions of the experts. The developed utilities provide the designer with useful tools in order to tackle this debugging phase with greater comfort.

In this paper we present a visual that allows both to interpret a ES based on FCM and to extract useful information from it. A problem presented by expert systems based on expert knowledge lies in the user’s ability to retain enough information in memory when analyzing system results. According to some works, like those presented in [35], [36], and [37], people can only hold between 4 and 9 elements of information simultaneously in the form of short-term memory. Because of this, it is easy not to be able to retain all the necessary information to analyze, for example, the implications of the variation of an input node on all the output nodes in a system like those mentioned above. In order to avoid this problem, some simplification tools have been developed, which will be presented later in this paper.

The rest of the paper is organized as follows. Section II reviews some background relevant for the problem at hand,

commenting on the currently available visualization tools and briefly addressing FCM. Section III focuses on the proposed tools, offering a detailed description of the capabilities of the module, while Section IV shows a practical use of our visualization system using a case study based on the virgin olive oil production process (VOOPP). Finally, Section V concludes this paper and shows our conclusions and states the future work.

II. BACKGROUND

A. VISUALIZATION TOOLS

As commented in the Introduction, although there are many tools available that assist in the development of FIS, their assistance for interpreting the outcomes of the system and the underlying inference process is usually not their strongest point. Some of the most relevant examples of these tools are briefly presented next.

- **MatLab®** includes fuzzy logic toolboxes, so users can define their own type-1 (T1) FLS, including membership functions (MFs) and rules, and simulate the resultant ES (Fig. 1). Some examples about the use of the Fuzzy Logic Toolbox of MatLab can be found in [38] and [39]. Additionally, MatLab® allows users to design T1 FLSs using command-line functions.

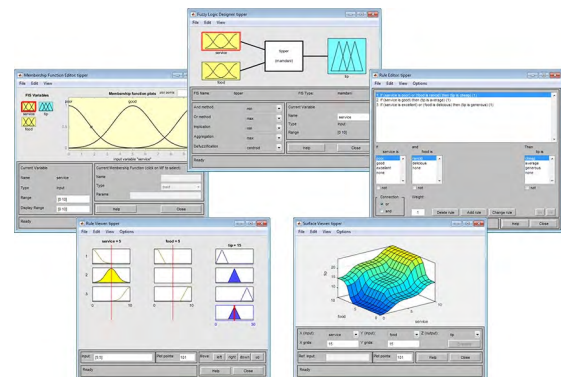


FIGURE 1. The Membership Function Editor (top left), FIS Editor (center), Rule Editor (top right), Rule Viewer (bottom left), and Surface Viewer (bottom right).

- **Juzzy Online** is an online Java based FLS toolkit that can be used to design T1, interval type-2 (IT2) and general type-2 (GT2) FLSs. Besides, input values can be set as non-singleton values. The toolkit provides the possibility of plotting FSs and some visual interpretability of the inference phase, via visualization of the inference steps via zSlices. (Fig. 2).
- **Xfuzzy**, and its following versions, provides CAD tools to develop GUIs to ease the design flow at the stages of the FLS, using its own language, XFL3. Nevertheless, it doesn’t provide interpretability tools or aids for it.
- **KBCT** is a multiplatform tool that eases the extraction and representation of knowledge within systems based in FL. Its main goal is the generation and refinement of

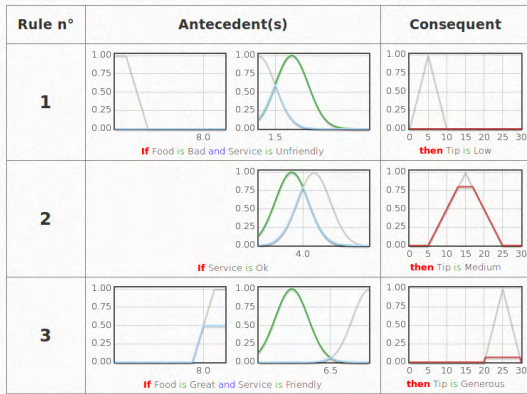


FIGURE 2. Visual representation of the results of the inference phase within the tipping problem using Juzzy Online.

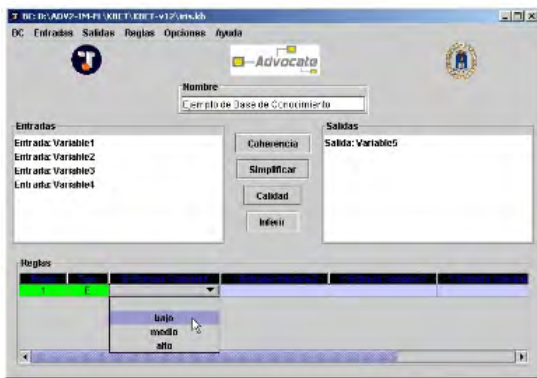


FIGURE 3. Rules definition in a system generated by KBCT.

T1 fuzzy knowledge bases (Fig. 3). As well as Xfuzzy, no interpretability tools are implemented.

Further information about tools mentioned above, and other not mentioned, can be found in [40].

B. FUZZY COGNITIVE MAPS

FCMs is a technique inspired in cognitive maps (CMs) developed by Axelrod in [41] and extended by Kosko [42]. FCMs are fuzzy-graph structures for representing causal reasoning. Their fuzziness allows hazy degrees of causality between hazy causal objects. Their graph structure allows to easily visualize the casual relations between the concepts of the model, and it allows knowledge bases to be grown by connecting different FCMs.

The modeling method supported by the library is a FCM extension based on Simplified Dynamic Cognitive Networks (sDCN) [43] and defined in greater detail in [44]. Formally, the system is defined as a tuple:

$$M = \langle V, A \rangle \tag{1}$$

where V expresses the set of nodes representing the concepts or the variables of the system and A expresses the set of established relations among them. The causal relations

established between nodes are modeled with matrices called causal relationship matrices.

Given the multilayer structure of a typical model, the nodes can be simultaneously predecessors and successors. The calculus process of the value of the nodes in set V for a successor, given the values of its predecessors nodes, is broken down into the three main steps that are described next.

In a first step, the impact received by node i is calculated, which is defined as follows:

$$w_i = \sum_{j=1}^{n_i} r_{ij} R_{ij} S_f(v_j) = [w_i^1 \ w_i^2 \ \dots \ w_i^{K_i}]^T \tag{2}$$

where w_i is an array denoting the impact received by node i from its predecessors. Here, n_i is the number of predecessor nodes of the node i , whereas r_{ij} is a scalar representing the intensity of the relation (element of set A) between nodes i and j . Finally, R_{ij} is the causal relationship matrix belonging to the relation between nodes i and j , whereas $S_f(v_j)$ is an array containing the degree of membership of for each fuzzy set of a given node v_j for the crisp value assigned to it.

As a second step, the value of the crisp output $S_c(v_i)$ is calculated. The calculation is made by means of a weighted average of the set peaks over the node:

$$S_c(v_i) = \frac{\sum_{k=1}^{K_i} w_i^k q_i^k}{\sum_{k=1}^{K_i} w_i^k} \tag{3}$$

where K_i denotes the total amount of peaks in a set of the system and q_i^k expresses the value of the peak k of the node i .

Finally, the grade of membership vector of of the node i ($S_f(v_i)$) can be calculated as:

$$S_f(v_i) = [\mu_{L_{v_i}^1}(S_c(v_i)), \mu_{L_{v_i}^2}(S_c(v_i)), \dots, \mu_{L_{v_i}^{K_i}}]^T \tag{4}$$

where $\mu_{L_{v_i}^{K_i}}$ represents the degree of membership of for each fuzzy set L_{v_i} of the crisp value of the node.

III. PROPOSED TOOL

In order to provide a useful and complete toolkit to effectively help in the interpretation of the system, two types of tools have been developed. On the one hand, the first type is called *Representation tools*. The main focus of these tools is the correct separation and representation of each node in the system when they are graphically represented. On the other hand, the second type of developed tools can be called *Simplification tools*, which are a series of tools designed to reduce the studied system to a new simpler one that fulfills certain characteristics. These tools are based on the reduction of nodes and system relationships, either to the most important or to a certain threshold of relevance within the model.

We must make clear that the purpose of these tools is to help with the improvement of the interpretation of the inference phase, being the simplification tools an addition to this purpose despite being able to be used as an end in themselves in specific occasions.

As the reader can see below, the simplification tools presented in this paper not try to simplify the model itself, but the representation of that model. While other papers propose the simplification of the model by deleting weak nodes or pruning weak relationships [45], we prefer not to simplify the model, because the nature of expert systems based on expert knowledge can cause a significant variation in the weight of a node over the whole.

The simplifications shown in [45] can be applied to the representation of the models, as we showed with the tool *Nodes extraction*, through a feature called Degree of Influence (DoI), explained later.

First, in Sections III-A and III-B we introduce the *Representation tools*. Then, from Section III-C to Section III-E we present the *Simplification tools*, together with some application examples.

A. NODE COLOR ASSIGNMENT

An adequate node color assignment (NCA) is the cornerstone of all the developed tools, as well as the basis of the correct differentiation between the different nodes. The fundamental premise to follow in the assignment will be that two predecessor nodes of the same successor node can't have the same color, since in this way there would be confusion in the representation when using the rest of the tools, as we can see in Figure 4, where the light orange color of predecessor nodes of *Successor node 3* (red color relations) will cause confusion in the representation. Colors are stored in color palettes that the user can easily define using the hexadecimal code of each color.

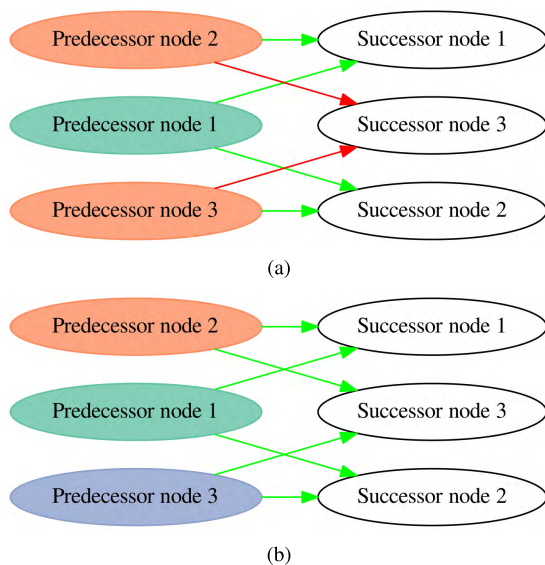


FIGURE 4. Different examples of node color assignment. (a) Example of incorrect assignment. (b) Example of correct assignment.

B. SUCCESSOR NODES BAR PLOTS

The optional use of bar plots in the successor nodes representation allows the user to obtain really important insight about the inference process of the system. On the one hand,

the user can quickly identify the predecessor nodes with high influence over their successors nodes just taking into account the size of each bar on the plot, that is, *how much* the successor node is being influenced by the corresponding predecessor nodes. On the other hand and possibly more important than the previous feature, the user may easily determine the sign of the current relation, it means, *how* the predecessor node is influencing on the successor node depending on its current value.

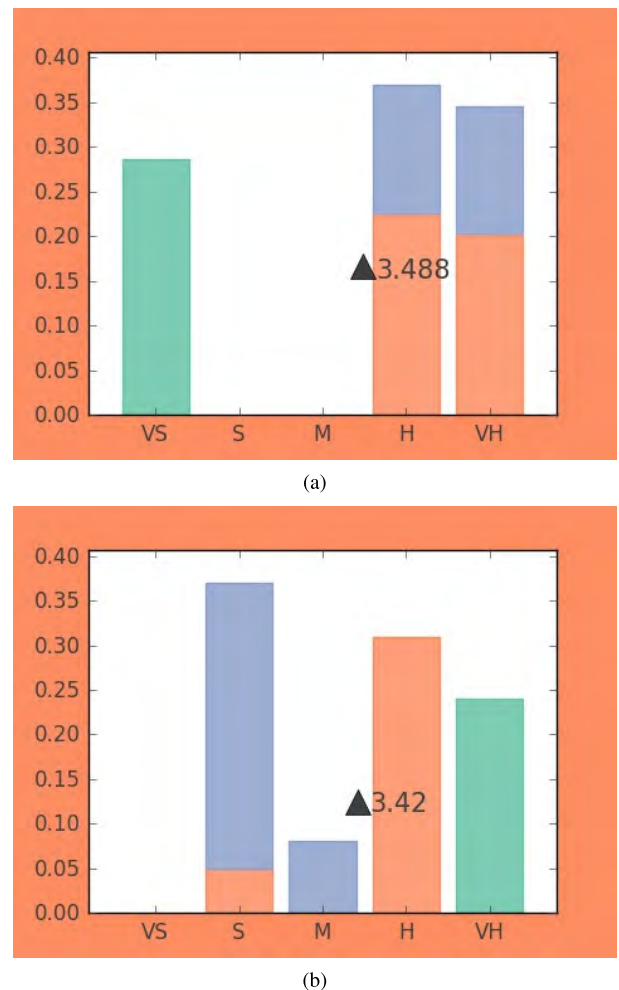


FIGURE 5. Different examples of a bar plot for a single node with two different set of input values for its predecessor nodes. (a) Example 1. (b) Example 2.

As can be seen in Figure 5, different input values provide different outputs and, therefore, different bar plots for each successor node. The differences between the examples highlight the relevance of the previous features. The first feature, weight of impact or *how much*, denotes that the node represented by the blue bar is the one that has the least impact on the successor node in Example 1, with an approximate value of 29 %, while this same node exerts the greatest weight on the successor node in Example 2, with 40 % of the weight. The second feature, sign of the relation or *how* the predecessor node influences, is shown through the position of the light

green bar in each example. In the first one, the predecessor node represented by green color is influencing the current node on the lower side of its value, while in the second example this node is influencing the current node on the higher side of its value.

Both examples show the crisp value of the current successor node, remaining almost the same in both examples for the set of selected input values.

C. NODES EXTRACTION

When a ES is composed of a number of nodes that exceeds about half a hundred, its graphic processing and comprehension begins to become difficult because of both graphic reasons (resolution of graphic representation) and cognitive reasons (the greater the number of nodes, the greater the number of relationships among them, which hinders their understanding).

The Node extraction tool allows to select either a single node or a subset of nodes, and only the nodes and arcs related to the selected ones are represented. There are three different modes for the extraction: *backward* mode, in which predecessor nodes are represented; *forward* mode, in which successor nodes are represented; and *complete* mode, a combination of the previous ones.

The systems also implements a feature that allows to extract nodes with a certain DoI. DoI is defined as the position occupied by a specific predecessor node based on the percentage weight that relate it with the extracted node of interest. For example, in an extraction of the node shown in Figure 5(b) from a two layers system with DoI equal to one, just the node itself and the node represented by blue color are represented; whereas if a DoI equal to two is used, the node represented by light orange color is represented too.

The main purpose of this tool is to serve as a debugging tool to check and improve the state of subsystems within the ES itself. Besides, this tool is the core of the rest tools described below.

D. TOTAL EXTRACTION

The node extraction tool provides the possibility of obtaining the simplified graph from a list of determined nodes. Nevertheless, sometimes it is more useful and convenient to obtain an individual and simultaneous simplification of all the nodes of the system.

This tool, which can be considered as an expansion of the previous one, allows the user to generate a certain amount of graphs, with the same types of extraction mode as the previous tool and allowing the distinction according to DoI, which provide a quick overview of the relationships of each individual node instead of recurrently resorting to the node extraction tool.

E. TOTAL WEIGHT BETWEEN NODES

Both tools, nodes extraction and total extraction, allows the simplification of a system, differentiating nodes according

to their weights over their neighbor nodes. However, in a multilayer system (with more than two layers), the representation of an input node and an output node in a multilayer graph doesn't mean that this initial node is the one that has more weight on that final node among all the initial nodes in the system because the relative weights can vary.

As an example, let us consider the three-layered graph 6(a), and let us further assume that we need to perform a backward extraction of node 9, so that the most important initial node related to it is found. This extraction is shown in Figure 6(b), showing node 2 as the initial node extracted.

However, observing the internodal weights it can be observed that the initial node that really exerts the highest influence on node 9 is node 3 (Figure 6(c) and Table 1).

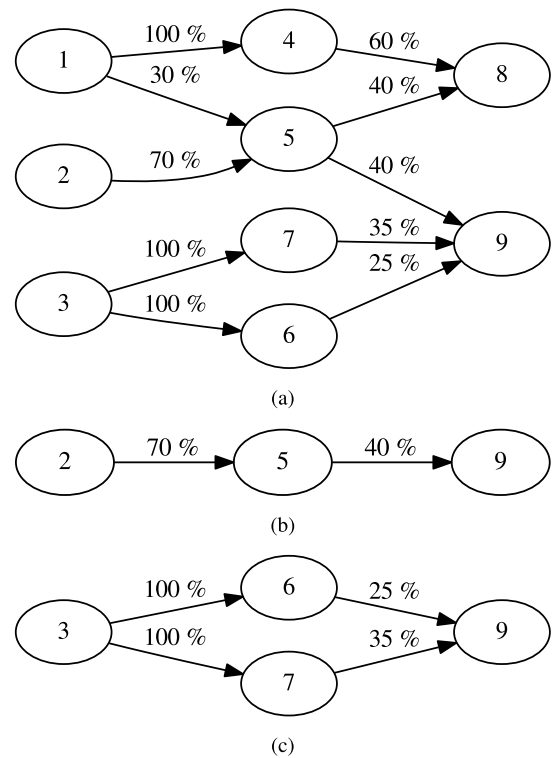


FIGURE 6. Explanatory graphs of the total weight between nodes tool. (a) Graph example. (b) Backward node 9 extraction, with DoI = 1. (c) Different paths linking nodes 3 and 9.

TABLE 1. Internodal weights for the different combinations Initial node - Final node of the graph in Figure 6(a).

		Output nodes	
		8	9
Input nodes	1	72 %	12 %
	2	28 %	28 %
	3	0 %	60 %

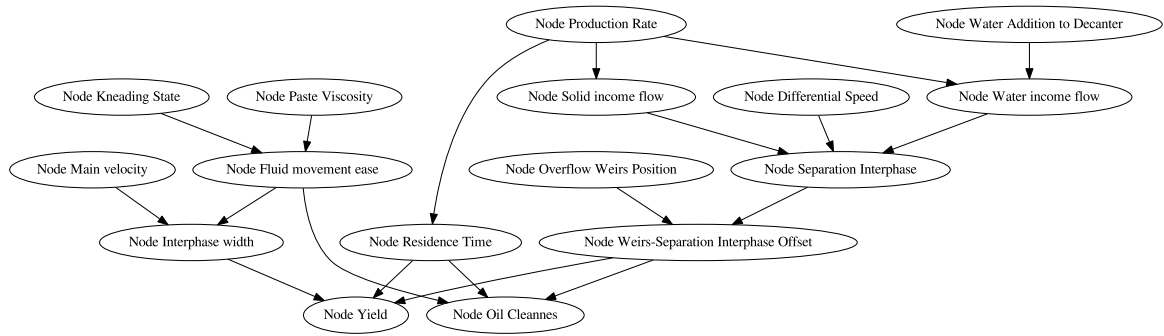


FIGURE 7. Model of solid-liquid interphase separation.

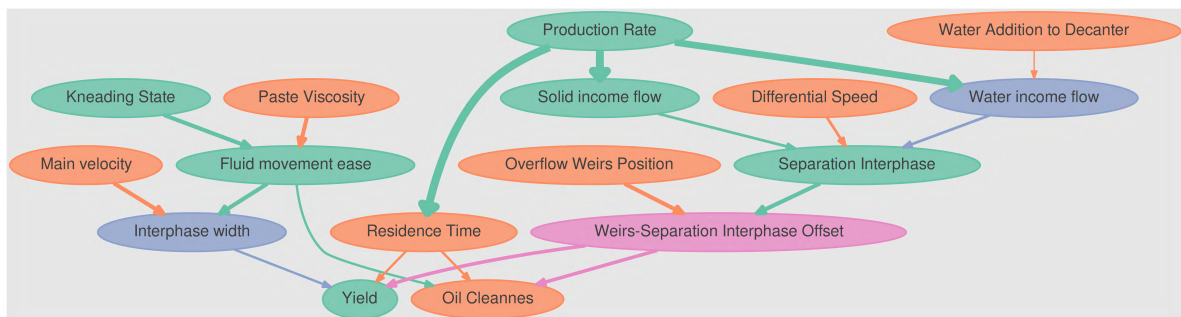


FIGURE 8. Model of solid-liquid interphase separation including automatic node color assignment. The width of the arcs is related with the weight of the relations.

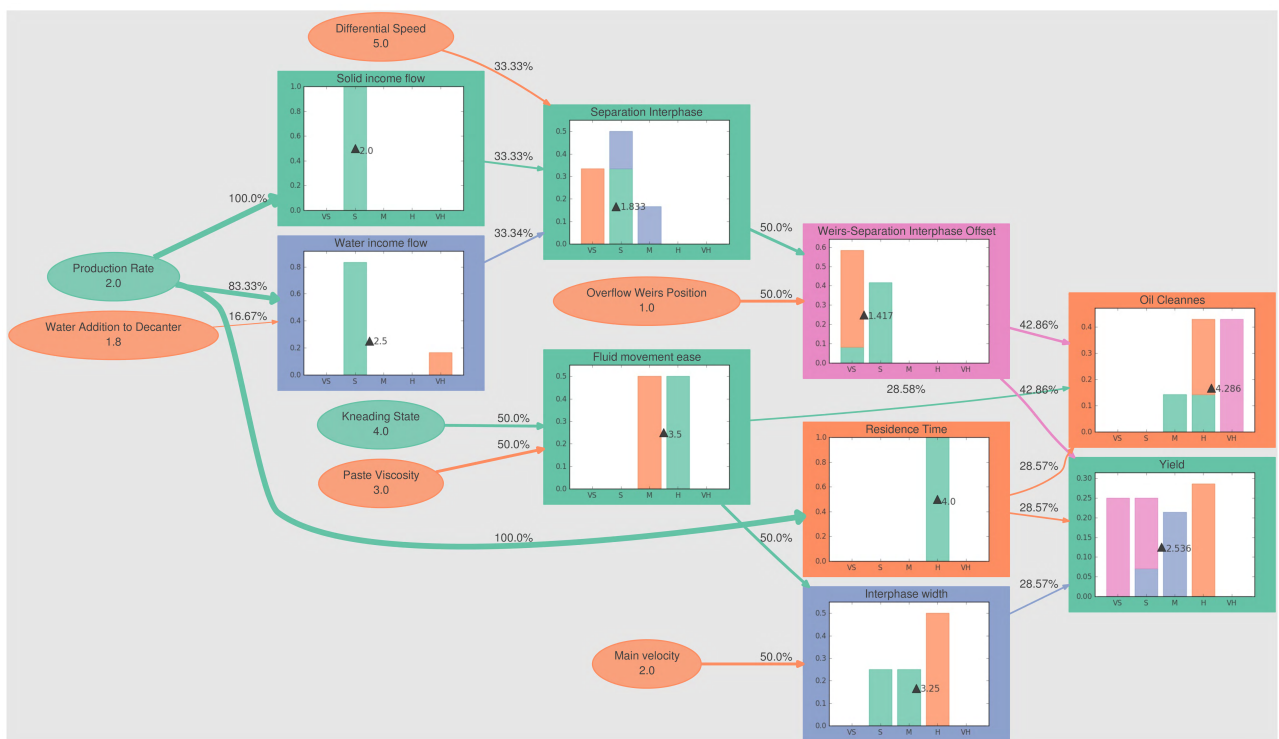


FIGURE 9. Model of solid-liquid interphase separation including automatic node color assignment, bar plots and information about nodes and their interaction.

Therefore, Figure 6(b) shows that node 2 is the node with the highest influence on node 5, and that, in turn, node 5 is the one with highest influence on node 9;

however, this doesn't mean that node 2 is the initial node with the highest influence on node 9, as has been shown.

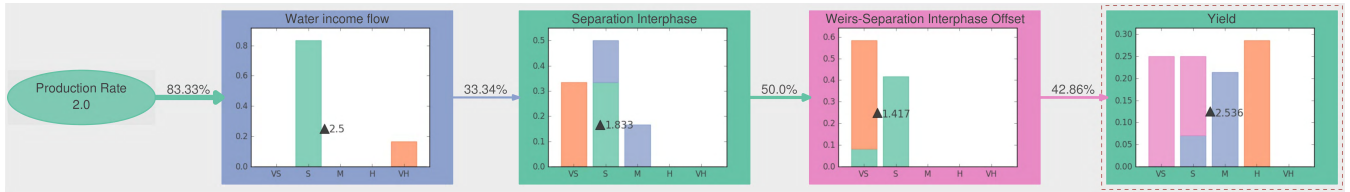


FIGURE 10. 1-degree of influence backward extraction of node Yield.

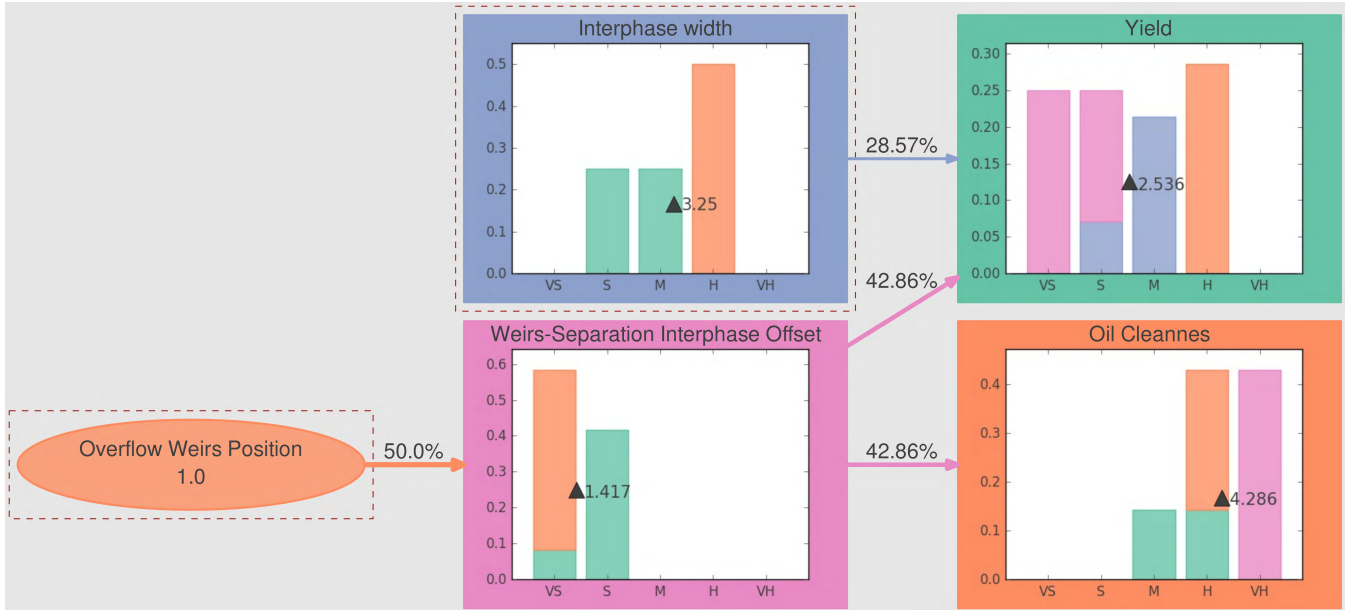


FIGURE 11. 2-degree of influence forward extraction of nodes Overflow Weirs Position and Interphase width.

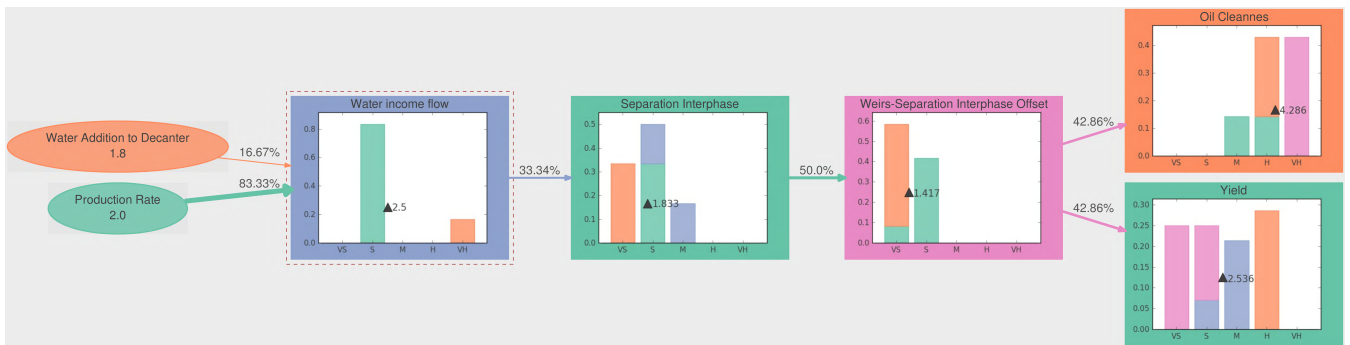


FIGURE 12. 2-degree of influence complete extraction of node Water income flow.

The actual internodal weight between node i and node j , \hat{W}_{ij} , can be computed according to the formula:

$$\hat{W}_{ij} = \sum_k \prod_m W_m^k \quad (5)$$

where:

- k represents the number of existing paths between node i and node j .
- m represents the number of existing relations between nodes inside each path k_1, k_2, \dots

- W_m^k denotes the internodal weight between two neighbor nodes inside the current path.

All of this can be checked in example included in Figure 6.

IV. CASE STUDY

The case study presented to illustrate the capabilities of the visualFCMlib library is the process of solid-liquid separation performed in the decanter during the Virgin Olive Oil Production Process (VOOPP). The solid-liquid separation

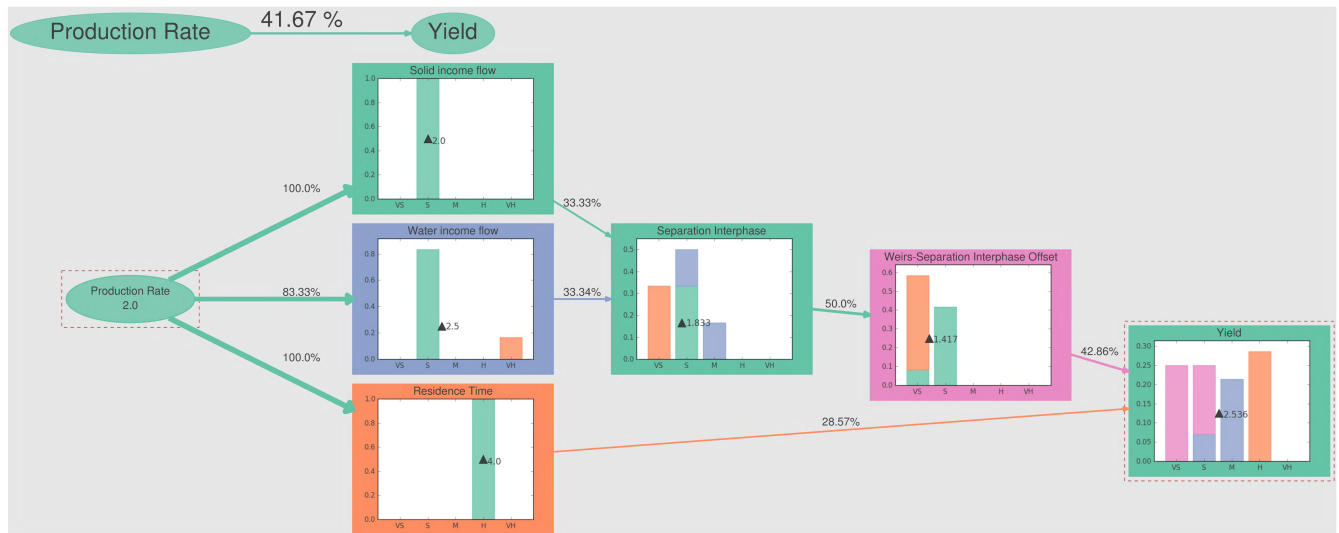


FIGURE 13. Resulting graph from obtaining total internodal weight of Production Rate node over Yield node.

process is particularly important for the oil recovery yield.

The model used to represent this process is based on the fact that the relative position between the oil-water interphase and the oil weirs is the key element that determines the efficiency of the decanter in its labor. The variable representing the offset between these elements is called weirs-separation line offset. Locations of the overflow weirs position, fixed by a plant operator, closer to the decanter rotation axis than to separation line, dependent on several operating parameters, cause a greater value of oil cleanness at the expense of the yield and vice versa.

If we think of the decanter as a simple settling tank [46] we can relate the outflow of pomace with the separation line and differential speed or axial velocity:

$$q_{out} = \alpha \Delta_{\omega} r_s, \tag{6}$$

where α is an unknown positive constant.

Assuming that the differential speed is constant, an increase in production rate causes an increase in separation line, since the total mass of the decanter content must remain constant, and therefore the outflow of pomace is increased. We can apply the same argument for the solid income flow and the water income flow.

Another key parameter in the decanter efficiency is the width of the oil-water interphase. This parameter decrease with the increase of main velocity, due to Stoke's law [47], and with the increase in fluid movement ease, a variable that takes into account paste kneading state and viscosity.

Finally, a longer residence time allows for better yield, since more time is granted for the particles to separate.

The variables mentioned above are summarized in Table 2.

Next, a series of examples are developed as a demonstration of the tools explained in section III. The complete

TABLE 2. Definition of the main variables related to each of the main variables of the model.

Main variable	Symbol	Variable
Weirs-Separation Line Offset	Δ_r	Weirs-Separation Line Offset
	r_1	Overflow Weirs Position
	r_s	Separation Interphase
	O_c	Oil Cleanness
	X	Yield
	Δ_{ω}	Differential Speed
Interphase width	F	Production Rate
	W_{wo}	Interphase width
	Ω	Main velocity
Residence Time	E	Fluid movement ease
	t_r	Residence Time

TABLE 3. Initial values given for examples to input variables.

Input variable	Value
Overflow Weirs Position	1
Differential Speed	5
Production Rate	2
Main velocity	2
Water Addition to Decanter	1.8
Paste Viscosity	3
Kneading State	4

solid-liquid separation model is depicted in Figs. 7 and 8, whereas the initial values used for each of the input nodes of the model are included in Table 3.

Figure 9 shows the representation of the system including the values assigned to the nodes, the impact exerted on

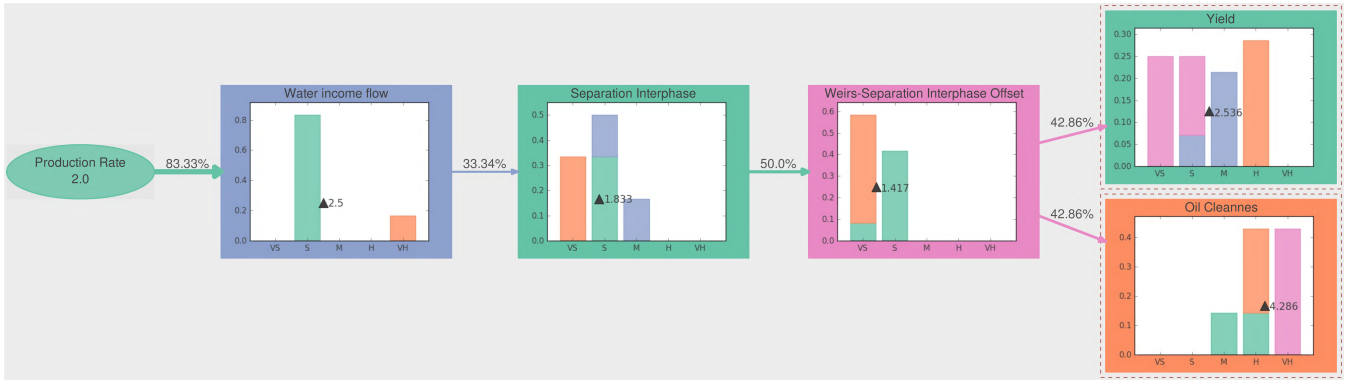


FIGURE 14. Backward simplification of the system with DoI = 1.

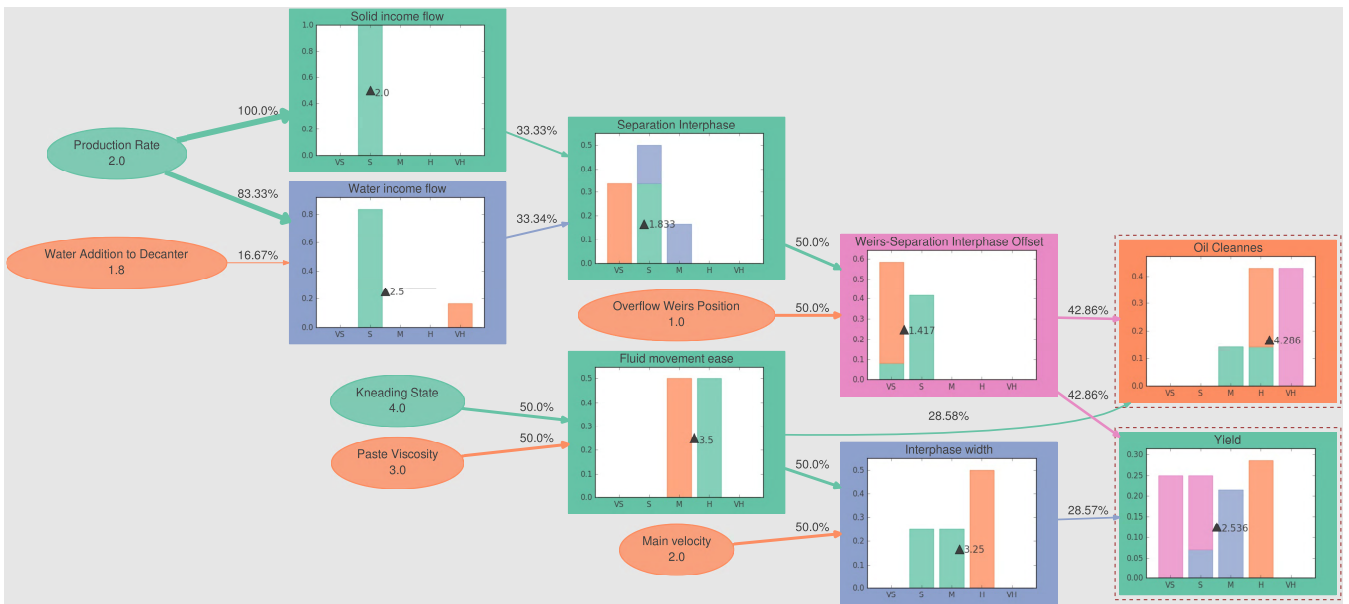


FIGURE 15. Backward simplification of the system with DoI = 2.

successor nodes and the bar plots generated for each successor node.

As already mentioned in section III-C, the basic node extraction tool allows the user to select just a single node or a subset of different nodes, representing only those nodes and arcs related to the selected ones. Figure 10 shows the backward extraction with DoI 1 of the output node Yield, whereas Figure 11 depicts the forward extraction with DoI 2 of the Overflow Weir Position and Interphase Width nodes. Finally, Figure 12 shows a complete extraction with DoI 2 of the Water income flow node.

In order to show the use of the tool for obtaining internodal weights, described in Section III-E, Figure 13 depicts the total weight exerted by the initial node Production Rate over the final node Yield. Additionally, Table 4 includes the total weight that each of the initial nodes exert on each of the two final nodes.

TABLE 4. Total weights exerted by the initial nodes on the two final nodes.

		Output nodes	
		Yield	Oil Cleanness
Input nodes	Production Rate	41.67 %	41.67 %
	Water Addition to Decanter	1.19 %	1.19 %
	Main velocity	14.29 %	0 %
	Overflow Weirs Position	21.43 %	21.43 %
	Differential Speed	7.14 %	7.14 %
	Kneading State	7.14 %	14.29 %
	Paste Viscosity	7.14 %	14.29 %

To conclude the exposition of the tools, two different simplifications of the model are shown, using different DoI to provide an example of the different levels of complexity achieved.

V. CONCLUSIONS

This paper has presented a new python-based (available online at <https://github.com/AlvaroGarzonCasado/visualFCMLib>) library called visualFCMLib that offers several tools to assist in the interpretation of the inference process of an expert system based on Fuzzy Cognitive Maps. Having appropriate interpretability of the system recommendations is useful both during the development and the deployment phases of the system. During the development stage, a better understanding of the current relations between the nodes aid in the fine tuning of the model parameters. In the deployment stage, these tools can be used to analyze the causes of abnormal recommendations, again simplifying the update of the system and offering better insight in the decision making process.

The different Sections have included a brief overview of the current tools of the developed library, among which we can highlight that thanks to the use of bar plots user can quickly discern the level of influence present between two variables, as well as the relative weight that each predecessor node has on a successor node, a measure that can also be verified by means of both thickness of arrows representing internodal arcs and tag of those arcs if they have been represented. Future work will include the design of a GUI to ease the use of the system for non-technical users.

REFERENCES

- [1] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S001999586590241X>
- [2] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proc. IEE*, vol. 121, no. 12, pp. 1585–1588, 1974.
- [3] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
- [4] A. Hamam and N. D. Georganas, "A comparison of Mamdani and Sugeno fuzzy inference systems for evaluating the quality of experience of Hapto-Audio-Visual applications," in *Proc. IEEE Int. Workshop Haptic Audio Vis. Environ. Games (HAVE)*, Oct. 2008, pp. 87–92.
- [5] P.-H. Kuo, T.-H. S. Li, Y.-F. Ho, and C.-J. Lin, "Development of an automatic emotional music accompaniment system by fuzzy logic and adaptive partition evolutionary genetic algorithm," *IEEE Access*, vol. 3, pp. 815–824, 2015.
- [6] L. Suganthi, S. Iniyan, and A. A. Samuel, "Applications of fuzzy logic in renewable energy systems—A review," *Renew. Sustain. Energy Rev.*, vol. 48, pp. 585–607, Aug. 2015.
- [7] I. Tal and G.-M. Muntean, "Towards reasoning vehicles: A survey of fuzzy logic-based solutions in vehicular networks," *ACM Comput. Surv.*, vol. 50, no. 6, 2017, Art. no. 80.
- [8] P. Melin and O. Castillo, "A review on type-2 fuzzy logic applications in clustering, classification and pattern recognition," *Appl. Soft Comput.*, vol. 21, pp. 568–577, Aug. 2014.
- [9] H. Wang, C. Sheng, and X. Lu, "Knowledge-based control and optimization of blast furnace gas system in steel industry," *IEEE Access*, vol. 5, pp. 25034–25045, 2017.
- [10] S. Anand, I. Sharma, and R. Goyal, "Extended understanding of dyadic friendship using fuzzy measures: A simulation approach," *IEEE Access*, vol. 5, pp. 21179–21192, 2017.
- [11] J. Liang and F. Zhu, "Soil moisture retrieval from UWB sensor data by leveraging fuzzy logic," *IEEE Access*, vol. 6, pp. 29846–29857, 2018.
- [12] S. Guillaume, "Designing fuzzy inference systems from data: An interpretability-oriented review," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 3, pp. 426–443, Jun. 2001.
- [13] *MATLAB Fuzzy Logic Toolbox*, MathWorks, Inc., Natick, MA, USA, Mar. 2010.
- [14] C. Wagner, "Juzzy—A Java based toolkit for type-2 fuzzy logic," in *Proc. IEEE Symp. Adv. Type-2 Fuzzy Logic Syst. (T2FUZZ)*, Apr. 2013, pp. 45–52.
- [15] C. Wagner, M. Pierfitt, and J. McCulloch, "Juzzy online: An online toolkit for the design, implementation, execution and sharing of type-1 and type-2 fuzzy logic systems," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2014, pp. 2321–2328.
- [16] D. R. López, C. J. Jiménez, I. Baturone, A. Barriga, and S. Sánchez-Solano, "Xfuzzy: A design environment for fuzzy systems," in *Proc. IEEE Int. Conf. Fuzzy Syst., IEEE World Congr. Comput. Intell.*, vol. 2, May 1998, pp. 1060–1065.
- [17] J. M. Alonso, L. Magdalena, and S. Guillaume, "KBCT: A knowledge extraction and representation tool for fuzzy logic based systems," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 2, Jul. 2004, pp. 989–994.
- [18] G. Nápoles, I. Grau, R. Bello, M. León, K. Vahooof, and E. Papageorgiou, "A computational tool for simulation and learning of fuzzy cognitive maps," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Aug. 2015, pp. 1–8.
- [19] S. Mohr, "Software design for a fuzzy cognitive map modeling tool," Rensselaer Polytech. Inst., Troy, NY, USA, Tech. Rep. 66.698, 1997.
- [20] G. Nápoles, M. Leon, I. Grau, and K. Vanhoof, "Fuzzy cognitive maps tool for scenario analysis and pattern classification," in *Proc. IEEE 29th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2017, pp. 644–651.
- [21] G. Felix, G. Nápoles, R. Falcon, W. Froelich, K. Vanhoof, and R. Bello, "A review on methods and software for fuzzy cognitive maps," in *Artificial Intelligence Review*. Amsterdam, The Netherlands: Springer, 2017, doi: [10.1007/s10462-017-9575-1](https://doi.org/10.1007/s10462-017-9575-1).
- [22] E. Papageorgiou. (2016). *FCM Wizard*. [Online]. Available: <http://www.fcmwizard.com>
- [23] G. Nápoles, E. Papageorgiou, R. Bello, and K. Vanhoof, "Learning and convergence of fuzzy cognitive maps used in pattern recognition," *Neural Process. Lett.*, vol. 45, no. 2, pp. 431–444, 2016.
- [24] C. De Maio, G. Fenza, V. Loia, and F. Orciuoli, "Linguistic fuzzy consensus model for collaborative development of fuzzy cognitive maps: A case study in software development risks," *Fuzzy Optim. Decis. Making*, vol. 16, no. 4, pp. 463–479, 2017.
- [25] V. Torra, "A review of the construction of hierarchical fuzzy systems," *Int. J. Intell. Syst.*, vol. 17, no. 5, pp. 531–543, 2002.
- [26] C. Eden, "Analyzing cognitive maps to help structure issues or problems," *Eur. J. Oper. Res.*, vol. 159, no. 3, pp. 673–686, 2004.
- [27] Z.-Q. Liu, "Fuzzy cognitive maps: Analysis and extensions," in *Soft Computing and Human-Centered Machines*. Tokyo, Japan: Springer, 2000, pp. 57–84.
- [28] A. Jastrzębska and A. Cislak, "Interpretation-aware cognitive map construction for time series modeling," *Fuzzy Sets Syst.*, to be published, doi: [10.1016/j.fss.2018.05.013](https://doi.org/10.1016/j.fss.2018.05.013).
- [29] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: State-of-the-art, needs and perspectives," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2389–2406, 3rd Quart., 2018.
- [30] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: Big data toward green applications," *IEEE Syst. J.*, vol. 10, no. 3, pp. 888–900, Sep. 2016.
- [31] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: Greening big data," *IEEE Syst. J.*, vol. 10, no. 3, pp. 873–887, Sep. 2016.
- [32] J. Wu, I. Bisio, C. Gniady, E. Hossain, M. Valla, and H. Li, "Context-aware networking and communications: Part 1 [guest editorial]," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 14–15, Jun. 2014.
- [33] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, and Y. Yang, "Big data meet cyber-physical systems: A panoramic survey," *IEEE Access*, vol. 6, pp. 73603–73636, 2018.
- [34] E. I. Papageorgiou and J. L. Salmeron, "A review of fuzzy cognitive maps research during the last decade," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 1, pp. 66–79, Feb. 2013.
- [35] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychol. Rev.*, vol. 63, no. 2, pp. 81–97, 1956.
- [36] S. J. Luck and E. K. Vogel, "The capacity of visual working memory for features and conjunctions," *Nature*, vol. 390, no. 6657, pp. 279–281, 1997.
- [37] R. Gallotti, M. A. Porter, and M. Barthelemy, "Lost in transportation: Information measures and cognitive limits in multilayer navigation," *Sci. Adv.*, vol. 2, no. 2, p. e1500445, 2016.

- [38] M. S. M. Aras, F. A. Ali, F. A. Azis, S. M. S. S. A. Hamid, and M. F. H. M. Basar, "Performances evaluation and comparison of two algorithms for Fuzzy Logic rice cooking system (MATLAB Fuzzy Logic Toolbox and FuzzyTECH)," in *Proc. IEEE Conf. Open Syst. (ICOS)*, Sep. 2011, pp. 400–405.
- [39] N. Hasim, M. S. M. Aras, M. Z. A. Rashid, A. M. Kassim, and S. S. Abdullah, "Development of fuzzy logic water bath temperature controller using MATLAB," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2012, pp. 11–16.
- [40] J. McCulloch, "Fuzzycrator: A python-based toolkit for automatically generating and analysing data-driven fuzzy sets," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2017, pp. 1–6.
- [41] R. Axelrod, Ed., *Structure of Decision: The Cognitive Maps of Political Elites*, 1st ed. Princeton, NJ, USA: Princeton Univ. Press, 1976.
- [42] B. Kosko, "Fuzzy cognitive maps," *Int. J. Man-Mach. Stud.*, vol. 24, no. 1, pp. 65–75, 1986. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020737386800402>
- [43] Y. Miao, C. Miao, X. Tao, Z. Shen, and Z. Liu, "Transformation of cognitive maps," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 114–124, Feb. 2010.
- [44] P. C. Marchal, J. G. García, and J. G. Ortega, "Application of fuzzy cognitive maps and run-to-run control to a decision support system for global set-point determination," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 2256–2267, Aug. 2017.
- [45] W. Homenda, A. Jastrzebska, and W. Pedrycz, "Time series modeling with fuzzy cognitive maps: Simplification strategies," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.* Berlin, Germany: Springer, 2014, pp. 409–420.
- [46] P. C. Marchal, "Contribution to the modeling and automatic control of the virgin olive oil elaboration process," Ph.D. dissertation, Univ. Jaén, Jaén, Spain, 2015.
- [47] L. Civantos, *Obtención del Aceite de Oliva Virgen*. Madrid, Spain: Editorial Agrícola Española, 1998.



JUAN GÓMEZ ORTEGA (M'10) received the degree in electrical engineering and the Ph.D. degree from the University of Seville, Seville, Spain, in 1989 and 1994, respectively. He became a Professor with the University of Jaén, Jaén, Spain, in 2001, where he is currently the Rector.

From 1987 to 2001, he was with the Departamento de Ingeniería de Sistemas y Automática, University of Seville, as a Research Assistant, and then as an Assistant Professor and Associate Professor. He has been responsible for several research projects on robotic systems, computer vision applied to fault detection, and automatic assembly, with some of them being directly applied to industry. His current research interests include force control and sensor fusion in robotic manipulators, sensor planning, mobile robot, and education in engineering. He has been serving as a Reviewer for different technical journals.



ÁLVARO GARZÓN CASADO received the graduate degree in mechanical engineering and the M.Eng. degree from the University of Jaén, Jaén, Spain, in 2015 and 2017, respectively.

He currently combines his studies in MBA with the University of Jaén, conducting research in the field of fuzzy logic.



PABLO CANO MARCHAL received the degree in electrical engineering from the University of Seville, Seville, Spain, in 2007, and the Ph.D. degree from the University of Jaén, Jaén, Spain, in 2015.

He was a Visiting Researcher with the Department of Automatic Control, University of Lund, Lund, Sweden, in 2013. Since 2015, he has been with the System Engineering and Automation Department, University of Jaén, as an Assistant Professor. His current research interests include automatic control applied to olive oil production process and education in engineering. He was a recipient of the Formación de Profesorado Universitario Grant from the Spanish Ministry of Education and Science, in 2009.



JAVIER GÁMEZ GARCÍA received the degree in electrical engineering and the Ph.D. degree from the University of Jaén, Jaén, Spain, in 2001 and 2006, respectively.

He was a Visiting Researcher with the Department of Automatic Control, University of Lund, Lund, Sweden, from 2003 to 2004. Since 2005, he has been with the System Engineering and Automation Department, University of Jaén, as an Assistant Professor. He is currently the Director of the Transfer and Entrepreneurship Secretariat with the University of Jaén, and the Research Results Transfer Office. His research interests include force control and sensor fusion in robotic manipulators, engineering education, and automatic control applied to olive oil and production process. He was a recipient of the Formación de Profesorado Universitario Grant from the Spanish Ministry of Education and Science, in 2002.

• • •