A Measure of Structural Complexity of Hierarchical Fuzzy Systems Adapted from Software Engineering

Tajul Rosli Razak[1,2], Jonathan M. Garibaldi[1] and Christian Wagner[1]
[1]Laboratory for Uncertainty in Data and Decision Making (LUCID),
School of Computer Science, University of Nottingham, United Kingdom.
[2]Faculty of Computer and Mathematical Science, Universiti Teknologi MARA, Perlis, Malaysia.
Email: [1]{tajul.razak,jon.garibaldi,christian.wagner}@nottingham.ac.uk, [2]tajulrosli@uitm.edu.my

***Abstract*—Hierarchical fuzzy systems (HFSs) have been seen as an effective approach to reduce the complexity of fuzzy logic systems (FLSs), largely as a result of reducing the number of rules. However, it is not clear completely how complexity of HFSs can be measured. In FLSs, complexity is commonly expressed using a multi-factorial approach, taking into consideration the number of rules, variables, and fuzzy terms. However, this may not be the best way to assess complexity in HFSs that have structures involving multiple subsystems, layers and different topologies. Thus far, structural complexity associated with the structure of HFSs has not been discussed. In the field of software engineering (SE), a complexity measure has been proposed to measure program complexity. This measure uses the concept of graph theory complexity, which considers the control structure complexity. The measure can also be applied to assess the complexity of a collection of programs known as a hierarchical nest. In this paper, we present an approach to mapping an SE complexity measure to HFS design. The approach includes several mapping alternatives that are outlined and illustrated using different HFS designs. This study contributes a new approach for the first time to assessing structural complexity in HFSs based on an approach from SE complexity measure.**

***Index Terms*—Hierarchical fuzzy systems, Complexity, Structural complexity, Mapping Process.**

I. Introduction

One of the most important motivations for using FLSs for system modelling is that they use linguistic variables and rules [1] that are easy to understand. Moreover, FLSs are also good at capturing the complexity of a wide range of problems through their linguistic modelling and approximate reasoning capabilities [2]. However, FLSs pose significant challenges, including the *curse of dimensionality*, whereby the number of required rules and the model complexity commonly increase exponentially with the number of input variables [3], [4], thus potentially reducing the transparency and interpretability of FLSs. In exploring this problem, several methods have been proposed for reducing the size of the rule-based structure in FLSs, such as that of rule selection [5], feature selection [6], rule interpolation [7], singular-value decomposition-QR [8], evolutionary algorithms [9], fuzzy similarity measures [4], rule learning [10] and HFSs [11].

HFSs were introduced as an approach to overcome the *curse of dimensionality* arising in conventional FLSs [11], [12]. HFSs have been shown to be an effective way to reduce the number of rules in FLSs, which is argued to reduce the model complexity and improve its interpretability. Indeed, in literature, it has been claimed that reducing complexity is a way of improving model interpretability in FLSs [4], [13], [14]. Thus, complexity appears to be a key component in the interpretability of FLSs. Despite the fact that people usually measure complexity in FLSs by looking principally at the rule-based complexity level [4], [15]–[17], this is not an ideal way to assess complexity, especially for HFSs which feature an overall structure comprising multiple subsystems, layers and different topologies.

Whilst it is an important consideration, it would appear that complexity measures associated with HFS structures have not been discussed in much depth. In the literature, a complexity measure has been proposed in the field of software engineering (SE), namely McCabe's measure [18], which considers the complexity of the control structure. This has been used to measure program complexity through the control graph as well as through decomposition into constituent sub-routines. Such decomposition of a complex program may be viewed to be a similar concept to that of HFSs. Thus, in this paper, we intend to present an approach to map an SE complexity measure to HFS design. Specifically, we focus on translating all elements in McCabe's measure to the structures of HFS designs that have multiple subsystems, layers and different topologies.

The rest of this paper is organised as follows. Section II explains the background on the state of the art of interpretability and complexity in FLSs and other fields, and also hierarchical fuzzy systems. Section III concisely outlines the McCabe measure from SE. Section IV introduces a new approach to mapping the McCabe measure to the elements of HFS design. Section V demonstrates the new complexity measure of HFS with real-world example, namely the Rotary crane (as used in [19]). Finally, Sections VI and VII present results, conclusion and future works.

II. Background

In this section, we briefly provide background in respect to interpretability and complexity in FLSs, complexity in other fields and HFSs.

A. Interpretability

Interpretability refers to the capability of FLSs to express the behavior of the system in an understandable way [20].

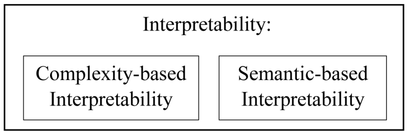

Fig. 1. A taxonomy to analyse interpretability of FLSs. Adapted from [21].

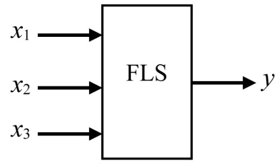

Fig. 2. Fuzzy Logic System

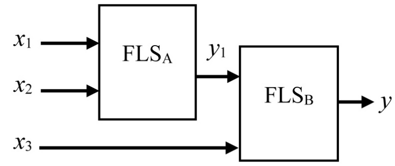

Fig. 3. Hierarchical Fuzzy System

Gacto et al. introduced a taxonomy for assessing interpretability of FLSs [21], that includes the two key components; (i) complexity-based interpretability; and (ii) semantic-based interpretability as shown in Fig. 1. Complexity-based interpretability is devoted to decreasing the complexity of the obtained model (usually measured as the number of rules, variables, labels per rule, and other factors). Meanwhile, semantic-based interpretability is dedicated to preserving the semantics associated with the membership functions (MFs), to ensure semantic integrity by imposing constraints on the MFs or approaches considering measures such as distinguishability, coverage and other factors.

Following Gacto et al., it is generally accepted that complexity is an essential component to determine the interpretability of FLSs [21]. Thus far, complexity has often been used as an indirect measurement of the interpretability of FLSs. Indeed, several researchers claim that the reduction of complexity in a system can lead to better interpretability of the fuzzy systems [4], [13], [14], [22].

B. Complexity in FLSs

In FLSs, complexity could be related to the specific problem described by the fuzzy model. In other words, from the structural analysis of a knowledge base, we should expect to gain information concerning the complexity of the underlying problem [23]. Complexity of conventional FLSs is usually evaluated by a multi-factorial approach, taking into consideration the number of rules, variables, and fuzzy terms.

1) Hierarchical Fuzzy Systems: The term 'hierarchical' has generally been used to refer to a complex system in which each of the subsystems is subordinated by an authority relating to the system it belongs to [24]. In fuzzy logic systems, HFSs are defined by composing the input variables into a collection of low-dimensional fuzzy logic subsystems [11], [12]. Also, HFSs can be illustrated as a cascade structure where the output of each layer is considered as an input to the following layer as shown in Fig. 3. Moreover, a system that goes from one layer as shown in Fig. 2 to two layers as in Fig. 3 has fewer rules than the one in one layer [25]. The most extreme reduction of rules will be if the structure of HFS has two input variables for each low dimensional FLS, with the number of layers being one fewer than the number of inputs [11].

Therefore, HFSs have been shown to be an effective approach to reducing the number of rules in FLSs, and thus reducing the complexity model. However, the reduction in the complexity of each individual subsystem is, to a degree, countered by the existence of a structure (and associated arrangement) of multiple connected subsystems. It is not clear how the complexity in HFSs in relation to their structure can be measured. To date, this issue has not been explored to any great degree.

C. Complexity in others fields

Complexity arises from either the structure of the interactions between very similar units or from the units and the interactions themselves having specific characteristics. In both cases, the abstract representation of a complex system can be achieved by a collection of nodes (units) and edges (representing interactions between the units) forming a network (or graph) [26].

One common view of a general complexity measure is that it is dependent on the number of structural features contained within an organisation rather than simply on the number of its basic elements [27]; this idea is also known as structural complexity.

Structural complexity is an attribute of any general type of system. This attribute can be assessed by different measures, and it is often linked to interaction among systems' properties such as nodes, edges and networks [26], [28]. Several measures that cover the structural complexity have been proposed to the other fields such as SE [18], circuit design [29] and Psychology [30].

III. A Complexity Measure in Software Engineering

McCabe [18] proposed a complexity measure that uses various concepts from graph-theory. The complexity measure was developed to measure and control the number of paths through a program using a cyclomatic number. The cyclomatic number $v(G)$ of graph G with n vertices / nodes, e edges, and p connected components can be expressed as follows:

$$v(G) = e - n + 2p \tag{1}$$

The overall strategy involves measuring the complexity of a program by computing the number of linearly independent paths $v(G)$. In a strongly connected graph G, the cyclomatic number $v(G)$ is equal to the maximum number of linearly independent paths in G. Note that a graph is define as strongly connected if every node is reachable from every other node. Other properties of cyclomatic complexity include: (i) $v(G) \geq 1$; (ii) G has only one path if and only if $v(G) = 1$; (iii) inserting a new edge (e) in G increases $v(G)$; and (iv) for fully connected graphs G, the maximum number of edges (e) is computed as:

$$e = \frac{n(n-1)}{2} \tag{2}$$

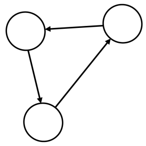

Fig. 4. A control graph, $C1$.

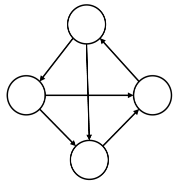

Fig. 5. A control graph, $C2$.

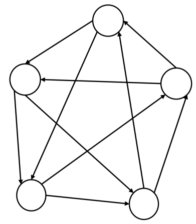

Fig. 6. A control graph, $C3$.

Figs. 4, 5 and 6 present several examples of the control graph (C). The cyclomatic complexity of each C can be computed using the cyclomatic number presented in (1).

For example, the complexity of control graphs $C1$, $C2$ and $C3$ can be computed using (1) as follows:

$$v(C1) = e - n + 2p = 3 - 3 + 2 \times 1 = 2$$
$$v(C2) = e - n + 2p = 6 - 4 + 2 \times 1 = 4$$
$$v(C3) = e - n + 2p = 10 - 5 + 2 \times 1 = 7$$

A control graph is more complex when the cyclomatic complexity $v(G)$ value is large and less complex when the $v(G)$ value is small. In the example above, based on $v(G)$ values, we may say that $C3$ is more complex than $C2$ and $C1$ ($C3 > C2 > C1$) in terms of their control structure complexity. It is also true that the increasing number of nodes (n) and edges (e) from the control graph $C1$ to $C3$ would result in increased complexity.

The control graphs ($C1, C2$, and $C3$) are presented in increasing order of complexity in order to suggest the correlation between the complexity numbers and an intuitive notion of control flow complexity.

A. Decomposition – Hierarchical Nest

As discussed earlier, p is the number of connected components as presented in (1). The way a program control graph is defined would result in all control graphs having only one connected component, that is $p = 1$. However, for control graphs (C) that have more than one connected component ($p \neq 1$), the collection of control graphs may be described as a hierarchical nest of subroutines. Thus, the complexity of a collection of control graphs C with p connected components, is equal to the summation of their complexities as $C1 \cup C2 ... \cup Cp$. It also can be expressed as follows:

$$v\Big(\bigcup_{i=1}^{p}\Big) = e - n + 2p \qquad (3)$$

For clarity, we used two cases as examples to illustrate (3) in the following subsection.

1) Case 1: In this case, we assume that a main program $M1$ and two subroutines $A1$ and $B1$ which have control structure as shown in Fig. 7. Let us denote the total graph in Case 1 with 3 connected components as $M1 \cup A1 \cup B1$. Also, this total graph will consist of $e = 13$ (only solid edges), $n = 13$ and $p = 3$. Now, since $p = 3$, the complexity of the total graph in Case 1 can be computed using (3) as follows:

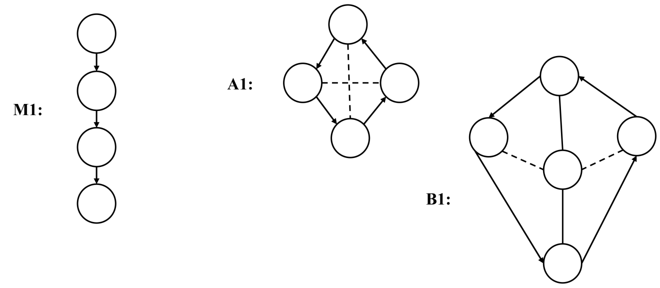

Fig. 7. Case 1: one main program M1, with two subroutines A1 and B1 (only include solid edges). Case 2: one main program M1, with two subroutines A1 and B1 (include solid and dotted edges). Adapted from McCabe [18].

$$\begin{aligned} v(M1 \cup A1 \cup B1) &= e - n + 2p \\ &= 13 - 13 + 2 \times 3 = 6 \end{aligned}$$

2) Case 2: Similarly, to Case 1, we assume that a main program $M1$ and two subroutines $A1$ and $B1$ which have control structures as shown in Fig. 7. However, for this case, this total graph will consist of $e = 17$ (include solid and dotted edges), $n = 13$ and $p = 3$. Now, since $p = 3$, the complexity the total graph in Case 2 can be computed using (3) as follows:

$$\begin{aligned} v(M1 \cup A1 \cup B1) &= e - n + 2p \\ &= 17 - 13 + 2 \times 3 = 10 \end{aligned}$$

In general, the computed complexity of the collection of programs in Case 2 ($v(M1 \cup A1 \cup B1) = 10$) is higher than the computed complexity of the collection of programs in Case 1 ($v(M1 \cup A1 \cup B1) = 6$). This could indicate that the collection of programs in Case 2 are more complex than in Case 1 in terms of the control structure complexity. Although both Case 1 and Case 2 have the same number of nodes (n) and the number connected component (p), the difference in the number of edges (e) explains the difference in the complexity they present.

IV. Mapping A Complexity Measure from SE to HFSs

In this paper, we aim to translate a measure of the complexity from SE (McCabe measure) to the HFSs' design. Particularly, we focus on proposing an approach to mapping all the components in a complexity measure from SE to the design of HFSs. However, some questions arise including: "How is the complexity measure from SE related to the design of HFSs?" and "How can a complexity measure from SE map to the design of HFSs?"

Clearly, one may argue whether the selection of a complexity measure from SE (McCabe measure) is appropriate, given that there are other measures available. However, for an initial step, in this paper we propose to use the McCabe measure based on the following reasons: (i) A control graph theory that was used in McCabe is a similar structure to the fuzzy cognitive map (FCM). FCMs are fuzzy-graph structures in FLSs that were used for representing causal reasoning [31]. However, none of the indices or measures have been proposed in measuring FCMs' structure of complexity. (ii) The McCabe measure takes into account the structural complexity in their measures. (iii) The McCabe measure can also be applied to assess the complexity of a collection of control graphs known as a hierarchical nest (see Section III-A). Moreover, the decomposition in the McCabe measure is very similar to the concept with the structure of HFSs that may feature an arbitrary number of multiple subsystems.

However, to map McCabe measure to the design of HFSs is a non-trivial task. Although the elements in both the structures of the control graph and HFSs' design have similarities, there are some that are not entirely similar. Thus, we propose an approach of mapping the McCabe measure to the HFSs' design as explained in the following steps.

A. Step 1: Identify elements in HFSs' Design

First, let us denote all the elements which are contained in HFSs' design (see an example of HFS design in Fig. 9): (i) x indicates the number of input variables in HFSs, (ii) l indicates the number of layers in HFSs, (iii) a indicates the number of input-output (IO) connections in HFSs, and (iv) s indicates the number of subsystems in HFSs.

B. Step 2: Mapping Edges (e) to the HFSs' Design

The first element of the McCabe measure is e, that is the number of edges in the control graph that need to be mapped to the HFSs' design. The mapping possibilities may be viewed as follows:

$$e \rightarrow (x \quad \text{or} \quad l \quad \text{or} \quad a \quad \text{or} \quad s).$$

Intuitively, one may say that the element of a (number of the IO connections in an HFS) could be the best option for mapping to the element e (number of edges). However, one might also argue that the element l (number of layers in HFSs' design) could be a better mapping to e. Thus, delaying the best choice, we present two alternative mappings which can now be viewed as follows:

$$e \rightarrow a \quad \text{OR} \quad e \rightarrow l.$$

C. Step 3: Mapping Nodes (n) to the HFSs' Design

The next element of the McCabe measure is n, that is the number of nodes in the control graph that will be mapped. Taking into account the mapping in Step 2, there are two alternative mapping possibilities:

$$\text{EITHER} \quad n \rightarrow (x \text{ or } l \text{ or } s) \quad \text{OR} \quad n \rightarrow (x \text{ or } a \text{ or } s).$$

For the first alternative, it seems most reasonable to choose s (number of subsystems) in HFSs' design to map to n (number of nodes) in the McCabe measure. For the second alternative, we choose x (number of inputs) to map to n. Consequently, both alternative mappings can now be represented as follows:

$$n \rightarrow s \quad \text{OR} \quad n \rightarrow x.$$

D. Step 4: Mapping Components (p) to the HFSs' Structure

The final element of the McCabe measure is p, that is the number of connected components in the control graph. Now, given the assignments in Steps 2 and 3, the remaining possibilities are:

$$\text{EITHER} \quad p \rightarrow (x \text{ or } l) \quad \text{OR} \quad p \rightarrow (a \text{ or } s).$$

The number of connected components in the control graph (p) is a very similar concept as the number of layers (l) in HFSs. In HFSs, all subsystems are placed and attached through the HFS layers. However, one may argue that the number of connected components in the control graph can also be related to the number of subsystems (s) in HFSs. Therefore, for the first alternative, we map l to p; whilst in the second alternative, we map s to p. Hence, both alternative mappings now are presented as follows:

$$p \rightarrow l \quad \text{OR} \quad p \rightarrow s.$$

E. Step 5: Construct complexity measure of HFSs – (C_{HFS})

Following the mapping selection in Steps 2, 3, and 4, two alternative mappings are obtained, namely the first alternative (from now on referred to as A), which constitutes the mapping $e \rightarrow a$, $n \rightarrow s$ and $p \rightarrow l$, whereas the second alternative (termed B) which constitutes $e \rightarrow l$, $n \rightarrow x$ and $p \rightarrow s$. Finally, a complexity measure for HFSs is assembled using both alternative mappings, and now (3) appears as follows:

$$\mathrm{C}_{HFS}\mathrm{A}\Big(\bigcup_{i=1}^{s}\Big) = a - s + 2l \tag{4}$$

$$\mathrm{C}_{HFS}\mathrm{B}\Big(\bigcup_{i=1}^{s}\Big) = l - x + 2s \tag{5}$$

where C_{HFS}A and C_{HFS}B indicate the non-normalised structural complexity in HFSs using alternative mappings A and B respectively.

F. Step 6: Normalise the complexity measure output of HFSs – ($\tilde{C}_{HFS}$)

McCabe's measure ranges from $[1, \infty]$ whereas C_{HFS}A and C_{HFS}B range from $[1, \infty]$ and $[-\infty, \infty]$ respectively. Whilst there is no inherent problem with this, other measures of interpretability for FLSs (such a Nauk's index [32] or Alonso et al. [33]) have traditionally been defined over $[0, 1]$. Given this, it will be easier to subsequently combine a HFS complexity measure with other components of interpretability if it lies over the same range. Thus, we would like to normalise the complexity values of C_{HFS}A and C_{HFS}B to be in the range $[0, 1]$. There are several alternative functions that can be used but for this paper, we will make an arbitrary choice to use one of the principal functions in mathematics, which is an inverse trigonometric function. This function can be expressed as follows:

$$\tilde{\mathrm{C}}_{HFS} = \frac{\arctan(\mathrm{C}_{HFS}) \times 2}{\pi} \tag{6}$$

TABLE I
DESCRIPTION OF THE VARIOUS HFS DESIGNS FOR ROTARY CRANE

Descriptions	Rotary Crane				
	FLS	HFS-1	HFS-2	HFS-3	HFS-4
Number of inputs (x)	6	6	6	6	6
Number of layers (l)	1	2	2	3	5
Number of subsystems (s)	1	3	4	5	5
Number of the IO connections (a)	7	9	10	11	11

For the case of the $\mathrm{C}_{HFS}\mathrm{B}$ that has a different range to $\mathrm{C}_{HFS}\mathrm{A}$, we also use the inverse trigonometric function to normalise its complexity values to be in the range $[0, 1]$. However, (6) appears as follow:

$$\tilde{\mathrm{C}}_{HFS} = \frac{\arctan(\mathrm{C}_{HFS}) \times 2 + \pi}{2\pi} \tag{7}$$

where $\arctan()$ is the inverse of the tangent function, C_{HFS} represents the values of $\mathrm{C}_{HFS}\mathrm{A}$ or $\mathrm{C}_{HFS}\mathrm{B}$, and $\tilde{\mathrm{C}}_{HFS}$ indicates the normalised structural complexity in HFSs.

An HFS model is less complex when the $\tilde{\mathrm{C}}_{HFS}$ is close to 0 and more complex when $\tilde{\mathrm{C}}_{HFS}$ is close to 1, in terms of structural complexity.

V. EXPERIMENTS AND RESULTS

In this section, we will illustrate the complexity measure $\tilde{\mathrm{C}}_{HFS}$ to HFSs design using the real-world example. Note that this example is not used to show any benefits of the hierarchical approach. In this experiment, we used an example of HFSs' design, namely the rotary crane example, as used in [19], which is a load rotation and hoisting system. A summary of the rotary crane for various HFS designs is presented in Table I. This experiment aims to demonstrate the complexity measure ($\tilde{\mathrm{C}}_{HFS}$) as proposed in Section IV, using the examples above and described in the following subsection.

A. Measuring the complexity of HFSs' design in the Rotary crane

As can be seen in Table I, there are five HFSs' designs use for the rotary crane example, namely a flat system (FLS), HFS's design 1 (HFS-1), HFS's design 2 (HFS-2), HFS's design 3 (HFS-3) and HFS's design 4 (HFS-4). In this section, the complexity of these designs are assessed using the proposed measure as in (4) and (5), and illustrated below.

1) The Flat FLS: In this case, the structure of a flat FLS design consists of $x = 6$, $l = 1$, $s = 1$, and $a = 7$, as shown in Fig. 8. According to equation (4) and (5), the suggested complexity for FLS are:

$$\begin{aligned}\mathrm{C}_{HFS}\mathrm{A}(\mathrm{FLS}) &= a - s + 2l \\ &= 7 - 1 + 2 \times 1 = 8 \\ \mathrm{C}_{HFS}\mathrm{B}(\mathrm{FLS}) &= l - x + 2s \\ &= 1 - 6 + 2 \times 1 = -3\end{aligned}$$

TABLE II
A SUMMARY OF STRUCTURAL COMPLEXITY MEASURE ($\tilde{\mathrm{C}}_{HFS}$) FOR HFSS' DESIGN OF THE ROTARY CRANE

Rotary Crane	Structural complexity measure				Number of rules
	Mapping alternative: A		Mapping alternative: B		
	$\mathrm{C}_{HFS}\mathrm{A}$	$\tilde{\mathrm{C}}_{HFS}$	$\mathrm{C}_{HFS}\mathrm{B}$	$\tilde{\mathrm{C}}_{HFS}$	
FLS	8	0.9208	-3	0.1024	729
HFS-1	10	0.9365	2	0.8524	63
HFS-2	10	0.9365	4	0.9220	54
HFS-3	12	0.9471	7	0.9548	45
HFS-4	16	0.9603	9	0.9648	45

2) HFS-1: For this case, the HFS-1 structure consists of $x = 6$, $l = 2$, $s = 3$, and $a = 9$, as shown in Fig. 9. According to equation (4) and (5), the suggested complexity for HFS-1 are:

$$\begin{aligned}\mathrm{C}_{HFS}\mathrm{A}(\text{HFS-1}) &= a - s + 2l \\ &= 9 - 3 + 2 \times 2 = 10 \\ \mathrm{C}_{HFS}\mathrm{B}(\text{HFS-1}) &= l - x + 2s \\ &= 2 - 6 + 2 \times 3 = 2\end{aligned}$$

Similarly, for HFS-2, HFS-3 and HFS-4. Finally, the results of $C_{HFS}\mathrm{A}$ and $C_{HFS}\mathrm{B}$ for these designs are then normalised using (6) and (7) as shown in Table II. For comparison, Table II also provides the number of rules in each fully specified rotary crane system, based on using three membership functions in each variable.

VI. DISCUSSIONS

This paper has presented a new approach to mapping a complexity measure from SE to HFSs' design. The approach introduces two key mapping alternatives, namely the mapping A and B. A detailed of these mapping alternatives consisted of six steps as described in Section IV. Obviously, the proposed mapping alternatives adopted in this paper is not unique and different methods may be adopted. However, as an initial step, the mapping alternatives are based on the similarities in both the structures of the control graph and HFSs' design. It is clear that further work is still required to establish ground truth on this mapping strategies.

The complexity measure for HFSs was demonstrated using different HFS designs in real-world example namely rotary crane as shown in Section V. The summaries of a complexity measure ($\tilde{\mathrm{C}}_{HFS}$) using alternative mappings A and B, to the different configuration of the rotary crane as shown in Table II.

For the case of mapping A, the measure produces a result that is *equally complex* for the design of HFS-1 and HFS-2 although they have a different number of subsystems (s) and number of IO connections (a). Some possible explanations for this finding include: (i) although there is equally incrementing for a and s in the design of HFS-1 to HFS-2, the number of layers (l) is still the same in both designs; and (ii) following the configuration in mapping A, the complexity of design in the HFS-1 ($9-3+2\times2 = 10$) and HFS-2 ($10-4+2\times2 = 10$) is equivalent. On the other hand, for the case of mapping B,

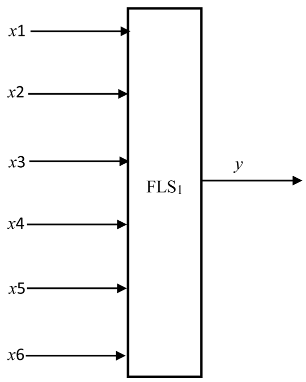

Fig. 8. Rotary crane: A flat FLS

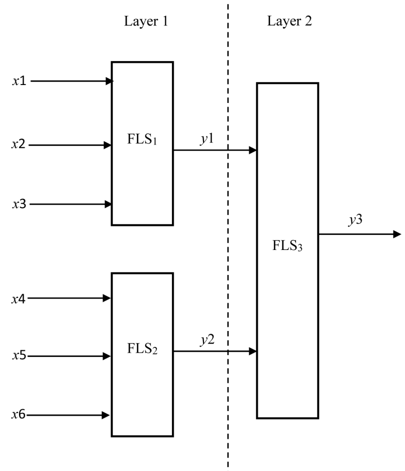

Fig. 9. Rotary crane: HFS's design 1 (HFS-1)

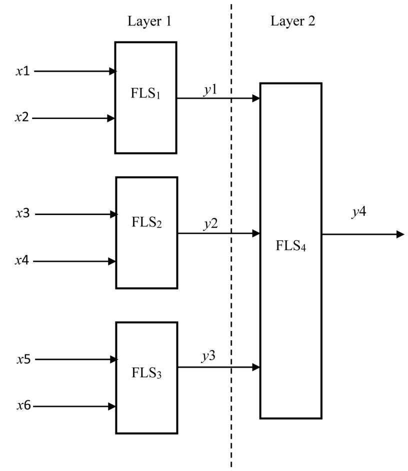

Fig. 10. Rotary crane: HFS's design 2 (HFS-2)

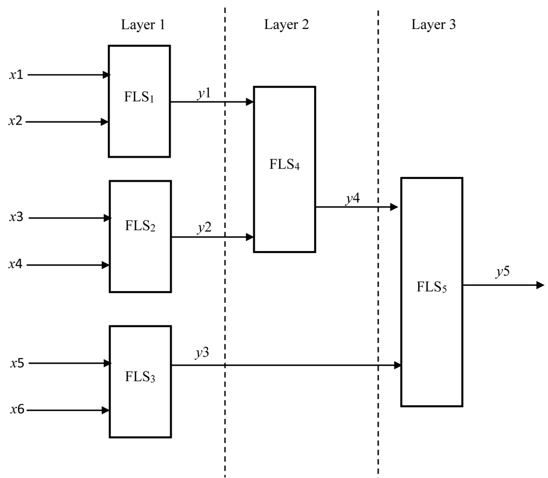

Fig. 11. Rotary crane: HFS's design 3 (HFS-3)

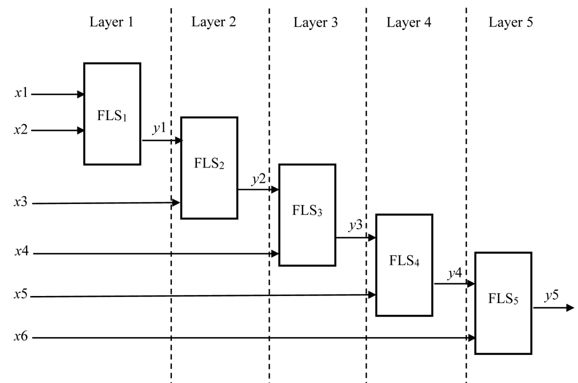

Fig. 12. Rotary crane: HFS's design 4 (HFS-4)

the measure produces a different complexity result for HFS-1 and HFS-2 that indicate HFS-2 ($\tilde{C}_{HFS} = 0.922$) is *more complex* than HFS-1 ($\tilde{C}_{HFS} = 0.8524$), in term of structural complexity. It seems possible that these results are due to a different number of subsystems for both HFS-1 and HFS-2 as shown in Figs. 9 and 10 respectively, which influence their structural complexity. Overall, the findings from the measure using both mapping A and B agreed that in HFSs, the complexity increases exponentially with the number of HFS layers regarding the structural complexity. The finding also follows intuition in the sense that one expects the HFS-4 to be more complicated than all designs, because of their number of layers.

Perhaps surprisingly, if we only consider the number of rules as the complexity assessment for the Rotary crane designs (see Table II), it is shown that HFS-3 and HFS-4 were *equally less complex* than other designs regarding rule-based complexity. However, in contrast to rule-based complexity, the structural complexity measure using the proposed $\tilde{C}_{HFS}$ showed that the HFS-4 is *more complex* than other designs for the Rotary crane. Also, although HFS-3 and HFS-4 have the same number of rules, the results of $\tilde{C}_{HFS}B$ indicate that they are different regarding structural complexity. On the other hand, $\tilde{C}_{HFS}A$ indicates the same structural complexity for HFS-1 and HFS-2, despite the fact that HFS-2 features fewer rules (54 compared to 63). Therefore, in this context, it is clear that further work is required to establish a comprehensive measurement of complexity in HFSs, i.e. merging rule-based complexity and structural complexity.

Obviously, there are limitations on this measure as it does not capture structure complexity perfectly. For example, in the case of the measure with mapping A as in (4), it is important to notice that $a-s$ is an invariant for any HFS when intermediate systems have a single output. Thus, the number of intermediate variables is always equal to the number of subsystems minus 1. Consequently, $a - s$ is always equal to the number of input variables. Hence, the final result is that our structural complexity measure only changes with the number of layers. Therefore, it is clear that more research on this measure needs to be undertaken in future work.

VII. Conclusion

In conclusion, we have proposed a new approach to assessing structural complexity in HFSs using an existing complexity measure from software engineering, mapped to the design of HFSs, which feature multiple subsystems, layers and different topologies. In order to do that, we introduced two mapping alternatives, namely mapping A and B, to the proposed complexity measure ($\tilde{C}_{HFS}$).

A real-world example was used to illustrate the measure $\tilde{C}_{HFS}$ to the different HFS designs. The measure shows promise for measuring the structural complexity in HFSs using both mapping A and B. Based on the current evidence, we tentatively suggest the use of mapping B which is based on mapping $e \rightarrow l$, $n \rightarrow x$ and $p \rightarrow s$, to the proposed complexity measure $\tilde{C}_{HFS}$ as it appears to behave in an intuitive and natural manner.

In future research, we will focus on incorporating this complexity measure for HFSs with other aspects of complexity, such as the rule-based complexity, in order to reach a more comprehensive understanding (together with measures of) the overall interpretability of HFSs. Further work may also be undertaken into exploring the use of alternative measures of structural complexity taken from other fields.

Acknowledgment

We gratefully acknowledge the support of the Universiti Teknologi Mara (UiTM) and Ministry of Higher Education, Malaysia for sponsoring the first author to pursue his study at the University of Nottingham (UoN), United Kingdom.

References

[1] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Trans. Syst., Man, and Cybern.*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[2] A. Gegov, D. Sanders, and B. Vatchova, "Complexity management methodology for fuzzy systems with feedforward rule bases," *Int. J. of Knowledge-based and Intell. Engineering Syst.*, vol. 19, no. 2, pp. 83–95, jan 2015.

[3] S.-M. Zhou, J. Garibaldi, R. John, and F. Chiclana, "On Constructing Parsimonious Type-2 Fuzzy Logic Systems via Influential Rule Selection," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 3, pp. 654–667, jun 2009.

[4] Y. Jin, "Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 212–221, apr 2000.

[5] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 260–270, 1995.

[6] O. Cordón, M. J. Del Jesus, F. Herrera, L. Magdalena, and P. Villar, "A Multiobjective Genetic Learning Process for joint Feature Selection and Granularity and Contexts Learning in Fuzzy Rule-Based Classification Systems," in *Interpretability issues in fuzzy modeling*. Springer Berlin Heidelberg, 2003, pp. 79–99.

[7] L. Koczy and K. Hirota, "Size reduction by interpolation in fuzzy rule bases," *IEEE Trans. Syst. Man Cybern. - Part B*, vol. 27, no. 1, pp. 14–25, 1997.

[8] Q. Liang and J. M. Mendel, "Designing interval type-2 fuzzy logic systems using an SVD-QR method: Rule reduction," *Int. J. Intell. Syst.*, vol. 15, no. 10, pp. 939–957, oct 2000.

[9] Y. Jin, "Decentralized adaptive fuzzy control of robot manipulators," *IEEE Trans. Syst. Man Cybern. - Part B*, vol. 28, no. 1, pp. 47–57, 1998.

[10] S.-M. Zhou and J. Gan, "Low-level interpretability and high-level interpretability: a unified view of data-driven interpretable fuzzy system modelling," *Fuzzy Sets Syst.*, vol. 159, no. 23, pp. 3091–3131, dec 2008.

[11] G. V. S. Raju, J. Zhou, and R. Kisner, "Hierarchical fuzzy control," *Int. J. Contr.*, vol. 54, no. 5, pp. 1201–1216, nov 1991.

[12] G. Raju and J. Zhou, "Adaptive Hierarchical Fuzzy Controller," *IEEE Trans. Syst., Man, and Cybern.*, vol. 23, no. 4, pp. 973–980, 1993.

[13] M. Setnes and R. Babuska, "Rule base reduction: some comments on the use of orthogonal transforms," *IEEE Trans. Syst., Man, and Cybern. - Part C*, vol. 31, no. 2, pp. 199–206, may 2001.

[14] J. Yen and L. Wang, "Simplifying fuzzy rule-based models using orthogonal transformation methods," *IEEE Trans. Syst., Man, and Cybern. - Part B*, vol. 29, no. 1, pp. 13–24, 1999.

[15] H. Ishibuchi and Y. Nojima, "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning," *Int. J. Approx. Reas.*, vol. 44, no. 1, pp. 4–31, jan 2007.

[16] M. Delgado, F. Von Zuben, and F. Gomide, "Multi-objective decision making: towards improvement of accuracy, interpretability and design autonomy in hierarchical genetic fuzzy systems," in *Proc. Int. Conf. FUZZ-IEEE*, vol. 2. IEEE, 2002, pp. 1222–1227.

[17] X. Zhu, J. Li, D. Wu, H. Wang, and C. Liang, "Balancing accuracy, complexity and interpretability in consumer credit decision making: A C-TOPSIS classification approach," *Knowledge-Based Syst.*, vol. 52, pp. 258–267, nov 2013.

[18] T. McCabe, "A Complexity Measure," *IEEE Trans. on Software Engineering*, vol. SE-2, no. 4, pp. 308–320, dec 1976.

[19] N. Masmoudi, C. Rekik, M. Djemel, and N. Derbel, "Optimal Control for Discrete Large Scale Nonlinear Systems Using Hierarchical Fuzzy Systems," in *2010 Second Int. Conf. on Mach. Learning and Comput.* IEEE, 2010, pp. 91–95.

[20] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, "Interpretability Improvements to Find the Balance Interpretability-Accuracy in Fuzzy Modeling: An Overview," in *Interpretability issues in fuzzy modeling*. Springer Berlin Heidelberg, 2003, pp. 3–22.

[21] M. J. Gacto, R. Alcalá, and F. Herrera, "Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures," *Inf. Sci.*, vol. 181, no. 20, pp. 4340–4360, 2011.

[22] T. R. Razak, J. M. Garibaldi, C. Wagner, A. Pourabdollah, and D. Soria, "Interpretability and Complexity of Design in the Creation of Fuzzy Logic Systems A User Study," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 420–426.

[23] C. Mencar, C. Castiello, R. Cannone, and A. M. Fanelli, "Interpretability assessment of fuzzy knowledge bases: A cointension based approach," *Int. J. Approx. Reas.*, vol. 52, pp. 501–518, 2011.

[24] H. A. Simon, "The Architecture of Complexity," pp. 467–482, 1962.

[25] T. R. Razak, J. M. Garibaldi, C. Wagner, A. Pourabdollah, and D. Soria, "Interpretability indices for hierarchical fuzzy systems," in *Proc. IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE)*. IEEE, jul 2017, pp. 1–6.

[26] E. Mones, L. Vicsek, and T. Vicsek, "Hierarchy Measure for Complex Networks," *PLoS ONE*, vol. 7, no. 3, p. e33799, mar 2012.

[27] W. G. Bail and M. V. Zelkowitz, "Program complexity using Hierarchical Abstract Computers," *Computer Languages*, vol. 13, no. 3-4, pp. 109–123, jan 1988.

[28] A. Gegov, *Fuzzy Networks for Complex Systems*, ser. Studies in Fuzziness and Soft Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 259.

[29] K.-H. Cho, S. H. Jung, and G.-Y. Kim, "Evolutionary design of complex digital circuits based on hierarchical module composition and predominant component prevention," *Electronics Letters*, vol. 51, no. 20, pp. 1568–1570, oct 2015.

[30] M. Commons, "Introduction to the model of hierarchical complexity." *Behavioral Development Bulletin*, vol. 13, no. 1, pp. 1–6, 2007.

[31] B. Kosko, "Fuzzy cognitive maps," *International Journal of Man-Machine Studies*, vol. 24, no. 1, pp. 65–75, jan 1986.

[32] D. Nauck, "Measuring interpretability in rule-based classification systems," in *Proc. 12th IEEE Int. Conf. Fuzzy Syst.* IEEE, 2003, pp. 196–201.

[33] J. M. Alonso, S. Guillaume, and L. Magdalena, "A hierarchical fuzzy system for assessing interpretability of linguistic knowledge bases in classification problems," in *Proc. PIMU*, 2006, pp. 348–355.