

# Rule base compression in fuzzy systems by filtration of non-monotonic rules

Alexander Gegov<sup>a,\*</sup>, Neelamugilan Gobalakrishnan<sup>a</sup> and David Sanders<sup>b</sup>

<sup>a</sup>*School of Computing, University of Portsmouth, Portsmouth, UK*

<sup>b</sup>*School of Engineering, University of Portsmouth, Portsmouth, UK*

**Abstract.** This paper proposes a rule base compression method for Mamdani fuzzy systems with non-monotonic rules. The method is based on filtration of non-monotonic rules whereby the redundant computations in the fuzzy inference with respect to the crisp values of the inputs to the fuzzy system are removed. The method identifies all redundant rules after fuzzification and removes them while preserving the defuzzified output from the fuzzy system for each simulation cycle. In comparison to other rule base reduction methods, this method does not compromise the solution and is more efficient in terms of on-line computations within a wide operating range. The method processes the rule base during simulation cycles by contracting it to a rule base of a smaller size at the start of each inference stage and then expanding it to its original size before the next fuzzification stage.

**Keywords:** Fuzzy systems, complexity theory, simulation, data compression, control systems

## 1. Introduction

Fuzzy systems are usually good at capturing the qualitative complexity of a wide range of problems by means of their linguistic modeling and approximate reasoning capabilities. However, this comes at a price because the associated operations during fuzzification, inference and defuzzification increase the quantitative complexity of the solution to these problems. This price gets even higher as the amount of fuzzy operations increases as a result of the increased number of rules in the fuzzy system.

The number of rules in a fuzzy system is often an exponential function of the number of inputs to the system and the number of linguistic values that these inputs can take [5, 21, 28, 40]. This exponential function has been used as a main indicator for the quantitative complexity of the associated fuzzy system. However, this is

a fairly rough indicator because the quantitative complexity depends on the overall amount of operations during fuzzification, inference and defuzzification. For example, a 4-input fuzzy system with 2 linguistic values per input has the same number of 16 rules as a 2-input fuzzy system with 4 linguistic values per input but the amount of operations in the first system is about twice as big as the one in the second system due to the twice bigger number of inputs in the rules.

There has been a growing interest recently in complexity issues of fuzzy systems [1, 10, 20, 29]. This is due to the fact that fuzzy systems are already more widely used in large-scale applications where their quantitative complexity becomes more obvious. In particular, many methods have been developed for reducing this quantitative complexity. These methods are known as rule base reduction methods as they reduce the number of rules by reducing the number of inputs or the number of linguistic values that these inputs can take. The main objective in this case is to suppress the associated exponential function. These rule base reduction methods can be classified into eight groups that are discussed below.

---

\*Corresponding author. Alexander Gegov, School of Computing, University of Portsmouth, Buckingham Building, Portsmouth PO1 3HE, UK. Tel.: +44 0 2392 846381; Fax: +44 0 2392 846364; E-mail: alexander.gegov@port.ac.uk.

The methods in group 1 are aimed at removing less significant or merging similar linguistic values [12, 27]. From these two strands, the one based on removal of linguistic values is more straightforward but it involves a higher risk as a result of the removal of the associated fuzzy set. On the other hand, the strand based on merging of linguistic values is more difficult for application due to the necessity to define a new fuzzy set for each of the merged linguistic values.

The methods in group 2 are aimed at removing less significant or merging similar inputs [22, 35]. From these two strands, the one based on removal of inputs is more straightforward but it involves a higher risk as a result of the removal of the associated physical variable. On the other hand, the strand based on merging of inputs is more difficult for application due to the necessity to justify physically the merging of the associated variables.

The methods in group 3 are based on singular value decomposition of the matrix representing the crisp values of the output from a fuzzy system [6, 41]. As a result of this decomposition, the number of linguistic values for the inputs to the system is reduced. Although these methods can be quite effective in reducing the number of rules in a fuzzy system, they are applicable mainly for systems with two inputs. In the case of more inputs, the singular value decomposition process becomes quite complex as the dimension of the space in which the associated matrix is defined increases significantly.

The methods in group 4 are based on conversion of the intersection rule configuration of a fuzzy system into a union rule configuration with a smaller number of rules [14, 36]. These methods can be quite effective in reducing the number of rules in a fuzzy system but they can only be applied to a special class of problems called 'additively separable'. For problems that do not belong to this class, the conversion of the intersection rule configuration into a union rule configuration is not possible.

The methods in group 5 convert a fuzzy system into spatially decomposed subsystems as a result of which the overall number of rules is reduced [3, 4, 7, 8, 32, 33]. In this case, the interactions among the subsystems are partially compensated and the resulting decomposed system has a decoupled structure. Although these methods have been widely used recently, the success of their application depends on the strength of interactions among the subsystems and the level of their compensation.

The methods in group 6 rearrange the inputs in a fuzzy system in a way that leads to the reduction of the

number of rules [11, 19, 23–25, 30]. In this case, the fuzzy system is decomposed into a multilayer hierarchical structure such that each layer has only two inputs and one output. Although these methods have become quite popular recently, they do not offer clear interpretation of the intermediate variables between the first and the last layer. Besides this, only two inputs are taken into account in each layer while all other inputs are ignored.

The methods in group 7 use the H-infinity approach to capture the uncertainty aspects of fuzzy systems [9, 15–17, 37, 38]. In this case, the filtration of fuzzy rules is based on the use of linear matrix inequalities and convex optimization techniques. Although these methods have attracted significant attention recently, they are only applicable to Takagi-Sugeno fuzzy systems.

The methods in group 8 use the genetic algorithm approach to capture the non-linearity aspects of fuzzy systems [31, 39]. In this case, the optimisation of fuzzy rules is based on the use of self-learning algorithms and parallel computation techniques. Although these methods have undergone significant development recently, they are only applicable to decision making problems.

Most of the above rule base reduction methods for fuzzy systems have serious drawbacks such as empirical nature, limited scope and approximate solutions. The empirical nature of the methods in groups 1–2 and 5–6 assumes the use of a 'trial and error' approach that can be unreliable. Besides this, the limited scope of the methods in groups 3–4 and 7–8 makes them inappropriate for a wide range of fuzzy systems. And finally, the approximate solutions obtained by most of these methods compromise to some extent their performance.

This paper addresses the above drawbacks of existing rule base reduction methods by proposing a novel rule base compression method for Mamdani fuzzy systems with non-monotonic rules. The method is characterised by systematic nature, universal scope and precise solutions. Besides this, the method uses a precise indicator for the quantitative complexity of fuzzy systems in terms of the amount of on-line operations as opposed to the rough indicator used by existing methods in terms of the number of rules.

The remaining part of this paper is structured as follows. Section 2 provides some theoretical preliminaries for fuzzy systems. Section 3 introduces the rule base compression method. Section 4 illustrates the application of this method for an aircraft landing control case study. Section 5 evaluates the performance of the method in a quantitative and comparative context. Section 6 summarises the main advantages of the method and highlights future research directions.



equal to 1 and are commonly used in fuzzy systems due to their simplicity.

In this case, the fuzzy membership degree  $f_{ps}$  for an input is derived by

$$\begin{aligned}
 f_{ps} &= 0, \text{ if } x_{ps} \leq a_{ps} \\
 f_{ps} &= (x_{ps} - a_{ps}) / (b_{ps} - a_{ps}), \text{ if } a_{ps} \leq x_{ps} \leq b_{ps} \\
 f_{ps} &= (c_{ps} - x_{ps}) / (c_{ps} - b_{ps}), \text{ if } b_{ps} \leq x_{ps} \leq c_{ps} \\
 f_{ps} &= 0, \text{ if } c_{ps} \leq x_{ps}
 \end{aligned} \tag{4}$$

where  $x_{ps}$ ,  $p = 1, \dots, m$ ,  $s = 1, \dots, r$  is the continuous crisp value of the  $p$ -th input in the  $s$ -th rule of the fuzzy system and  $a_{ps}$ ,  $b_{ps}$ ,  $c_{ps}$  are the parameters of the triangular fuzzy membership function used for fuzzification of this input. In particular,  $a_{ps}$  is the point at which the membership function becomes greater than 0,  $b_{ps}$  is the point at which the membership function reaches its maximum at 1 and  $c_{ps}$  is the point at which the membership function becomes equal to 0 again. The symbol ‘/’ denotes arithmetic division in Equation (4) and all subsequent equations.

The application substage in a fuzzy system maps the fuzzy membership degrees of the inputs in each rule to a firing strength for this rule. The considerations presented here are based on rule bases with conjunctive terms in the ‘if’ part. Such rule bases are commonly used in fuzzy systems due to their ability to represent the simultaneous effect of all inputs.

In this case, the firing strength  $g_s$  for a rule is derived by

$$\begin{aligned}
 g_1 &= \min (f_{11}, \dots, f_{m1}) \\
 \dots &\dots \dots \dots \dots \dots \\
 g_r &= \min (f_{1r}, \dots, f_{mr})
 \end{aligned} \tag{5}$$

where  $f_{ps}$ ,  $p = 1, \dots, m$ ,  $s = 1, \dots, r$  is the fuzzy membership degree for the  $p$ -th input in the  $s$ -th rule of the fuzzy system.

The implication substage in a fuzzy system maps the firing strength for each rule to a fuzzy membership function for the output in this rule. The considerations presented here are based on horizontal truncation that cuts the normal fuzzy triangular membership function for the output in each rule to a subnormal fuzzy trapezoidal membership function whose maximum is equal to the firing strength for this rule. This type of truncation is commonly used in fuzzy systems due to its simplicity.

In this case, the fuzzy membership function  $F_{sq}$  for an output is defined by

$$F_{sq} = \{f_{1sq} * y_{1sq}, \dots, f_{tsq} * y_{tsq}\} \tag{6}$$

where  $f_{ksq}$ ,  $k = 1, \dots, t$ ,  $s = 1, \dots, r$ ,  $q = 1, \dots, n$  is the fuzzy membership degree for the  $k$ -th element from a discrete variation range for the  $q$ -th output in the  $s$ -th rule of the fuzzy system,  $y_{ksq}$  is the associated element from this range and  $t$  is the number of such elements. The symbol ‘\*’ in Equation (6) denotes binary association, i.e. the fuzzy membership degree  $f_{ksq}$  is associated with the element  $y_{ksq}$  from the discrete variation range for this output.

As the subscript  $k$  for  $f_{ksq}$  and  $y_{ksq}$  in Equation (6) is not required further, this subscript will be omitted for simplicity. Therefore, the element  $y_{sq}$  is mapped to its fuzzy membership degrees  $f_{sq}$  by

$$\begin{aligned}
 f_{sq} &= 0, \text{ if } y_{sq} \leq a_{sq} \\
 f_{sq} &= (y_{sq} - a_{sq}) / (b_{sq} - a_{sq}), \text{ if } a_{sq} \leq y_{sq} \leq b_{sq} \\
 f_{sq} &= g_s, \text{ if } b_{sq} \leq y_{sq} \leq c_{sq} \\
 f_{sq} &= (d_{sq} - y_{sq}) / (c_{sq} - b_{sq}), \text{ if } c_{sq} \leq y_{sq} \leq d_{sq} \\
 f_{sq} &= 0, \text{ if } d_{sq} \leq y_{sq}
 \end{aligned} \tag{7}$$

where  $y_{sq}$ ,  $s = 1, \dots, r$ ,  $q = 1, \dots, n$  is the discrete crisp value of the  $q$ -th output in the  $s$ -th rule of the fuzzy system and  $a_{sq}$ ,  $b_{sq}$ ,  $c_{sq}$ ,  $d_{sq}$  are the parameters of the trapezoidal fuzzy membership function for this output. This function is obtained during the implication substage from the initial triangular fuzzy membership function for the output. In particular,  $a_{sq}$  is the point at which the membership function becomes greater than 0,  $b_{sq}$  is the point at which the membership function becomes equal to its maximum  $g_s$ ,  $c_{sq}$  is the point at which the membership function becomes less than its maximum at  $g_s$  and  $d_{sq}$  is the point at which the membership function becomes equal to 0 again.

The aggregation substage in a fuzzy system maps the fuzzy membership functions for all rules to an aggregated fuzzy membership function representing the overall output for the rules. The considerations presented here are based on disjunctive rule bases. Such rule bases are commonly used in fuzzy systems due to their ability to represent the effect from the most dominant rule.

In this case, the aggregated fuzzy membership function  $F_q$  for an output is derived by

$$F_q = F_{1q} \text{ or } \dots \text{ or } F_{rq} \tag{8}$$

where  $F_{sq}$ ,  $s = 1, \dots, r$ ,  $q = 1, \dots, n$  is the fuzzy membership function for the  $q$ -th output in the  $s$ -th rule of the fuzzy system. The symbol ‘or’ denotes a union operation that is applied to the fuzzy membership functions for the output in all rules. This operation is applied to the fuzzy membership degrees for all the elements from the discrete variation range for this output.

The defuzzification stage in a fuzzy system maps the aggregated fuzzy membership function for an output to a crisp value from the discrete variation range for this output. As this value is of a continuous type, the associated discrete variation range is mapped to its continuous counterpart. The considerations presented assume that the defuzzified value of the output is the centre of gravity for the aggregated fuzzy membership function for this output. This defuzzification method is commonly used in fuzzy systems due to its applicability for any shape of aggregated fuzzy membership function for the output.

In this case, the defuzzified value  $D_q$  for an output is derived by

$$D_q = (f_{1q} \cdot y_{1q} + \dots + f_{iq} \cdot y_{iq}) / (f_{1q} + \dots + f_{iq}) \tag{9}$$

where  $f_{kq}$ ,  $k = 1, \dots, t$ ,  $q = 1, \dots, n$  is the aggregated fuzzy membership degree for the  $k$ -th element from the discrete variation range for the  $q$ -th output of the fuzzy system and  $y_{kq}$  is the associated element from this range. Equation (9) represents  $f_{ksq}$  and  $y_{ksq}$  from Equation (6) without the rule index  $s$  as the defuzzification stage is independent of the rules. Obviously,  $D_q$  can take any values within the continuous counterpart for the discrete variation range for this output. The symbols ‘ $\cdot$ ’ and ‘ $+$ ’ in Equation (9) denote arithmetic multiplication and addition, respectively.

### 3. Rule base compression method

The method introduced here removes dynamically the redundancy in a fuzzy system during the fuzzification, inference and defuzzification stages for each simulation cycle. The redundancy is expressed by the presence of non-monotonic rules and it is removed by filtering the redundant subset of these rules with the aim of making the rule base monotonic. The method is applicable to Mamdani fuzzy systems with non-monotonic rule bases.

Filtration of non-monotonic rules in a fuzzy system is equivalent to representing a ‘many-to-one’ mapping as a ‘one-to-one’ mapping. A mathematical theorem

for this representation is shown below. The proof of the theorem is based on Boolean logic laws and it is also shown further below.

**Theorem 1.** *A set of non-monotonic disjunctive rules in the form*

$$\text{If } (A_{11} \text{ and } \dots \text{ and } A_{m1}) \text{ then } C_q \tag{10}$$

$$\dots \dots \dots \text{If } (A_{1z} \text{ and } \dots \text{ and } A_{mz}) \text{ then } C_q$$

where  $A_{pj} = (i_p \text{ is } v_{ip,j})$ ,  $p = 1, \dots, m$ ,  $j = 1, \dots, z$  and  $C_q = (o_q \text{ is } v_{oq})$ ,  $q = 1, \dots, n$  are logical propositions describing the terms for the  $p$ -th input in the  $j$ -th rule and the terms for the  $q$ -th output in accordance with Equation (1),  $q$  is a set label and  $z$  is the set cardinality, can be represented as a single rule in the form

$$\text{If } [(A_{11} \text{ and } \dots \text{ and } A_{m1})] \text{ and } \dots \text{ and } (A_{1z} \text{ and } \dots \text{ and } A_{mz}) \text{ then } C_q \tag{11}$$

**Proof 1.** Equation (10) represents a set of ‘if-then’ implications that can be rewritten as

$$(A_{11} \text{ and } \dots \text{ and } A_{m1}) \text{ imp } C_q \tag{12}$$

$$\dots \dots \dots (A_{1z} \text{ and } \dots \text{ and } A_{mz}) \text{ imp } C_q$$

where the ‘if-then’ notations are replaced by ‘implication’ operators.

The implications in Equation (12) are also disjunctive rules that can be rewritten as

$$[(A_{11} \text{ and } \dots \text{ and } A_{m1}) \text{ imp } C_q] \text{ or } \dots \text{ or } [(A_{1z} \text{ and } \dots \text{ and } A_{mz}) \text{ imp } C_q] \tag{13}$$

where all rules are disjuncted together in one rule.

Using implication related laws, Equation (13) can be rewritten as

$$[\text{not } (A_{11} \text{ and } \dots \text{ and } A_{m1}) \text{ or } C_q] \text{ or } \dots \text{ or } [\text{not } (A_{1z} \text{ and } \dots \text{ and } A_{mz}) \text{ or } C_q] \tag{14}$$

where the ‘implication’ operators are replaced by ‘negation’ and ‘disjunction’ operators.

Using commutative laws, Equation (14) can be rewritten as

$$\{[\text{not } (A_{11} \text{ and } \dots \text{ and } A_{m1})] \text{ or } \dots \text{ or } [\text{not } (A_{1z} \text{ and } \dots \text{ and } A_{mz})]\} \text{ or } (C_q \text{ or } \dots \text{ or } C_q) \tag{15}$$

where the terms for the inputs are grouped separately from the terms for the output.

Using idempotent laws, Equation (15) can be rewritten as

$$\begin{aligned} & \{[not (A_{11} \text{ and } \dots \text{ and } A_{m1})] \text{ or } \dots \\ & \text{or } [not (A_{1z} \text{ and } \dots \text{ and } A_{mz})] \} \text{ or } C_q \end{aligned} \quad (16)$$

where only one of the  $z$  identical terms for the output is preserved.

Using De Morgan laws, Equation (16) can be rewritten as

$$\begin{aligned} & not [(A_{11} \text{ and } \dots \text{ and } A_{m1})] \text{ and } \dots \\ & \text{and } (A_{1z} \text{ and } \dots \text{ and } A_{mz}) \} \text{ or } C_q \end{aligned} \quad (17)$$

where the  $z$  groups of terms for the inputs are conjuncted together and negated as a whole.

Using again implication related laws, Equation (17) can be rewritten as

$$\begin{aligned} & [(A_{11} \text{ and } \dots \text{ and } A_{m1})] \text{ and } \dots \\ & \text{and } (A_{1z} \text{ and } \dots \text{ and } A_{mz}) \} \text{ imp } C_q \end{aligned} \quad (18)$$

where the ‘negation’ and ‘disjunction’ operator are replaced by an ‘implication’ operator.

Equation (18) represents an implication that can be rewritten as Equation (11) where the implication operator is replaced by an ‘if-then’ notation. So, this concludes the proof.

The ‘many-to-one’ mapping from Equation (10) is represented equivalently as a ‘one-to-one’ mapping from Equation (11). In this case, the  $z$  logical propositions  $(A_{11} \text{ and } \dots \text{ and } A_{m1}) \dots (A_{1z} \text{ and } \dots \text{ and } A_{mz})$  in the ‘if’ part of the non-monotonic set of rules in Equation (10) are represented by a single logical proposition  $[(A_{11} \text{ and } \dots \text{ and } A_{m1}) \text{ and } \dots \text{ and } (A_{1z} \text{ and } \dots \text{ and } A_{mz})]$  in the ‘if’ part of a single equivalent rule in Equation (11).

Theorem 1 can be trivially extended to an arbitrary number of sets of non-monotonic rules where each of these sets can be represented by a separate single equivalent rule. In this way, the non-monotonic rule base of a fuzzy system can be converted to an equivalent monotonic rule base of a smaller size that contains only the single equivalent rules.

Theorem 1 describes the theoretical foundations of the rule base compression method. The practical implementation of this method is given by the algorithm below.

Algorithm 1:

1. Put all non-monotonic rules in separate groups sorted in an increasing order with respect to the linguistic values of the output, whereby the rules in each group have the same linguistic values of the output and different permutations of linguistic values of the inputs.
2. For each group of rules, find a single equivalent rule whose effect on the defuzzified output is the same as the effect of all rules in this group.
3. For each group of rules, keep the single equivalent rule and remove all other rules.

Algorithm 1 guarantees that there are only monotonic rules left in a fuzzy rule base after the completion of the filtration process. In this case, the number of monotonic rules is equal to the number of non-monotonic groups of rules and the number of different of linguistic values of the output. Therefore, the filtration process can be applied with a guaranteed success whereby the resulting compressed rule base is always monotonic.

Step 1 in Algorithm 1 can be applied off-line but steps 2–3 can only be applied on-line. This is because the single equivalent rule is a dominant rule that can be found only after the completion of the fuzzification stage and the application substage. This dominance is expressed by the rule with the maximal firing strength for each group as a result of which the effect of all other rules from the group on the defuzzified output is completely neutralised. When there is more than one such dominant rule in a group, i.e. two or more rules with maximal firing strength for the group, any of these rules can be selected arbitrarily as a single equivalent rule.

As Algorithm 1 is applied right after the completion of the fuzzification stage and the application substage for each simulation cycle, the remaining substages of implication and aggregation as well as the defuzzification stage can make use of the compressed monotonic rule base. However, for the next simulation cycle, the original non-monotonic rule base must be used again in the fuzzification stage and the application substage.

Theorem 1 and Algorithm 1 are presented above for a single-output fuzzy system but they can be trivially extended to a multiple-output fuzzy system with an arbitrary number of outputs. In this case, the multiple-output fuzzy system from Equation (1) can be represented by the following  $n$  equivalent single-output fuzzy systems

$$\begin{aligned}
 & \text{If } i_1 \text{ is } v_{i1l} \text{ and } \dots \text{ and } i_m \text{ is } v_{iml} \\
 & \quad \text{then } o_q \text{ is } v_{q1l} \\
 & \dots \dots \dots \dots \dots \dots \dots \dots \dots \\
 & \text{If } i_1 \text{ is } v_{i1r} \text{ and } \dots \text{ and } i_m \text{ is } v_{imr} \\
 & \quad \text{then } o_q \text{ is } v_{q1r} \\
 & \quad q = 1, \dots, n
 \end{aligned} \tag{19}$$

where by all considerations from the theorem and the algorithm can be applied repetitively to each of these systems.

#### 4. Simulation results

The rule base compression method is applied to a case study for aircraft landing control [34]. A fuzzy system for aircraft landing control is described by the inputs  $i_1, i_2$  and the output  $o_1$  where  $i_1$  is the *relative height* (h) of the aircraft in *feet* (ft),  $i_2$  is the *vertical velocity* (v) of the aircraft in *feet per second* (ft/s) and  $o_1$  is the *control effort* (e) in *libras* (lb) that must be applied to the aircraft. In this case,  $i_1$  can take the four linguistic values *near zero* (NZ), *small* (S), *medium* (M) and *large* (L), whereas both  $i_2$  and  $o_1$  can take the five linguistic values *down large* (DL), *down small* (DS), *zero* (Z), *up small* (US) and *up large* (UL).

By making the substitutions NZ=1, S=2, M=3, L=4 for  $i_1$  and the substitutions DL=1, DS=2, Z=3, US=4, UL=5 for both  $i_2$  and  $o_1$ , the rule base for this fuzzy system can be presented, as shown in Table 1. Then, by applying step 1 from Algorithm 1, this initial rule base can be rearranged into a sorted rule base by putting all non-monotonic rules in separate groups, as shown in Table 2. The empty rows in Tables 1–2 are used for the purpose of visual separation of rules from different groups.

Further on, by applying steps 2–3 from Algorithm 1, the sorted rule base can be transformed into a compressed rule base, as shown in Table 3. This rule base contains only the single equivalent rule from each of the five non-monotonic groups of rules from the sorted rule base. The process leading to the determination of these single equivalent rules is described below.

The following paragraphs consider in detail the fuzzification, inference and defuzzification stages for the initial rule base first and then for the compressed rule base. The aim of these considerations is to show the behavioural equivalence of the two rule bases, i.e. that the defuzzified output is the same for any crisp values

Table 1  
Initial rule base

Rule number	First input	Second input	Output
1	1	1	5
2	1	2	5
3	1	3	3
4	1	4	2
5	1	5	2
6	2	1	5
7	2	2	4
8	2	3	3
9	2	4	2
10	2	5	1
11	3	1	4
12	3	2	3
13	3	3	2
14	3	4	1
15	3	5	1
16	4	1	3
17	4	2	2
18	4	3	1
19	4	4	1
20	4	5	1

Table 2  
Sorted rule base

Rule number	Fist input	Second input	Output
10	2	5	1
14	3	4	1
15	3	5	1
18	4	3	1
19	4	4	1
20	4	5	1
4	1	4	2
5	1	5	2
9	2	4	2
13	3	3	2
17	4	2	2
3	1	3	3
8	2	3	3
12	3	2	3
16	4	1	3
7	2	2	4
11	3	1	4
1	1	1	5
2	1	2	5
6	2	1	5

of the inputs. In the current example, these values are taken as  $h=400$  and  $v=15$ .

At the fuzzification stage for the initial rule base, it is necessary to consider all possible linguistic values for each input. In this context, it should be specified how the fuzzy membership degree for a particular linguistic value of a given input can be obtained. This specification is based on the parameters of the fuzzy membership functions used for fuzzification of the inputs, as shown below.

Table 3  
Compressed rule base

Rule number	First input	Second input	Output
10	2	5	1
9	2	4	2
3	1	3	3
7	2	2	4
1	1	1	5

The fuzzy membership degree  $f_h^{NZ}$  for the linguistic value *near zero* of the input *height* can be obtained by the formula

$$\begin{aligned} f_h^{NZ} &= 0, \text{ if } h < a_h^{NZ} \\ f_h^{NZ} &= 1, \text{ if } a_h^{NZ} \leq h \leq b_h^{NZ} \\ f_h^{NZ} &= (c_h^{NZ} - h)/(c_h^{NZ} - b_h^{NZ}), \text{ if } b_h^{NZ} \leq h \leq c_h^{NZ} \\ f_h^{NZ} &= 0, \text{ if } c_h^{NZ} \leq h \end{aligned} \quad (20)$$

where  $a_h^{NZ}$ ,  $b_h^{NZ}$ ,  $c_h^{NZ}$  are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_h^{NZ} = 0, b_h^{NZ} = 0, c_h^{NZ} = 500 \quad (21)$$

Equation (21) differs slightly from Equation (4). In particular, the 'equality' part of the sign in the first line of Equation (4) is removed from Equation (21) to account for the vertical left shoulder of the membership function. Also, the arithmetic division in the second line of Equation (4) is removed from Equation (21) to avoid division by zero as a result of the equality of the parameters  $a_h^{NZ}$  and  $b_h^{NZ}$ .

The fuzzy membership degree  $f_h^S$  for the linguistic value *small* of the input *height* can be obtained by the formula

$$\begin{aligned} f_h^S &= 0, \text{ if } h \leq a_h^S \\ f_h^S &= (h - a_h^S)/(b_h^S - a_h^S), \text{ if } a_h^S \leq h \leq b_h^S \\ f_h^S &= (c_h^S - h)/(c_h^S - b_h^S), \text{ if } b_h^S \leq h \leq c_h^S \\ f_h^S &= 0, \text{ if } c_h^S \leq h \end{aligned} \quad (22)$$

where  $a_h^S$ ,  $b_h^S$ ,  $c_h^S$  are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_h^S = -200, b_h^S = 300, c_h^S = 800 \quad (23)$$

The fuzzy membership degree  $f_h^M$  for the linguistic value *medium* of the input *height* can be obtained by the formula

$$\begin{aligned} f_h^M &= 0, \text{ if } h \leq a_h^M \\ f_h^M &= (h - a_h^M)/(b_h^M - a_h^M), \text{ if } a_h^M \leq h \leq b_h^M \\ f_h^M &= (c_h^M - h)/(c_h^M - b_h^M), \text{ if } b_h^M \leq h \leq c_h^M \\ f_h^M &= 0, \text{ if } c_h^M \leq h \end{aligned} \quad (24)$$

where  $a_h^M$ ,  $b_h^M$ ,  $c_h^M$  are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_h^M = 300, b_h^M = 800, c_h^M = 1300 \quad (25)$$

The fuzzy membership degree  $f_h^L$  for the linguistic value *large* of the input *height* can be obtained by the formula

$$\begin{aligned} f_h^L &= 0, \text{ if } h \leq a_h^L \\ f_h^L &= (h - a_h^L)/(b_h^L - a_h^L), \text{ if } a_h^L \leq h \leq b_h^L \\ f_h^L &= 1, \text{ if } b_h^L \leq h \leq c_h^L \\ f_h^L &= 0, \text{ if } c_h^L \leq h \end{aligned} \quad (26)$$

where  $a_h^L$ ,  $b_h^L$ ,  $c_h^L$  are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_h^L = 500, b_h^L = 1000, c_h^L = 1000 \quad (27)$$

Equation (26) differs slightly from Equation (4). In particular, the arithmetic division in the third line of Equation (4) is removed from Equation (26) to avoid division by zero as a result of the equality of the parameters  $b_h^L$  and  $c_h^L$ . Also, the 'equality' part of the sign in the fourth line of Equation (4) is removed from Equation (25) to account for the vertical right shoulder of the membership function.

The parameters of the fuzzy membership functions for the first input to the fuzzy system are summarised in Table 4.

The fuzzy membership degree  $f_v^{DL}$  for the linguistic value *down large* of the input *velocity* can be obtained by the formula

Table 4

Fuzzy membership function parameters for first input

Linguistic value/Input	Relative height
Near zero	[0 0 500]
Small	[-200 300 800]
Medium	[300 800 1300]
Large	[500 1000 1000]



$$\begin{aligned}
f_v^{DL} &= 0, \text{ if } v \leq a_v^{DL} \\
f_v^{DL} &= 1, \text{ if } a_v^{DL} \leq v \leq b_v^{DL} \\
f_v^{DL} &= 1, \text{ if } b_v^{DL} \leq v \leq c_v^{DL} \\
f_v^{DL} &= (d_v^{DL} - v)/(d_v^{DL} - c_v^{DL}), \text{ if } c_v^{DL} \leq v \leq d_v^{DL} \\
f_v^{DL} &= 0, \text{ if } d_v^{DL} \leq v
\end{aligned} \quad (28)$$

where  $a_v^{DL}$ ,  $b_v^{DL}$ ,  $c_v^{DL}$ ,  $d_v^{DL}$  are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_v^{DL} = -30, b_v^{DL} = -30, c_v^{DL} = -20, d_v^{DL} = -10 \quad (29)$$

Equation (28) differs slightly from Equation (7). In particular, the ‘equality’ part of the sign in the first line of Equation (7) is removed from Equation (28) to account for the vertical left shoulder of the membership function. Also, the arithmetic division in the second line of Equation (7) is removed from Equation (28) to avoid division by zero as a result of the equality of the parameters  $a_v^{DL}$  and  $b_v^{DL}$ .

The fuzzy membership degree  $f_v^{DS}$  for the linguistic value *down small* of the input *velocity* can be obtained by the formula

$$\begin{aligned}
f_v^{DS} &= 0, \text{ if } v \leq a_v^{DS} \\
f_v^{DS} &= (v - a_v^{DL})/(b_v^{DS} - a_v^{DS}), \text{ if } a_v^{DS} \leq v \leq b_v^{DS} \\
f_v^{DS} &= (c_v^{DL} - v)/(c_v^{DS} - b_v^{DS}), \text{ if } b_v^{DS} \leq v \leq c_v^{DS} \\
f_v^{DS} &= 0, \text{ if } c_v^{DS} \leq v
\end{aligned} \quad (30)$$

where  $a_v^{DS}$ ,  $b_v^{DS}$ ,  $c_v^{DS}$  are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_v^{DS} = -20, b_v^{DS} = -10, c_v^{DS} = 0 \quad (31)$$

The fuzzy membership degree  $f_v^Z$  for the linguistic value *zero* of the input *velocity* can be obtained by the formula

$$\begin{aligned}
f_v^Z &= 0, \text{ if } v \leq a_v^Z \\
f_v^Z &= (v - a_v^Z)/(b_v^Z - a_v^Z), \text{ if } a_v^Z \leq v \leq b_v^Z \\
f_v^Z &= (c_v^Z - v)/(c_v^Z - b_v^Z), \text{ if } b_v^Z \leq v \leq c_v^Z \\
f_v^Z &= 0, \text{ if } c_v^Z \leq v
\end{aligned} \quad (32)$$

where  $a_v^Z$ ,  $b_v^Z$ ,  $c_v^Z$  are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_v^Z = -10, b_v^Z = 0, c_v^Z = 10 \quad (33)$$

The fuzzy membership degree  $f_v^{US}$  for the linguistic value *up small* of the input *velocity* can be obtained by the formula

$$\begin{aligned}
f_v^{US} &= 0, \text{ if } v \leq a_v^{US} \\
f_v^{US} &= (v - a_v^{US})/(b_v^{US} - a_v^{US}), \text{ if } a_v^{US} \leq v \leq b_v^{US} \\
f_v^{US} &= (c_v^{US} - v)/(c_v^{US} - b_v^{US}), \text{ if } b_v^{US} \leq v \leq c_v^{US} \\
f_v^{US} &= 0, \text{ if } c_v^{US} \leq v
\end{aligned} \quad (34)$$

where  $a_v^{US}$ ,  $b_v^{US}$ ,  $c_v^{US}$  are the parameters of the associated triangular fuzzy membership function whose values are given by

$$a_v^{US} = 0, b_v^{US} = 10, c_v^{US} = 20 \quad (35)$$

The fuzzy membership degree  $f_v^{UL}$  for the linguistic value *up large* of the input *velocity* can be obtained by the formula

$$\begin{aligned}
f_v^{UL} &= 0, \text{ if } v \leq a_v^{UL} \\
f_v^{UL} &= (v - a_v^{UL})/(b_v^{UL} - a_v^{UL}), \text{ if } a_v^{UL} \leq v \leq b_v^{UL} \\
f_v^{UL} &= 1, \text{ if } b_v^{UL} \leq v \leq c_v^{UL} \\
f_v^{UL} &= 1, \text{ if } c_v^{UL} \leq v \leq d_v^{UL} \\
f_v^{UL} &= 0, \text{ if } d_v^{UL} < v
\end{aligned} \quad (36)$$

where  $a_v^{UL}$ ,  $b_v^{UL}$ ,  $c_v^{UL}$ ,  $d_v^{UL}$  are the parameters of the associated trapezoidal fuzzy membership function whose values are given by

$$a_v^{UL} = 10, b_v^{UL} = 20, c_v^{UL} = 30, d_v^{UL} = 30 \quad (37)$$

Equation (36) differs slightly from Equation (7). In particular, the arithmetic division in the fourth line of Equation (7) is removed from Equation (36) to avoid division by zero as a result of the equality of the parameters  $c_v^{UL}$  and  $d_v^{UL}$ . Also, the ‘equality’ part of the sign in the fifth line of Equation (7) is removed from Equation (28) to account for the vertical right shoulder of the membership function.

The parameters of the fuzzy membership functions for the second input to the fuzzy system are summarised in Table 5.

At the application substage of the inference stage for the initial rule base, it is necessary to find the firing strength for each rule. For this purpose, the fuzzy membership degrees for the inputs obtained during the fuzzification stage are processed, as shown by Equations (38–57).

Table 5

Fuzzy membership function parameters for second input

Linguistic value/Input	Vertical velocity
Down large	[-30 -30 -20 -10]
Down small	[-20 -10 0]
Zero	[-10 0 10]
Up small	[0 10 20]
Up large	[10 20 30 30]

$$g_1^{UL} = \min(f_h^{NZ}, f_v^{DL}) = \min(0.2, 0) = 0 \quad (38)$$

$$g_2^{UL} = \min(f_h^{NZ}, f_v^{DS}) = \min(0.2, 0) = 0 \quad (39)$$

$$g_3^Z = \min(f_h^{NZ}, f_v^Z) = \min(0.2, 0) = 0 \quad (40)$$

$$g_4^{DS} = \min(f_h^{NZ}, f_v^{US}) = \min(0.2, 0.5) = 0.2 \quad (41)$$

$$g_5^{DS} = \min(f_h^{NZ}, f_v^{UL}) = \min(0.2, 0.5) = 0.2 \quad (42)$$

$$g_6^{UL} = \min(f_h^S, f_v^{DL}) = \min(0.8, 0) = 0 \quad (43)$$

$$g_7^{US} = \min(f_h^S, f_v^{DS}) = \min(0.8, 0) = 0 \quad (44)$$

$$g_8^Z = \min(f_h^S, f_v^Z) = \min(0.8, 0) = 0 \quad (45)$$

$$g_9^Z = \min(f_h^S, f_v^{US}) = \min(0.8, 0.5) = 0.5 \quad (46)$$

$$g_{10}^{DL} = \min(f_h^S, f_v^{UL}) = \min(0.8, 0.5) = 0.5 \quad (47)$$

$$g_{11}^{US} = \min(f_h^M, f_v^{DL}) = \min(0.2, 0) = 0 \quad (48)$$

$$g_{12}^Z = \min(f_h^M, f_v^{DS}) = \min(0.2, 0) = 0 \quad (49)$$

$$g_{13}^{DS} = \min(f_h^M, f_v^Z) = \min(0.2, 0) = 0 \quad (50)$$

$$g_{14}^{DL} = \min(f_h^M, f_v^{US}) = \min(0.2, 0.5) = 0.2 \quad (51)$$

$$g_{15}^{DL} = \min(f_h^M, f_v^{UL}) = \min(0.2, 0.5) = 0.2 \quad (52)$$

$$g_{16}^Z = \min(f_h^L, f_v^{DL}) = \min(0, 0) = 0 \quad (53)$$

$$g_{17}^{DS} = \min(f_h^L, f_v^{DS}) = \min(0, 0) = 0 \quad (54)$$

$$g_{18}^Z = \min(f_h^L, f_v^Z) = \min(0, 0) = 0 \quad (55)$$

$$g_{19}^{DL} = \min(f_h^L, f_v^{US}) = \min(0, 0.5) = 0 \quad (56)$$

$$g_{20}^{DL} = \min(f_h^L, f_v^{UL}) = \min(0, 0.5) = 0 \quad (57)$$

After the application substage of the inference stage, it is necessary to put all the rules from the initial rule base in non-monotonic groups in accordance with step 1 of Algorithm 1.

Then, it is necessary to identify the single equivalent rule for each group and remove all other rules from the initial rule base in accordance with steps 2–3 of Algorithm 1, as shown below.

The firing strength for the rules in each group is given by Equations (58–62).

$$g_{10}^{DL} = 0.5, g_{14}^{DL} = 0.2, g_{15}^{DL} = 0.2, \\ g_{18}^{DL} = 0, g_{19}^{DL} = 0, g_{20}^{DL} = 0 \quad (58)$$

$$g_4^{DS} = 0.2, g_5^{DS} = 0.2, g_9^{DS} = 0.5, \\ g_{13}^{DS} = 0, g_{17}^{DS} = 0 \quad (59)$$

$$g_3^Z = 0, g_8^Z = 0, g_{12}^Z = 0, g_{16}^Z = 0 \quad (60)$$

$$g_7^{US} = 0, g_{11}^{US} = 0 \quad (61)$$

$$g_1^{UL} = 0, g_2^{UL} = 0, g_6^{UL} = 0 \quad (62)$$

The single equivalent rules for the above groups are given by Equations (63–67).

$$\text{Group DL} : g_{10}^{DL} = 0.5 \quad (63)$$

$$\text{Group DS} : g_9^{DS} = 0.5 \quad (64)$$

$$\text{Group Z} : g_3^Z = 0 \quad (65)$$

$$\text{Group US} : g_7^{US} = 0 \quad (66)$$

$$\text{Group UL} : g_1^{UL} = 0 \quad (67)$$

Therefore, the compressed rule base will contain only the five single equivalent rules presented above. In this case, the implication and aggregation substages of the inference stage will be quite simple, as shown by Equations (68–72) and Equation (73), respectively. These substages take into account the parameters of the fuzzy membership functions for the output from the fuzzy system that are summarised in Table 6.

$$F_{10} = \{0.5/ -30, 0.5/ -25, 0.5/ -20, 0.5/ -15, \\ 0/ -10, 0/ -5, 0/0, 0/5, 0/10, 0/15, \\ 0/20, 0/25, 0/30\} \quad (68)$$

$$F_9 = \{0/ -30, 0/ -25, 0/ -20, 0.5/ -15, \\ 0.5/ -10, 0.5/ -5, 0/0, 0/5, 0/10, 0/15, \\ 0/20, 0/25, 0/30\} \quad (69)$$

Table 6

Fuzzy membership function parameters for output

Linguistic value/Output	Control effort
Down large	[-30 -30 -20 -10]
Down small	[-20 -10 0]
Zero	[-10 0 10]
Up small	[0 10 20]
Up large	[10 20 30 30]

$$F_3 = \{0/ - 30, 0/ - 25, 0/ - 20, 0/ - 15, 0/ - 10, 0/ - 5, 0/0, 0/5, 0/10, 0/15, 0/20, 0/25, 0/30 \} \tag{70}$$

$$F_7 = \{0/ - 30, 0/ - 25, 0/ - 20, 0/ - 15, 0/ - 10, 0/ - 5, 0/0, 0/5, 0/10, 0/15, 0/20, 0/25, 0/30 \} \tag{71}$$

$$F_1 = \{0/ - 30, 0/ - 25, 0/ - 20, 0/ - 15, 0/ - 10, 0/ - 5, 0/0, 0/5, 0/10, 0/15, 0/20, 0/25, 0/30 \} \tag{72}$$

$$F = F_{10} \text{ or } F_9 \text{ or } F_3 \text{ or } F_7 \text{ or } F_1 = \{0.5/ - 30, 0.5/ - 25, 0.5/ - 20, 0.5/ - 15, 0.5/ - 10, 0.5/ - 5, 0/0, 0/5, 0/10, 0/15, 0/20, 0/25, 0/30 \} \tag{73}$$

Table 7  
Numerical values of 4 × 5 point output surfaces

Point number/ Component	First input	Second input	CS output	FS output
1	0.0	-30.0	22.4	22.4
2	0.0	-15.0	16.9	16.9
3	0.0	0.0	0.0	0.0
4	0.0	15.0	-15.6	-15.6
5	0.0	30.0	-14.7	-14.7
6	333.3	-30.0	21.5	21.5
7	333.3	-15.0	15.4	15.4
8	333.3	0.0	-0.9	-0.9
9	333.3	15.0	-16.4	-16.4
10	333.3	30.0	-19.4	-19.4
11	666.7	-30.0	10.0	10.0
12	666.7	-15.0	5.0	5.0
13	666.7	0.0	-11.3	-11.3
14	666.7	15.0	-17.9	-17.9
15	666.7	30.0	-21.8	-21.8
16	1000.0	-30.0	4.5	4.5
17	000.0	-15.0	0.0	0.0
18	1000.0	0.0	-18.3	-18.3
19	1000.0	15.0	-21.3	-21.3
20	1000.0	30.0	-22.4	-22.4

The aggregated fuzzy membership function  $F$  for the output of the compressed rule base is the same as the one for the output of the initial rule base. This is because the single equivalent rules for each non-monotonic group override the effect of the other rules during the implication and aggregation substages of the inference stage. Therefore, the five rules of the compressed rule base will lead to the same defuzzified output as the twenty rules of the initial rule base.

Like the fuzzification stage and the application substage of the inference stage, the defuzzification stage for the compressed rule base is the same as the one for the initial rule base. However, the efficiency gained by the removed redundant operations during the implication and aggregation substages of the inference stage significantly outweighs the complexity added by the selection process for single equivalent rules, as shown further below. For terminological clarity, the initial rule base and the compressed rule base will be associated with a conventional fuzzy system and a filtered fuzzy system, respectively.

The behavioural equivalence of the conventional fuzzy system and the filtered fuzzy system is illustrated analytically in Table 7. This table shows the numerical crisp values of the output surfaces for the two systems, which are identical for each permutation of crisp values of the inputs. These numerical values are derived for 4 × 5 equally spaced points within the discrete variation ranges for the inputs in analogy with the rule bases for the two systems which have 4 × 5 possible permutations of linguistic values of the inputs.

Table 8  
Dominant rules for filtered system

Point number/ Component	First input	Second input	Dominant rule numbers
1	0.0	-30.0	10, 4, 3, 7, 1
2	0.0	-15.0	10, 4, 3, 7, 2
3	0.0	0.0	10, 4, 3, 7, 1
4	0.0	15.0	10, 5, 3, 7, 1
5	0.0	30.0	10, 5, 3, 7, 1
6	333.3	-30.0	10, 4, 3, 11, 6
7	333.3	-15.0	10, 4, 12, 7, 6
8	333.3	0.0	10, 13, 8, 7, 1
9	333.3	15.0	10, 9, 3, 7, 1
10	333.3	30.0	10, 5, 3, 7, 1
11	666.7	-30.0	10, 4, 16, 11, 6
12	666.7	-15.0	10, 17, 12, 11, 6
13	666.7	0.0	18, 13, 8, 7, 1
14	666.7	15.0	15, 9, 3, 7, 1
15	666.7	30.0	15, 4, 3, 7, 1
16	1000.0	-30.0	10, 4, 16, 11, 1
17	1000.0	-15.0	10, 17, 16, 11, 1
18	1000.0	0.0	18, 13, 3, 7, 1
19	1000.0	15.0	20, 4, 3, 7, 1
20	1000.0	30.0	20, 4, 3, 7, 1

The dominant single equivalent rules for all permutations of crisp values of the inputs for the filtered fuzzy system are presented in Table 8. This table shows the corresponding rule numbers from the conventional fuzzy system in accordance with Table 1.

The behavioural equivalence of the conventional fuzzy system and the compressed fuzzy system is illustrated also graphically in Figs. 1 and 2. These figures

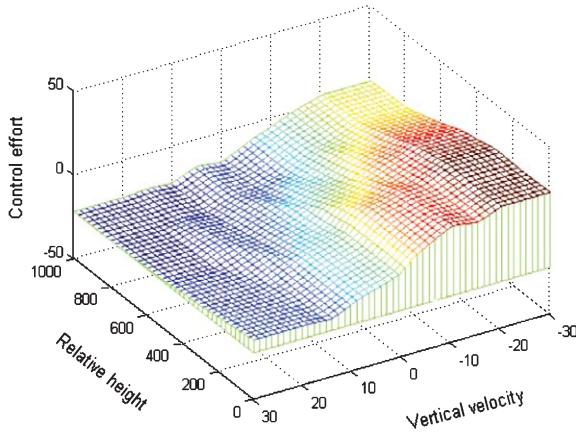


Fig. 1. Output surface with  $40 \times 50$  points for conventional system in case study.

show the output surfaces for the two systems, which are identical for each permutation of crisp values of the inputs. The underlying numerical crisp values for these output surfaces are derived for  $40 \times 50$  equally spaced points within the discrete variation ranges for the inputs. In this case, the precise numerical values and dominant rules are not shown as in Tables 7 and 8 due to the high space requirements for presenting such as a large number of points.

## 5. Performance evaluation

The proposed rule base compression method is evaluated comparatively in terms of its computational complexity, which reflects the level of quantitative complexity in general. In particular, a filtered fuzzy system that uses this rule base compression method is compared to a conventional fuzzy system that operates with all its rules and a hierarchical fuzzy system that uses the most popular rule base reduction method. The presented evaluation is two-fold – for the aircraft landing control case study and in a general context. The purpose of the evaluation is to demonstrate the efficiency of a filtered system in relation to a conventional system and a hierarchical system whereby unlike the hierarchical system whose accuracy is usually worse than the one of the conventional system, the accuracy of the filtered system is always equal to the one of the conventional system.

The parameters used for evaluating the computational complexity of the three systems are:  $m$  – number of inputs,  $w$  – number of linguistic values per input,  $n$  – number of outputs,  $t$  – number of elements in the

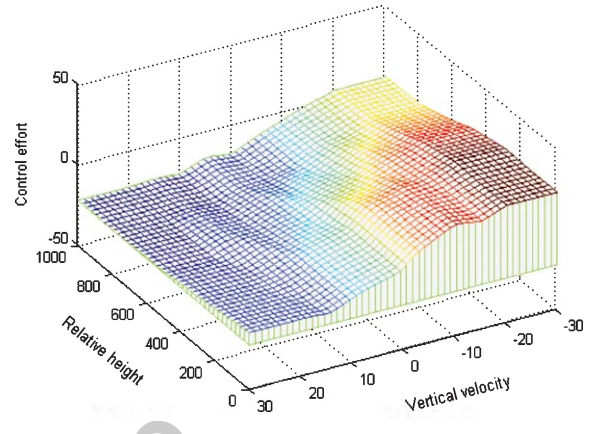


Fig. 2. Output surface with  $40 \times 50$  points for filtered system in case study.

discrete variation range for the output,  $h$  – number of simulation cycles. For the case study where the number of linguistic values per inputs varies, the associated complexity evaluation formulas are modified accordingly to reflect this.

The amount of elementary operations (EO) for the separate stages and substages in a fuzzy system is determined by the overall number of arithmetic and logical operations such as addition, subtraction, multiplication, division and comparison. For simplicity, it is assumed that each of these operations is equal to one computational time unit. Therefore, each stage and substage in the fuzzy system is quantified by means of the overall number of these units.

The next step is to find the overall number of EO for the three systems under consideration, i.e. the conventional system (CS), the hierarchical system (HS) and the filtered system (FS). This is done by summing the number of operations for the stage of fuzzification (FU), the substages of application (AP), implication (IM), aggregation (AG) of the inference stage and the stage of defuzzification (DE) for each of the three systems as well as the stage of comparison (CO) for the FS only, as shown by Equations (74–76).

$$\begin{aligned}
 EO^{CS} &= EO_{FU}^{CS} + EO_{AP}^{CS} + EO_{IM}^{CS} + EO_{AG}^{CS} + EO_{DE}^{CS} \\
 &= (6. m. w. n. h) + (w + m - 2). w^{m-1}. n. h \\
 &\quad + (7. w^{m2}. t. n. h) + (w^m - 1). t. n. h \\
 &\quad + (3. t - 1). n. h \\
 &= [(8. t + 1). w^m + (m - 2). w^{m-1} \\
 &\quad + 6. m. w + 2. t - 1]. n. h
 \end{aligned} \tag{74}$$

$$\begin{aligned}
 EO^{HS} &= EO_{FU}^{HS} + EO_{AP}^{HS} + EO_{IM}^{HS} + EO_{AG}^{HS} + EO_{DE}^{HS} \\
 &= (m - 1). (12. w. n. h) + (m - 1). w^2. n. h \\
 &\quad + (m - 1). (7. w^2. t. n. h) \\
 &\quad + (m - 1). (w^2 - 1). t. n. h \\
 &\quad + (m - 1). (3. t - 1). n. h \\
 &= (m - 1). [(8. t + 1). w^2 + 12. w \\
 &\quad + 2. t - 1]. n. h \tag{75}
 \end{aligned}$$

$$\begin{aligned}
 EO^{FS} &= EO_{FU}^{FS} + EO_{AP}^{FS} \\
 &\quad + EO_{IM}^{FS} + EO_{AG}^{FS} + EO_{DE}^{FS} + EO_{CO}^{FS} \\
 &= (6. m. w. n. h) + (W + m - 2). w^{m-1}. n. h \\
 &\quad + (7. w. t. n. h) + (w - 1). t. n. h \\
 &\quad + (3. t - 1). n. h + (w^m - w). n. h \\
 &= [2.w^m + (m - 2). w^{m-1} + (6. m - 1). w \\
 &\quad + (8. w + 2). t - 1]. n. h \tag{76}
 \end{aligned}$$

Equations (74–76) are used for comparative evaluation of the computational complexity of the three fuzzy systems in relation to the case study. As the two inputs there can take 4 and 5 linguistic values, respectively, the value of  $w$  is set equal to the average of the two numbers, i.e. 4.5. The overall number of EO for one simulation step is 2205 for the CS and the HS whereas this number is 583 for the FS. This shows that the FS is almost 4 times more efficient than the CS and the HS.

As Equations (74–76) represent general formulas, they can also be used in a wider context, i.e. outside the scope of specific case studies. By varying some of the parameters in these formulas, it would be possible to see the dependency between the computational complexity for each system and the values of its parameters.

As the increase of the parameters  $n$  and  $h$  would always lead to a similar linear increase of the computational complexity for the three systems, it would be reasonable to keep these parameters fixed. As far as the other parameters  $m$ ,  $w$  and  $t$  are concerned, it would be necessary to vary them because their increase would usually lead to a different exponential increase of the computational complexity of the three systems. Therefore, the parameters will have the following fixed values and discrete variation ranges

$$\begin{aligned}
 n &= 1; h = 1; m = 2, 3, 4; \\
 w &= 3, 5, 7, 9, 11; t = 7, 13, 19, 25, 31 \tag{77}
 \end{aligned}$$

In order to reduce the number of possible permutations of values for  $m$ ,  $w$  and  $t$ , it would be reasonable to assume that the variation of the parameters  $w$  and  $t$  is fixed by the formula

$$t = 3. w - 2 \tag{78}$$

Equations (77, 78) define a fairly wide scope for evaluating the computational complexity of fuzzy systems. In particular, most fuzzy systems are initially considered for one simulation step of one output before more simulation steps of this output or simulations of other outputs are considered. Also, fuzzy systems are usually represented with up to several inputs because the number of rules for more inputs would be almost unmanageable. In addition, the inputs and outputs of fuzzy systems are often described by an odd number of linguistic values as this provides better coverage of the associated discrete variation ranges. And finally, the number of elements in the discrete variation ranges for an output is often between 2 and 3 times greater than the number of linguistic values for this output, whose number is often close or equal to the number of linguistic values that each input can take.

Table 9 presents the results from a general comparative evaluation of the computational complexity of the CS, the HS and the FS. This evaluation is made using Equations (74–76) and in accordance with the assumptions made for the values of all relevant parameters, as shown by Equations (77, 78).

The numerical results for the HS and the FS in Table 9 are also illustrated graphically in Fig. 3 in terms of on-line operations. The CS is not shown because it is very

Table 9  
Computational complexity of three fuzzy systems

Number of rules/On-line operations	CS complexity	HS complexity	FS complexity
$3^2 = 9$	562	562	232
$3^3 = 27$	1,615	1,124	295
$3^4 = 81$	4,756	1,686	466
$5^2 = 25$	2,710	2,710	650
$5^3 = 125$	13,265	5,420	905
$5^4 = 625$	66,020	8,130	2,160
$7^2 = 49$	7,618	7,618	1,276
$7^3 = 343$	52,691	15,236	1,955
$7^4 = 2,401$	368,244	22,854	6,750
$9^2 = 81$	16,438	16,438	2,110
$9^3 = 729$	146,821	32,876	3,541
$9^4 = 6,561$	1,320,484	49,314	16,636
$11^2 = 121$	30,322	30,322	3,152
$11^3 = 1,331$	331,779	60,644	5,759
$11^4 = 14,641$	3,648,596	90,966	34,986

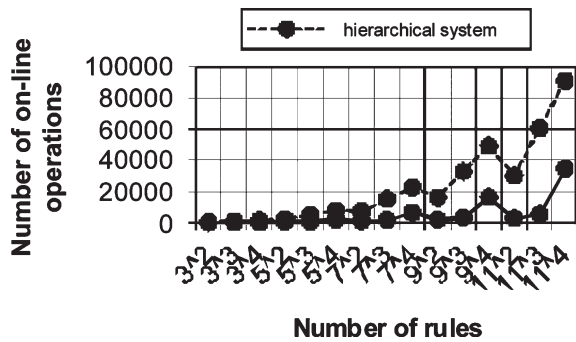


Fig. 3. Computational complexity of two fuzzy systems.

inefficient and would obstruct the graphical interpretation of the results for the HS and the FS.

Table 9 and Fig. 3 show that the FS is more efficient for all considered permutations of values for the relevant parameters. As in terms of accuracy the HS is only an approximation of the CS whereas the FS is equivalent to the CS, it is obvious that the FS outperforms significantly the CS and the HS. This superiority is valid for fuzzy systems whose number of rules is up to 14000 and above, as shown in Table 9.

## 6. Conclusion

The proposed rule base compression method for Mamdani fuzzy systems with non-monotonic rule bases reduces significantly the number of rules during the implication and aggregation substages of the inference stage. This leads to a substantial reduction in the overall amount of on-line operations which is a precise indicator of the associated computational complexity of the fuzzy system. In addition, the solution is not compromised because the defuzzified output of the filtered system is the same as the output of the conventional system. Therefore, the method is suitable for both time-critical and safety-critical applications.

The proposed method can be extended easily for other types of fuzzification, inference and defuzzification. For example, instead of triangular membership functions for fuzzification, it is possible to use trapezoidal functions or others. Also, instead of truncation type of implication, it is possible to use scaling type or others. And finally, instead of centre of gravity type of defuzzification, it is possible to use weighted average type or others. In all these cases, only small adjustments have to be made to ensure that the defuzzified value of the output for the filtered system is the same as the one for the conventional system.

The proposed method is illustrated for single simulation cycles of single-output fuzzy systems but it can be easily extended to multiple simulation cycles and multiple-output fuzzy systems. In this case, all procedures presented can be applied in exactly the same way to each simulation cycle for each output. This would lead only to a linear increase of the associated computational complexity, which would be proportional to the number of simulation cycles and the number of outputs.

The proposed method is illustrated for a two-input fuzzy system whereby each of the inputs can take up to a few linguistic values. However, it is also applicable to fuzzy systems with an arbitrary number of inputs that can take an arbitrary number of linguistic values. In this context, the method is evaluated for fuzzy systems with up to four inputs and up to eleven linguistic values per input, i.e. fuzzy systems with up to 14000 rules and above. The comparison shows the superiority of the method to two widely used other methods. In particular, the proposed method has the same accuracy as the method used in a conventional system but much better efficiency while at the same time it has fairly better accuracy and quite better efficiency than the method used in a hierarchical system.

## Acknowledgments

The first author would like to thank the Faculty of Technology at the University of Portsmouth for the granted research sabbatical that made possible the writing of this paper.

## References

- [1] A. Gegov, *Distributed Fuzzy Control of Multivariable Systems* Springer, Berlin, 1996.
- [2] A. Gegov, *Complexity Management in Fuzzy Systems: A Rule Base Compression Approach*, Springer, Berlin, 2007.
- [3] A. Gegov and M. Frank, Hierarchical fuzzy control of multivariable systems, *Fuzzy Sets and Systems* **72** (1995), 299–310.
- [4] A. Gegov and M. Frank, Decomposition of multivariable systems for distributed fuzzy control, *Fuzzy Sets and Systems* **73** (1995), 329–340.
- [5] B. Lazzerini and F. Marcelloni, Reducing computation overhead in MISO fuzzy systems, *Fuzzy Sets and Systems* **113** (2000), 485–496.
- [6] C. Tao, Comments on “reduction of fuzzy rule base via singular value decomposition”, *IEEE Transactions on Fuzzy Systems* **9**(4) (2001), 675–676.
- [7] C. Xu and Y. Lu, Decoupling in fuzzy systems: A cascade compensation approach, *Fuzzy Sets and Systems* **29** (1989), 177–185.

- [8] C. Xu, Linguistic decoupling control of fuzzy multivariable processes, *Fuzzy Sets and Systems* **44** (1991), 209–217.
- [9] F. Li, P. Shi, L. Wu and X. Zhang, Fuzzy model based stability and non-fragile control for discrete-time descriptor systems with multiple delays, *IEEE Transactions on Fuzzy Systems* **PP(99)** (2013), 1.
- [10] F. Wan, H. Shang, L. Wang and Y. Sun, How to determine the minimum number of fuzzy rules to achieve given accuracy: A computational geometric approach to SISO case, *Fuzzy Sets and Systems* **150** (2005), 199–209.
- [11] G. Raju, J. Zhou and R. Kisner, Hierarchical fuzzy control, *International Journal of Control* **54(5)** (1991), 1201–1216.
- [12] H. Roubos and M. Setnes, Compact and transparent fuzzy models and classifiers through iterative complexity reduction, *IEEE Transactions on Fuzzy Systems* **9(4)** (2001), 516–524.
- [13] J. Jang, C. Sun and E. Mizutani, *Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice Hall, Upper Saddle River, 1997.
- [14] J. Mendel and Q. Liang, Comments on “combinatorial rule explosion eliminated by a fuzzy rule configuration”, *IEEE Transactions on Fuzzy Systems* **7(3)** (1999), 369–371.
- [15] J. Qui, G. Feng and H. Gao, A new design of delay dependent robust H-infinity filtering for discrete-time T-S fuzzy systems with time-varying delay, *IEEE Transactions on Fuzzy Systems* **17(5)** (2009), 1044–1058.
- [16] J. Qui, G. Feng and H. Gao, Fuzzy model based piecewise H-infinity static output feedback controller for networked nonlinear systems, *IEEE Transactions on Fuzzy Systems* **18(5)** (2010), 919–934.
- [17] J. Qui, G. Feng and H. Gao, Non-synchronised state estimation of multichannel nonlinear systems with multiple packet dropouts, *IEEE Transactions on Fuzzy Systems* **19(1)** (2011), 75–90.
- [18] J. Yan, M. Ryan and J. Power, *Using Fuzzy Logic*, Prentice Hall, New York, 1994.
- [19] L. Wang, Analysis and design of hierarchical fuzzy systems, *IEEE Transactions on Fuzzy Systems* **7(5)** (1999), 617–624.
- [20] M. Gupta, J. Kiszka and G. Trojan, Multivariable structure of fuzzy control systems, *IEEE Transactions on Systems, Man and Cybernetics* **16(5)** (1986), 638–655.
- [21] M. Guven and K. Passino, Avoiding exponential parameter growth in fuzzy systems, *IEEE Transactions on Fuzzy Systems* **9(1)** (2001), 194–199.
- [22] M. Jamshidi, *Large Scale Systems: Modelling, Control and Fuzzy Logic*, Prentice Hall, Upper Saddle River, 1997.
- [23] M. Joo and J. Lee, Universal approximation by hierarchical fuzzy system with constraints on the fuzzy rule, *Fuzzy Sets and Systems* **130** (2002), 175–188.
- [24] M. Joo and J. Lee, A class of hierarchical fuzzy systems with constraints on the fuzzy rules, *IEEE Transactions on Fuzzy Systems* **13(2)** (2005), 194–203.
- [25] M. Lee, H. Chung and F. Yu, Modelling of hierarchical fuzzy systems, *Fuzzy Sets and Systems* **138** (2003), 343–361.
- [26] M. Negnevitsky, *Artificial Intelligence: A guide to intelligent systems*, Pearson Education, Harlow, 2002.
- [27] M. Setnes, R. Babuska and H. Verbruggen, Rule-based modelling: Precision and transparency, *IEEE Transactions on Systems, Man and Cybernetics* **28(1)** (1998), 165–169.
- [28] N. Pal, V. Eluri and G. Mandal, Fuzzy logic approaches to structure preserving dimensionality reduction, *IEEE Transactions on Fuzzy Systems* **10(3)** (2002), 277–286.
- [29] N. Xiong and L. Litz, Reduction of fuzzy control rules by means of premise learning – method and case study, *Fuzzy Sets and Systems* **132** (2002), 217–231.
- [30] O. Huwendiek and W. Brockmann, Function approximation with decomposed fuzzy systems, *Fuzzy Sets and Systems* **101** (1999), 273–286.
- [31] R. Perzina and J. Ramik, Self-learning genetic algorithm for a timetabling problem with fuzzy constraints, *International Journal of Innovative Computing, Information and Control* **9(11)** (2013), 4565–4582.
- [32] S. Chen, F. Yu and H. Chung, Decoupled fuzzy controller design with single-input fuzzy logic, *Fuzzy Sets and Systems* **129** (2002), 335–342.
- [33] S. Mollov, *Fuzzy Control of Multiple-input-multiple-output Processes*, Ph. D. Dissertation, Delft University of Technology, 2002.
- [34] T. Ross *Fuzzy Logic with Engineering Applications*, Wiley, Chichester, 2004.
- [35] V. Lacrose, *Complexity Reduction of Fuzzy Controllers: Application to Multivariable Control*, Ph. D. Dissertation, Toulouse Laboratory for Systems Analysis and Architecture, 1997.
- [36] W. Combs and J. Andrews, Combinatorial rule explosion eliminated by a fuzzy rule configuration, *IEEE Transactions on Fuzzy Systems* **6(1)** (1998), 1–11.
- [37] X. Su, L. Wu, P. Shi and Y. Song, H-infinity model reduction of Takagi-Sugeno fuzzy stochastic systems, *IEEE Transactions on Systems, Man and Cybernetics, Part B Cybernetics* **42(6)** (2012), 1574–1585.
- [38] X. Su, P. Shi, L. Wu and Y. Song, A novel approach to filter design for T-S fuzzy discrete-time systems with time-varying delay, *IEEE Transactions on Fuzzy Systems* **20(6)** (2012), 1114–1129.
- [39] Y. Huang, D. Xu and M. Tan, A parallel fuzzy learning approach to determine the hitting point for ping-point playing robot, *International Journal of Innovative Computing, Information and Control* **9(10)** (2013), 4181–4192.
- [40] Y. Kim, S. Ahn and W. Kwon, Computational complexity of general fuzzy logic control and its simplification for a loop controller, *Fuzzy Sets and Systems* **111** (2000), 215–224.
- [41] Y. Yam, P. Baranyi and C. Yang, Reduction of fuzzy rule base via singular value decomposition, *IEEE Transactions on Fuzzy Systems* **7(2)** (1999), 120–132.