

# Training, Retraining, and Self-training Procedures for the Fuzzy Logic-Based Intellectualization of IoT&S Environments

Juozas Dovydaitis<sup>1</sup> · Raimundas Jasinevicius<sup>1</sup> · Vytautas Petrauskas<sup>1</sup> · Arunas Vrubliauskas<sup>1</sup>

Received: 26 May 2014/Revised: 17 December 2014/Accepted: 25 March 2015/Published online: 24 April 2015  
© Taiwan Fuzzy Systems Association and Springer-Verlag Berlin Heidelberg 2015

**Abstract** In the past (few) years a lot of research and development projects were proposed concerning the implementation of smart home environment, sometimes even called an intelligent environment, comforting people who use various adaptations of internet of things and services. This paper advocates an opinion that a set of electronically controlled smart things must be intellectualized using human-type reasoning. A novel approach and new algorithms for the hierarchical fuzzy training, retraining, and self-training for intellectualized home environments are proposed in this paper. Training algorithms based on fuzzy logic use top-down hierarchical analysis of home situations under consideration to conquer the curse of increasing number of rules. A successful combination of crisp algorithms for the identification of presence/absence of users in the environment with fuzzy logic-based algorithms for corresponding rules subsets development and processing enables the number of necessary rules to decrease significantly. In the paper a case is presented with the starting number of 2500 rules which later diminished approximately 5 times. For the first time, the changes in users' wishes are taken into account during the retraining process. An entirely new ability of the system was

investigated, and a fuzzy logic-based algorithm for initiating a self-training process without any a priori information is developed as well. The vitality and efficiency of the proposed methodology was tested and simulated on a specialized virtual software/hardware modeling system. The proposed and simulated algorithms are delivered for use in two industrial projects.

**Keywords** Fuzzy logic · Hierarchical training · Intellectualized home environment · Internet of things and services · Retraining · Self-training · Smart home environment

## 1 Introduction: The State-of-the-Art

In spite of the fact that the idea of Internet of Things (IoT) was born many years ago after Nikola Tesla's interview in *Colliers* magazine (1926, [1]), it was stated: "When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole... and the instruments through which we shall be able to do this will be amazingly simple compared with our present telephone." We were forced to wait approximately 100 years to be able to start the full-fledged implementation of the idea into our home ambience. A number of research publications regarding the solutions of IoT problems can be found today; for example, Google presents 39 600 000 citations in the year 2008, and 130 000 000—in the year 2013. Six commonly agreed areas of human life are considered as ready for IoT implementation: Smart manufactory, Smart transport, Smart health, Smart shopping, Smart city, and Smart home/office are on the top of the list of the broad spectrum of application of different IoT-computerized gadgets, artifacts, and

---

✉ Vytautas Petrauskas  
vytautas.petrauskas@ktu.lt

Juozas Dovydaitis  
juozas.dovydaitis@ktu.lt

Raimundas Jasinevicius  
raimundas.jasinevicius@ktu.lt

Arunas Vrubliauskas  
arunas.vrubliauskas@ktu.lt

<sup>1</sup> RTCSC, KTU, K. Barsausko 59-A314, 51423 Kaunas, Lithuania

services [2], [3]. The significance of the problem under consideration is recognized on a global level: this is indicated not only by annual IoT forums under the hospice of the IEEE (The IEEE Standards Association; an Internet of Things (IoT); 5–6 November 2013 in Silicon Valley, California [4], IEEE World Forum on Internet of Things (WF-IoT); 6–8 March 2014, Seoul, Korea [5]), but also by a very serious attitude of the European Commission (EC) to the problem expressed by organizing the IERC—European Research Cluster on the IoT&S consisting of 33 research and development projects [6]. Currently, a European road map for the future research and innovations in the field of IoT and services for smart home environment is encouraged and supported in Horizon 2020 program as well [7–9]. All six areas of human activity mentioned above are intrinsically fuzzy. This is the reason why: (1) we have uncountable number of fuzzy logic-related patents (560,000 by 2014), and (2) a wide range of applications from smart manufactory control [10, 11] to hand-shake compensation and blood pressure monitoring gadgets [12, 13].

A thorough analysis of multiple sources focusing on architecture of IoT&S and environmental intelligence, like [14–16], shows that the main tendency to create a Smart Home Environment (SHE) is based on the usage of multi-agent systems. However, in this paper we advocate an approach different from [14–22] and many others; we claim that a significant difference exists between smart and intelligent things (and smart and intelligent agents as well). This new approach is widely presented in [23, 24]. According to this approach, a thing that can be electronically controlled and/or a “crisply programmed” agent, is called smart. However, we can call things and/or agents intelligent only if they are able to behave by modeling most simple features of human beings widely considered as his/her intellectual activity: (1) recognition and classification (of patterns, processes, situations), (2) behavior according to a set of fuzzy rules, and (3) functioning according to some prescribed tendency. In all these three activities, reasoning and control based on fuzzy logic play a decisive role.

## 2 SHE Intellectualized Control

The SHE itself must be considered not only as a certain set of complex sensors, actuators, means of communication, and smart things, but also as a complex sociotechnical fabric with the inherited mixture of what is really human, as well as elements of artificial intellect. Generally speaking, SHE is a fuzzy control system [25]. Its functional organization is presented in Fig. 1. Here, HE stands for the home environment functioning under the parameters of real

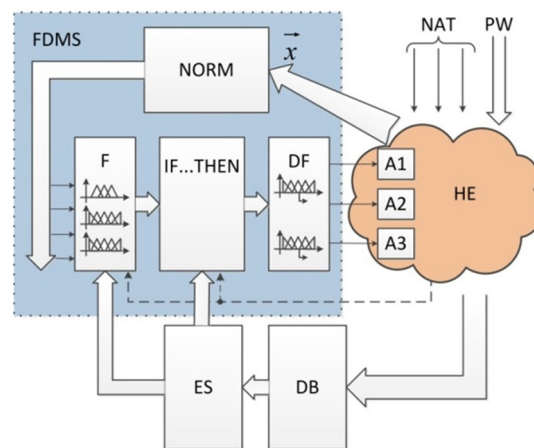


Fig. 1 SHE intellectualized control system

ambient nature (NAT) and signals that express user’s personal wishes (PW). Fuzzy decision-making system (FDMS) must perform an intellectualized HE control. Its intellectual activity is based on a set of fuzzy rules that are created during the initial training, retraining (if user changes his/her wishes), and self-training (if there is no a priori information) processes that the system undergoes. The normalized (NORM) description  $\vec{x}$  of a concrete situation in the HE is fuzzified (F), processed according to a set of IF... THEN rules and converted by the defuzzifier (DF) into reasonable actions (A1–A3). By normalizing here must be understood procedures serving for the transformation of real physical data into form acceptable for computerized processing. All events and situation changes taking place in a home environment are collected and stored in a specialized data base (DB) which is further analyzed and evaluated by special expert system and/or by certain experts (ES). The results of this analysis serve as data for creating reasonable terms for verbalization of environmental parameters and for producing a proper list of IF–THEN type inference rules as well. The dotted line emphasizes an automatic online possibility (if it is necessary) to influence or even change the shapes of F terms and IF–THEN rules.

This research is focused on algorithmic technologies for the intellectualization of a smart home environment based on fuzzy logic and fuzzy rules processing.

A novel approach to fuzzy training, retraining, and self-training, as well as new algorithms for intellectualized home environment are proposed. All three fuzzy logic-based training algorithms take top-down hierarchical analysis of home situations under consideration to conquer the curse of the increasing number of rules. The paper is organized as follows: Sect. 2 presents the general description of intellectualization of the Smart Home Environment; Sects. 3–5 describe fuzzy algorithms for SHE training,

retraining, and self-training, respectively. Results of modeling and simulation are presented in Sect. 6. Concluding remarks are presented at the end of the paper.

Note: all denotations of crisp and fuzzy variables that can be found in this paper are in normal style and their values are in italics.

### 3 SHE Initial Training

In general, SHE training process can be considered as a construction of a fuzzy rules set. An old-fashioned process of constructing fuzzy rules for SHE behavior, mostly based on the results of user’s interview, is considered as SHE training process. Usually a team of users, professional experts, and IT people accumulate, discuss, and prepare a set or requirements to be formalized and presented in the form of a special type of user cases and functional requirements for the system. Defined linguistic variables ( $x_1, x_2, \dots, x_N$ ) and types of their membership functions created rule set base (RB, consisting of general number  $I$  rules) and the identified defuzzification methods must be consider as the training result of a system.

Usually, if no special means are taken, the number of rules is expressed by the Descartes multiplication of all term numbers of all variables. So fighting the curse of dimensionality is the main problem of the initial training process. The goal of this section is to propose a hierarchical procedure to decrease the number of rules based on pragmatic causal interpretation of situations occurred in the HE.

In this paper, a simple SHE example (Fig. 2) is chosen to present the method used for minimizing the number of rules.

Let us imagine the room (HE) with two working places (WP1, WP2) equipped with light sources (L1, L2). These light sources are controlled from FDMS using actions (A1, A2) which mean the status of voltage supplied to the

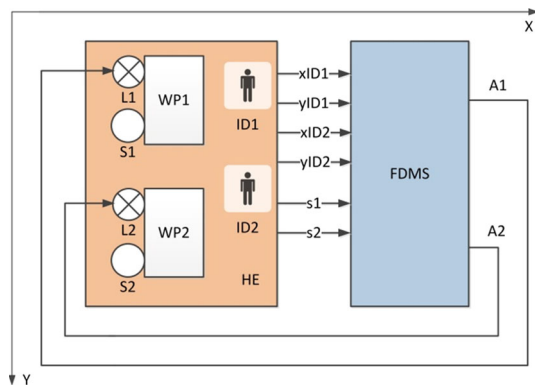


Fig. 2 Simplified example of SHE

corresponding light source, as “Increase”, “Decrease”, and “N” (do nothing).

Lighting ( $s_1, s_2$ ) in each working place is measured by sensors (S1, S2).

It is considered that two users (ID1, ID2) may express their different wishes concerning the light intensity in different working places and corresponding to different users’ positions in the HE room. It must be emphasized that user’s presence in the room and his or her position in it is determined by corresponding space coordinates ( $x_{ID1}, y_{ID1}$ , and  $x_{ID2}, y_{ID2}$ ).

A thorough analysis of the habits and wishes of users permits to create: (a) main terms for each variable under consideration, and (b) a set of rules to be performed in FDMS. The terms for space coordinates  $X$  and  $Y$  are shown in Fig. 3a, the terms for  $s_1$  and  $s_2$  lighting—in Fig. 3b and the output terms—in Fig. 3c.

In general, the frame of every rule consists of concatenation of two strings: the string of antecedences  $x_1, x_2, \dots, x_N$  in the IF part of the rule, and the string of consequences  $y_1, y_2, \dots, y_M$  in the THEN part of the rule. So the  $r$ -th rule is written in the following way:

$$\text{IF } \{x_1, x_2, \dots, x_N\} \text{ THEN } \{y_1, y_2, \dots, y_M\}$$

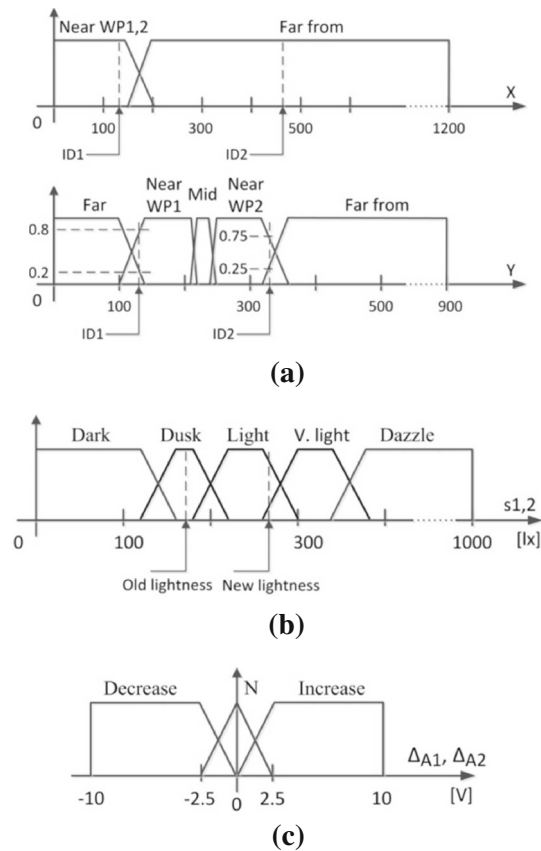


Fig. 3 Terms of Input a, b, and Output c Variables

According to [26], the first row in the list of rules can be considered as a string of corresponding attributes assigned to the sensors and actuators of the environment under consideration. All other rows are defined as verbal antecedence connected by a fuzzy logic relation AND (it means *min*) in the IF part, and verbal consequence obtained by a fuzzy logic operation CoG in the THEN part. For example, in our case such a frame is shown in Fig. 4.

As it was mentioned above, the number of rules is expressed by the Descartes multiplication of all term numbers of all variables; in HE example, the number is  $2 \times 5 \times 2 \times 5 \times 5 \times 5 = 2500$ .

The goal of this section is to propose a hierarchical procedure to decrease the number of rules based on pragmatic causal interpretation of situations that occur in the HE.

There are several rule reduction methods. In [27], a method of merging the variables is presented. [28] is based on the usage of a multi-level logic. [29–32] consider natural variable separations based on the specificity of controller design. Coefficients evaluating the rule’s redundancy are successfully used in [33]. Some of these methods are very efficient, but to our view, they are too artificial and specified (suitable only for the problem under investigation); they require too many additional efforts.

This paper presents a new approach for the minimization of rules, and this approach is based on a hierarchical description of HE situations. The hierarchy covers the augmenting knowledge of the events occurring in the home environment that needs to be intellectualized. The sequence of events is presented by hierarchical decision tree, as it is shown in Fig. 5.

In the experimental model of HE (Fig. 2), an equipped room and its users with their different PW concerning the light intensity in working places, was investigated. For example, the analysis of situations that occur in the room starts from the question: “Is one of the ID1 or ID2 in, or are they both in?” This analysis permits to use information included in the identity cards of users and behave according to the real situation instead of processing full information, even if part of it is irrelevant/redundant to this particular situation. For the sake of simplicity, only a crisp analysis of a situation was presented in this paper, however, the fuzzy reasoning-based situation analysis can be applied., for example, the algorithms of persons’/users’ identity recognition are possible as well.

After the crisp answers to those questions are received, the selection of a proper set of rules is usually made in the

|    |      |      |      |      |    |    |      |    |    |
|----|------|------|------|------|----|----|------|----|----|
| IF | XID1 | YID1 | XID2 | YID2 | S1 | S2 | THEN | A1 | A2 |
| IF |      |      |      |      |    |    | THEN |    |    |

Fig. 4 The frame of rules

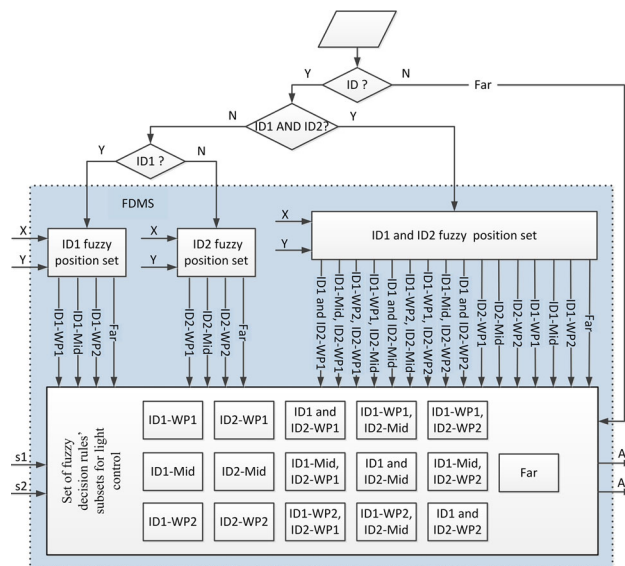


Fig. 5 Top-down hierarchical decision tree based on fuzzy logic

following hierarchical fuzzy blocks: “ID1 fuzzy position set”; “ID2 fuzzy position set”; “ID1 and ID2 fuzzy position set”, where the set of Takagi–Sugeno rules are used in the following way:

$$(r) \text{ IF } XID2 = a_r \text{ and } YID2 = b_r \text{ THEN GO TO } \max_{\forall r} z_r;$$

where  $z_r = K_r \mu(a_r) \times \mu(b_r)$  for  $\forall r$ ;  $\mu(*)$ —membership function of the corresponding value,  $K_r$ —positive weight coefficient, which emphasizes the significance of the  $r$ -th rule.

Frames of rules for sets of a fuzzy position are shown in Fig. 6.

In our example, blocks “ID1 fuzzy position set” and “ID2 fuzzy position set” contain 10 rules each, and “ID1 and ID2 fuzzy position set” contains a 100 rules.

After the user’s position in the room is determined, the levels of light intensity  $s1, s2$  must be investigated. One or another set of rules from the block “Set of fuzzy decision rules’ subsets for light control” (from Fig. 5) starts to work and provides a certain decision. It should be emphasized that the frame of rules in each set for light control has the same frame, presented in Fig. 7.

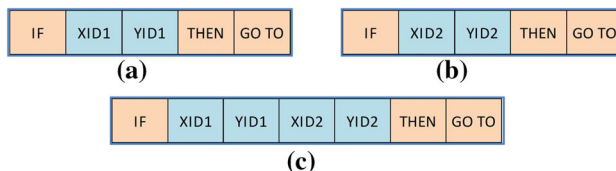


Fig. 6 Frames of rules for fuzzy position sets: for ID1-(a), for ID2-(b), and for ID1 & ID2-(c)

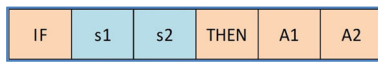


Fig. 7 Frame of rules for light control

Each of 16 blocks of “Set of fuzzy decision rules’ subsets for light control” contains 25 rules. So, a total number of rules for light control in our pending example in FDMS, when a hierarchical decision tree is used, is  $10 + 10 + 100 + 16 \cdot 25 = 520$ . The efficiency gained in comparison with the initial case is  $2500/520 = 4.8$ .

### 4 SHE Retraining

The trained SHE system has a rule set base (RB) consisting of I rules, as it was mentioned in Sect. 2. In this section a training algorithm is delivered for the general case according to the frame of rules presented in Fig. 4. It is obvious that in case of the hierarchical structure of a decision tree, the retraining process must be repeated for each structural block (Fig. 5).

SHE retraining process may be explained and understood by analyzing the changing wishes of user ID1, as given in Fig. 3b. It is shown that the former wish (Former brightness—Dusk) is changed into a new one—NPV (New Parameter Value—New brightness (Fig. 3b)). It means that from now on, the user ID1 will be satisfied with brightness of 270 lx at the working place WP1 ( $xID1 = 130$ ,  $yID1 = 130$ ), as it is shown in Fig. 3a.

The retraining may be considered as a construction of amendments in consequence of corresponding rules. So the first step is finding the rules that need to be changed.

According to F data of sensors that describe positions of users, a set of all possible antecedent parts R of rules is created (see Fig. 8a). For the sake of clarity, the truth values of corresponding terms (Fig. 3) are presented in the same figure as well. One antecedent part R(j) which corresponds to the highest product of membership functions of

| IF | XID1      | YID1    | XID2    | YID2    | S1 | S2 | THEN | A1 | A2 |
|----|-----------|---------|---------|---------|----|----|------|----|----|
| IF | 1         | 0,8     | 1       | 0,25    |    |    | THEN |    |    |
| IF | NearWP1,2 | NearWP1 | Farfrom | Farfrom |    |    | THEN |    |    |
| IF | 1         | 0,2     | 1       | 0,25    |    |    | THEN |    |    |
| IF | NearWP1,2 | Far     | Farfrom | Farfrom |    |    | THEN |    |    |
| IF | 1         | 0,8     | 1       | 0,75    |    |    | THEN |    |    |
| IF | NearWP1,2 | NearWP1 | Farfrom | NearWP2 |    |    | THEN |    |    |
| IF | 1         | 0,2     | 1       | 0,75    |    |    | THEN |    |    |
| IF | NearWP1,2 | Far     | Farfrom | NearWP2 |    |    | THEN |    |    |

(a)

| IF | XID1      | YID1    | XID2    | YID2    | S1 | S2 | THEN | A1 | A2 |
|----|-----------|---------|---------|---------|----|----|------|----|----|
|    | NearWP1,2 | NearWP1 | Farfrom | NearWP2 |    |    |      |    |    |

(b)

Fig. 8 Generated set R of all possible rules (a) and the selected rule R(j) (b)

the corresponding values of parameters is left and used for further operations (Fig. 8b).

An antecedent part of this rule is used to select all rules to be changed in Rule set base obtained during the initial training process. Consequent parts of the selected rules are changed according to the pseudo code and the algorithm presented in Fig. 9.

The whole process is organized as follows (Fig. 9): (1) the selected rule R(j) is consequently compared with the rule R(i)  $\forall i$  (here i is the number of the rule) from the Rule set base; (2) if the correspondence of A of the rule R(j) with the part B of the rule R(i) is determined, then the

Generate a set of all possible antecedent parts R of rules under investigation. Select one antecedent part R(j) corresponding to the highest product of membership function values of the corresponding parameters.

```

for i = 1,2,...,I
  if R(i) contains all antecedents of R(j) then
    if NPV > R(i).s1T[c] then
      set A1 to Increase
    else
      if NPV < R(i).s1T[b] then
        set A1 to Decrease
      else
        set A1 to N
      end if
    end if
  end if
end for
    
```

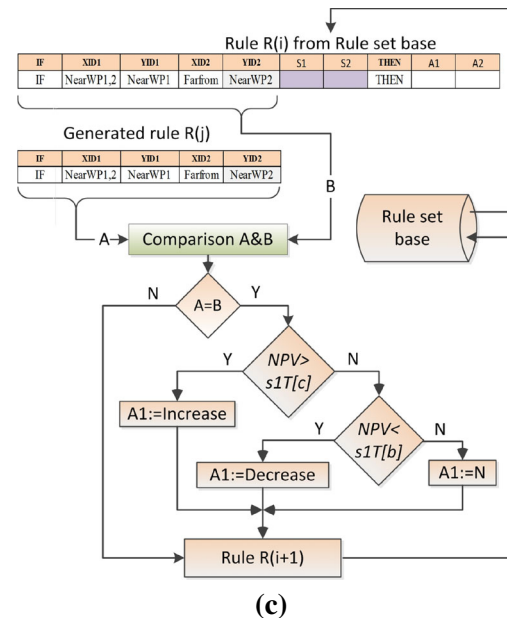
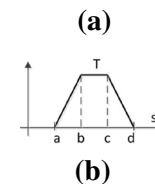


Fig. 9 Pseudo code to determine output term A1 (a), a typical term T of the input variable s1 (b), and algorithm corresponding to the pseudo code (c)



consequent A1 in rule R(i) must be changed to “Increase”, “Decrease”, or “N” (see Fig. 3c) according to the antecedent of s1 in this rule, as it is presented by the pseudo code (Fig. 9a and the block diagram in Fig. 9c). Here  $sIT[b]$  and  $sIT[c]$  are values s1 of the term T (Fig. 9b). The T is each possible term of variable S1—“Dark”, “Dusk”, “Light”, “V. Light”, “Dazzle” (Fig. 3b).

After the retraining process is finished, SHE will take decisions according to the new wish of user ID1.

### 5 SHE Self-training

In this section, the self-training process is considered as a SHE training “from zero”, i.e., when we have no priori information concerning the user’s wishes. At least two important cases must be distinguished: (1) terms of variables under consideration are known in advance; (2) only minimal and maximal values of variables are known. The main difference between these two cases is the need to build necessary terms from data produced at the moments when user expresses his wishes in case 2. Both cases are demonstrated in a simple SHE example, described in Sect. 3.

#### 5.1 Self-training: Case 1

As it was mentioned before, in this case all terms of input and output variables are known and shown for our simplified example in Fig. 3. The frame of possible rules is considered as known as well, and is presented in Fig. 4. Let us imagine that the user ID1 while at a position xID1, yID1, as it is shown in Fig. 3a, expresses his first wish, for example, “New lighting” by adjusting the lighting 270 lx in the working place WP1 (Fig. 3b). The second user ID2 has not expressed any wish and at the moment is not taken into account, except for his position.

The self-training process in case 1 is organized as follows: (1) Set of all possible antecedent parts R of rules is created to determine the positions of ID1 and ID2 according to terms shown in Fig. 3a similarly as it was done in Fig. 8a. (2) One antecedent part R(j) corresponding to the highest product of membership functions of the corresponding parameters values is left for further operations (see Fig. 8b). (3) This left antecedent part R(j) is expanded by the burst B, consisting of K new rules with all possible combinations of five terms S1 and S2 (Fig. 10). As a matter of fact, the preliminary list of those 25 rules consists only of antecedents, and the next step serves for the determination of their consequents. (4) All those consequents are determined according to the pseudo code and algorithm depicted in Fig. 11. It must be emphasized that  $sIT[b]$  and  $sIT[c]$  are s1 values of S1 of the term T. The T is each

| IF | XID1      | YID1    | XID2    | YID2    | S1      | S2      | THEN | A1       | A2 |
|----|-----------|---------|---------|---------|---------|---------|------|----------|----|
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dark    | Dark    | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dusk    | Dark    | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Light   | Dark    | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | V.light | Dark    | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dazzle  | Dark    | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dark    | Dusk    | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dusk    | Dusk    | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Light   | Dusk    | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | V.light | Dusk    | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dazzle  | Dusk    | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dark    | Light   | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dusk    | Light   | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Light   | Light   | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | V.light | Light   | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dazzle  | Light   | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dark    | V.light | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dusk    | V.light | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Light   | V.light | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | V.light | V.light | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dazzle  | V.light | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dark    | Dazzle  | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dusk    | Dazzle  | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Light   | Dazzle  | THEN | Increase | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | V.light | Dazzle  | THEN | Decrease | N  |
| IF | NearWP1,2 | NearWP1 | FarFrom | NearWP2 | Dazzle  | Dazzle  | THEN | Decrease | N  |

Fig. 10 Burst B of generated rules

```

Generate a set of all possible antecedent parts R of rules under investigation.
Select one antecedent part R(j) corresponding to the highest product of membership function values of the corresponding parameters
Expand rule R(j) by generating a burst B consisting of K new rules with all possible combinations of five terms S1 and S2
for k = 1,2,...,K
    if NPV > B(k) sIT[c] then
        set A1 to Increase
    else
        if NPV < B(k) sIT[b] then
            set A1 to Decrease
        else
            set A1 to N
        end if
    end if
    Add rule B(k) to existing rule set base
end for
    
```

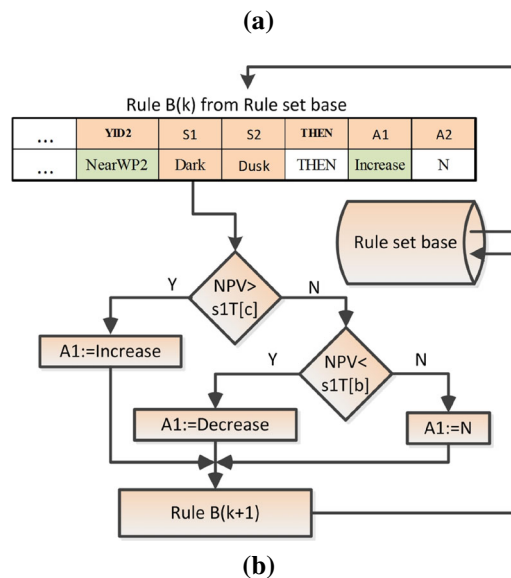


Fig. 11 Pseudo code to determine the output term A1 (a) and the algorithm corresponding to the pseudo code (b)

possible term of variable S1 (like as S1 in Fig. 9b). Determined consequents are presented in the action column A1 (Fig. 10). A column A2 must be left unchanged because there is no information concerning brightness s2 at the working place WP2.

Analogous steps must be performed in case if user ID1 expresses another wish that is different and/or the second user ID2 expresses any wish, and so on.

### 5.2 Self-training: Case 2

In case when only minimal and maximal values of input variables are known, the self-training procedure starts from building necessary terms from data produced at the moments when a user expresses his wishes. It must be stressed that terms expressing possible SHE actions are known in advance, as it was described in Sect. 3. By the way, rule generation procedure consists of the same steps as in case 1 (Sect. 4 A). The number of produced terms increases with every new wish the user makes. So in general, FDMS (Fig. 1, 2) may be considered as a complex of cumulative set of terms and cumulative list of rules.

The process of set generation for terms during the self-training in case 2 is organized as follows:

- (1) Starting with the first new wish *NW*, a corresponding term “Wish1” is introduced (Fig. 12b). It must be emphasized that the triangular term with a base from *Min S* till *Max S* is chosen only for the example under consideration. Many other types of terms (trapezium type, with different fixed base width and so on) may be used after consulting with the user and IT specialists.
- (2) Two additional terms expressing the so-called contradicting wishes (C\_Wish) for terms “Less” (C\_WL) and “More” (C\_WM) must be introduced as well. The shape of all those terms ready for computerized processing is described in Fig. 12. Here a, b, and c denote the values of parameters of a corresponding triangular term (Fig. 12c, d). In Fig. 12a, pseudo code is presented for the generation of terms, here  $Sterm[i] = \{Wish[i], C\_WL, C\_WM\}$ . In Fig. 12e, f, the process of generating new terms (Wish2, Wish3, Wish4) is presented.
- (3) Further, step 1 and step 2 have to be repeated every time when a new wish appears and the terms of wishes in the Figs. 12e are to be listed according to increasing values of the parameter b. During this process, [a, c] of all terms are subjected to the changes widely recommended in literature [34].

It is important to mention that in the process, we use the same output action terms in the self-training as it was done in training and retraining processes.

```

Find input variable S to be fuzzified
if S has no terms then
  Create new term C_WL with values of a=Min S, b = Min S and c=NW
  Create new term Wish1 with values of a = Min S, b = NW and c=Max S
  Create new term C_WM with values of a=NW, b = Max S and c=Max S
  Add newly created terms to existing terms of S
else
  if value b of any S term equals to NW then
    Do not create new term
  else
    for i = 1, ..., 2, 1; (I is number of all existing S terms)
      if Sterm[i] b < NW then
        LeftTerm = Sterm[i]
        RightTerm = Sterm[i+1]
        LeftTerm c = NW
        RightTerm a = NW
        Create new term WishX with values of a=LeftTerm[b], b =
        NW and c = RightTerm[b], where X=(I-2)+1, (-2 – because we subtract two
        terms: C_WL and C_WM to get a number of Wish terms, +1 – to increase
        term numeration by 1)
        Add newly created term WishX to existing terms of S
        Increase I by 1
      end if
    end for
  end if
end if
Continue by creating new rules (if necessary)
    
```

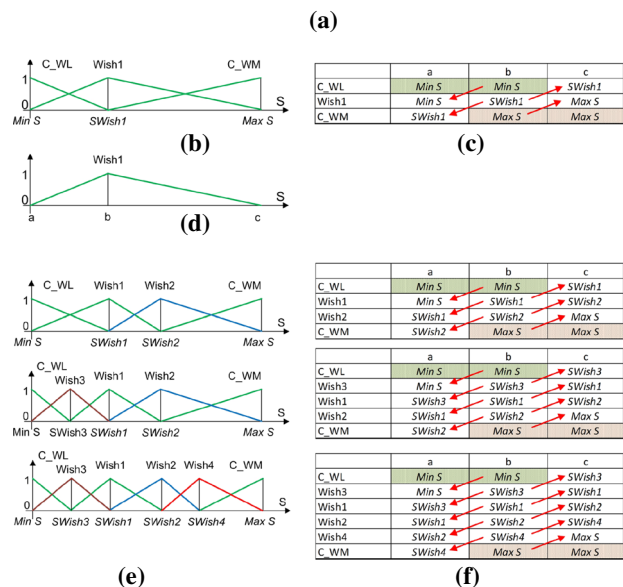


Fig. 12 Description of term generation procedure in the self-training case

## 6 Modeling and Simulation Results

All training, retraining, and self-training procedures were simulated in a simple SHE test environment presented in Fig. 2. The simulation was performed on the SHE model developed as a special software/hardware system BIASim shown in Fig. 13. The BIASim permits to represent and visualize all gadgets, sensors, actuators, and moving SHE users who are to benefit from the comfort provided by intellectualized actions of the environment. Agent-based fuzzy intelligence is stored in the Fuzzy Decision Model library (FDM). Possible environmental situations are stored





**Table 1** Retraining results for four cases

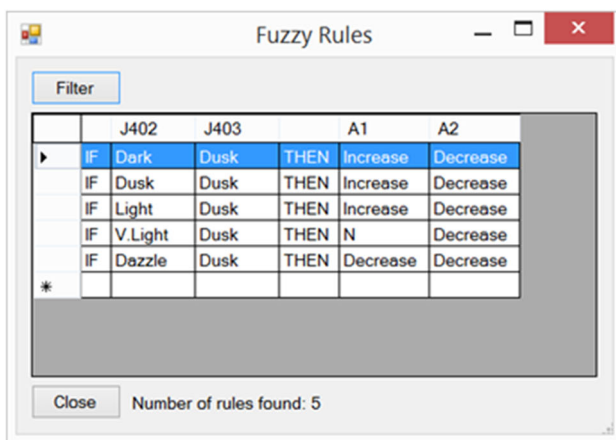
|                               |         |         |         |         |
|-------------------------------|---------|---------|---------|---------|
| Previous lighting             | 160–180 | 220–260 | 300–340 | 300–340 |
| New lighting set by ID1, lx   | 240     | 310     | 290     | 190     |
| Lighting after retraining, lx | 220     | 300     | 280     | 197     |

Otherwise, when the user’s new wish belongs to the bases of two neighboring terms, the intensity of lighting reached after retraining in the working place is determined by the point of crossing of these terms.

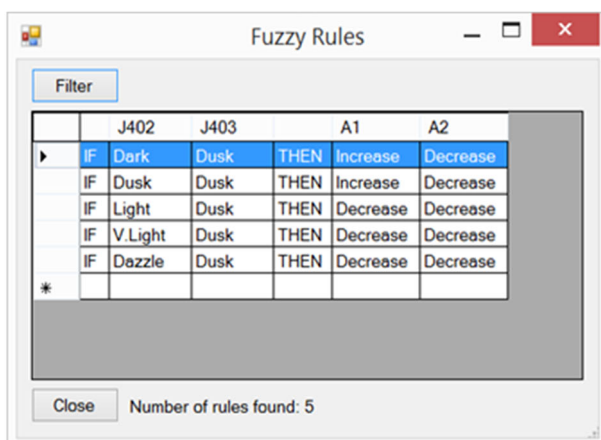
An example of rules for lighting control generated experimentally according to the proposed algorithms under consideration and used before retraining (initial rules) and rules after retraining are presented in Fig. 15a, b correspondingly. In Fig. 15, J402 stands for a lighting sensor S in the working place WP1.

Experiment for self-training in case 1 was made for users ID1 and ID2 separately with the same coordinates: xID = 130 and yID = 130. In this case, all terms are known and they are as presented in Fig. 3. But up until now, the user has not expressed any wishes concerning

concrete level of lighting in the working place. When modeling the self-training process, the lighting desired and set by user ID1 is 210 lx, and the one set by user ID2—240. Five rules were generated for ID1 and five for ID2. After the self-training process, lighting in the working place WP1 for ID1 and ID2 was set to 200 lx and 220 lx accordingly. The concrete type of terms “Dusk” and “Light” (Fig. 3b) predetermine a precision of system’s response as described above. Here it must also be

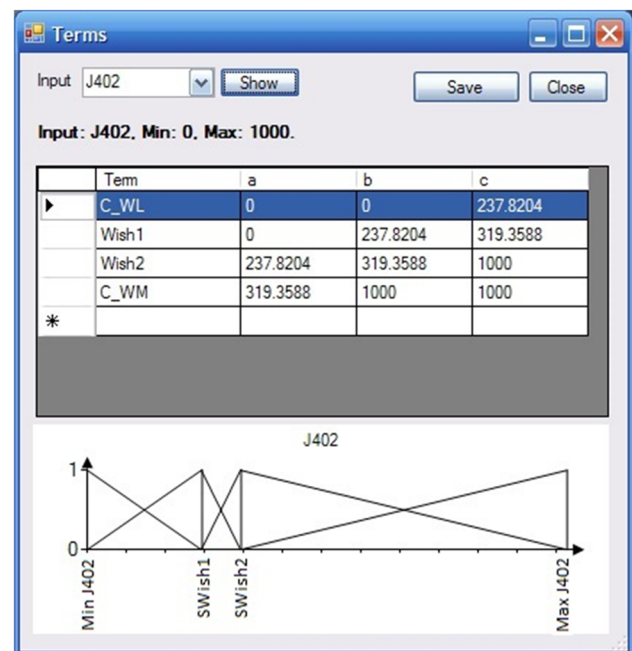


(a)

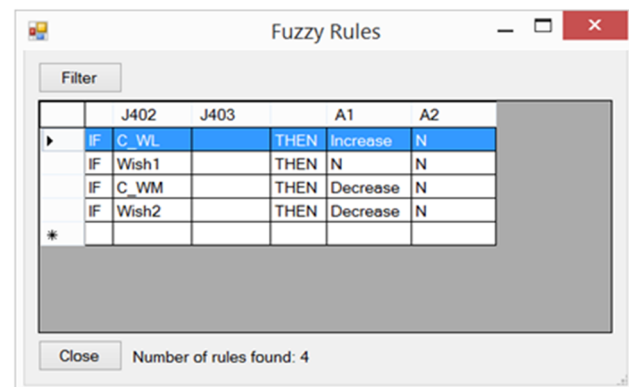


(b)

**Fig. 15** Initial rules (a), and rules after retraining (b)



(a)



(b)

**Fig. 16** BIAsim screenshot for visualization of two terms Wish1 and Wish2 generation (a) and created rules (b)

emphasized that modeling was performed when user starts acting in a darkened working place.

Experiment for self-training in case 2, when only Min and Max values of parameters are known (Min  $x = 0$ , Max  $x = 1200$ , Min  $y = 0$ , Max  $y = 900$ , Min  $S = 0$  and Max  $S = 1000$ ), was performed in two steps: (1) User ID1 at coordinates  $x_{ID1} = 130$  and  $y_{ID1} = 130$  expresses his wish (SWish1) and sets lighting to 238 lx. Three terms for  $x$ , three terms for  $y$ , three terms for  $S$ , and three rules were generated. After the self-training process, and the ID1  $x$  and  $y$  coordinates were set, lighting supplied by the system was 231 lx. 2) User ID2 at coordinates  $x_{ID1} = 130$  and  $y_{ID1} = 130$  expresses his wish (SWish2) and sets lighting to 320 lx. The fourth term for  $S$  and four rules were generated. After the self-training process, when ID1 or ID2 position in terms of  $x$  and  $y$  coordinates were set, lighting achieved by the system was 231 and 318 lx correspondingly.

An example of modeling results with generated  $S$  terms according to the proposed algorithm is presented in Fig. 16a. Here, J402 corresponds to the lighting  $S$  in WP1. Generated rules for lighting ID1 in WP1 are presented in Fig. 16b.

This modeling and simulation confirms the vitality of the whole idea. Including the newly proposed algorithms for fuzzy training, retraining, and self-training for the intellectualization of home environment has expanded SHE's possibilities. Efficiency of top-down hierarchical analysis of home situations under consideration when conquering the curse of increasing number of rules is demonstrated.

## 7 Concluding Remarks

The main research ideology was developed and implemented under the guidance of the COST Action IC0702 "Combining Soft Computing Techniques and Statistical Methods to Improve Data Analysis Solutions (SOFT-STAT)". This research was supported and used in the EU Structural Funds projects: VP1—3.1-ŠMM-08-K-01-018 "Research and Development of Internet Infrastructure for IoT&S in the Smart Environment (IDAPI)" and VP1-3.1-ŠMM-10-V-02-020 "Research on Smart Home Environment and Development of Intelligent Technologies (BIATech)".

In this paper, the opinion that a set of electronically controlled smart things and the whole home environment must be intellectualized using human-type reasoning to increase person's comfort was advocated.

A novel approach and new algorithms for a hierarchical fuzzy training, retraining, and self-training of an intellectualized home environment are proposed. Fuzzy logic-based training algorithms take top-down hierarchical analysis of home situations under consideration to conquer the

curse of increasing number of rules. A successful combination of crisp algorithms for users' presents/absence identification in the environment and the fuzzy logic-based algorithms enables significantly to decrease the number of necessary rules. In the paper, a case is presented where 2500 rules were decreased approximately 5 times.

For the first time the changes of users' wishes are taken into account during the retraining process.

An entirely new ability of the system was investigated: a fuzzy logic-based algorithm to start the self-training process without any a priori information is developed. Vitality and efficiency of the proposed methodology was tested and simulated on a specialized virtual software/hardware modeling system. The modeling performed for the working place light control in conditions close to real home environment confirmed the possible practical efficiency of the investigated methodology.

Investigations of the control speed and accuracy, automation of the inclusion procedure of new variables, solving the scalability problems can be considered as topics for further research and development.

## References

- History of the internet of things: <http://postscapes.com/internet-of-things-history>
- D1.1: Requirements and exploitation strategy/BUTLER, ALBLF, No 287901, p. 178. <http://www.iot-butler.eu/download/deliverables> (2012)
- D3.2: Integrated system architecture and initial pervasive BUTLER proof of concept/BUTLER, ERC, No 287901, p. 191. <http://www.iot-butler.eu/download/deliverables> (2013)
- The IEEE Standards Association (IEEE-SA): Internet of Things (IoT) Workshop, 5–6 November 2013 in Silicon Valley, Calif. <http://sites.ieee.org/wf-iot/> (2013)
- IEEE World Forum on Internet of Things (WF-IoT): 6–8 March 2014, Seoul, Korea. <http://sites.ieee.org/wf-iot/> (2014)
- Consultation on Future Network Technologies Research and Innovation in HORIZON2020: 29 June 2012, Brussels, Workshop Booklet. <http://cordis.europa.eu/fp7/ict/future-networks/documents/h2020-fn-consultation-booklet.pdf>
- Internet of Things in 2020: A ROADMAP FOR THE FUTURE, INFSO D.4 NETWORKED ENTERPRISE & RFID INFSO G.2MICRO & NANOSYSTEMS, in co-operation with the RFIDWORKING GROUP OF THE EUROPEAN TECHNOLOGY PLATFORM ON SMART SYSTEMS INTEGRATION (EPOSS); 05 September 2008; European Commission; Information Society and Media, (2008)
- The Internet of Things 2012: New Horizons. In: Smith, I.G. (ed.), Vermesan, O., Friess, P., Furness, A. (Tech. eds.). Printed by Platinum, published in Halifax, UK (2012)
- Future Network Technologies Research and Innovation in HORIZON2020, Workshop Report, 29 June 2012, Brussels
- Zeng, Y.-R., Wang, L., He, J.: A novel approach for evaluating control criticality of spare parts using fuzzy comprehensive evaluation and GRA. *Int. J. Fuzzy. Syst.* **14**(3), 392–401 (2012)
- Wang, L., Fu, Q.-L., Lee, C.-G., Zeng, Y.: Model and algorithm of fuzzy joint replenishment problem under credibility measure on fuzzy goal. *Knowl. Based Syst.* **39**, 57–66 (2013)

12. Shih, Y.-Y., Su, S.-F., Rudas, I.J.: Fuzzy based compensation for image stabilization in a camera hand-shake emulation system. *Int. J. Fuzzy. Syst.* **16**(3), 350–357 (2014)
13. Unbridled Spirit: The Story Behind Fuzzy Blood Pressure Monitoring Development. OMRON HEALTHCARE Co., Ltd. Published by Medical Public Relation Group, p. 16 (2013)
14. Uckermann, D., Harrison, M., Michahelles, F. (eds.). *Architecting the internet of things*. ISBN: 978-3-642-19156. [www.springerlink.com](http://www.springerlink.com) (2011)
15. Nakashima, H., Aghajan, H., Augusto, J.C. (eds.): *Handbook of Ambient Intelligence and Smart Environments*. Springer, New York (2010)
16. Vasseur, J.-P., Dunkels, A.: *Interconnecting Smart Objects with IP—The Next Internet*. Morgan Kaufmann, Burlington (2010)
17. Duman, H., Hagra, H., Callaghan, V.: Intelligent association exploration and exploitation of fuzzy agents in ambient intelligent environments. *J. Uncertain Syst.* **2**(2), 133–143 (2008)
18. Luo, M., Sun, F., Liu, H.: Hierarchical structured sparse representation for T-S fuzzy systems identification. *IEEE Trans. Fuzzy Syst.* **21**(6), 1032–1043 (2013)
19. Cook, D.J., Crandall, A.S., Thomas, B.L., Krishnan, N.C.: CASAS: a smart home in a box. *Computer* pp. 62–69 (2013)
20. Roggen, D., Troster, G., Lukowicz, P., Ferscha, A., Millan, J.R., Chavarriaga, R.: Opportunistic human activity and context recognition. *Computer* **46**, 36–45 (2013)
21. Acierno, A., Esposito, M., Pietro, G.: An extensible six-step methodology to automatically generate fuzzy DSSs for diagnostic applications. *BMC Bioinform.* **14**(Suppl 1) S4. <http://www.biomedcentral.com/1471-2105/14/S1/S4> (2013)
22. Vainio, A.M., Valtonen, M., Vanhala, J.: Learning and Adaptive Fuzzy Control System for Smart Home. *Developing Ambient Intelligence*, pp. 28–47. Springer, New York (2006)
23. Jasinevicius, R., Kazanavicius, E., Petrauskas, V.: Intellectualized home environment as a Cpm[lex System. *ISCS 2014: Interdisciplinary Symposium on Complex Systems*, pp. 87–97. Springer, New York, (2014)
24. Jasinevicius, R., Jukavicius, V., Liutkevicius, A., Petrauskas, V., Taraseviciene, A., Vrubliauskas, A.: Methods for smart home environment's intellectualization: the comparative analysis. In: *Information and Software Technologies. ICIST 2014, Proceedings*, pp. 150–159. Springer, New York (2014)
25. Jasinevicius, R., Petrauskas, V.: On fundamentals of global systems control science (GSCS). In: *ISCS 2013 Interdisciplinary Symposium on Complex Systems*, pp. 77–88. Springer, Berlin (2014)
26. Lautert, L.R., Scheidt, M.M., Dorneles, C.F.: Web table taxonomy and formalization. *SIGMOD Rec.* **42**(3), 28–33 (2013)
27. Ledeneva, Y., Garcia, C., Martinez, J.: Automatic estimation of fusion method parameters to reduce rule base of fuzzy control complex systems. In: Gelbukh A., et al. (eds.) *MICAI 2006, LNAI 4293*, pp. 146–155. Springer, Berlin (2006)
28. Cococcioni, M., Foschini, L., Lazzarini, B., Marcelloni, F.: Complexity reduction of mamdani fuzzy systems through multi-valued logic minimization. In: *Proceedings of 2008 IEEE System, Man and Cybernetics (IEEE SMC'08)*, pp. 1782–1787. Singapore, 12–15 October 2008
29. Kandroodi, M.R., Mansouri, M., Shoorehdeli, M.A., Teshnehlab, M.: Control of flexible joint manipulator via reduced rule-based fuzzy control with experimental validation. *Int. Sch. Res. Netw. ISRN Artificial Intelligence*, vol. 2012, Article ID 309687. doi:10.5402/2012/309687. <http://www.hindawi.com/journals/isrn.artificial.intelligence/2012/309687/>
30. Piltan, F., Sulaiman, N., Zargari, A., Keshavarz, M., Badri, A.: Design PID-like fuzzy controller with minimum rule base and mathematical proposed on-line tunable gain: applied to robot manipulator. *Int. J. Artif. Intell. Exp. Syst. (IJAE)* **2**(4), 184–195 (2011)
31. Chopra, S., Mitra, R., Kumar, V.: Reduction of Fuzzy Rules and Membership Functions and Its Application to Fuzzy PI and PD Type Controllers. *Int. J. Control Autom. Syst.* **4**(4), 438–447 (2006)
32. García, F., Martínez, P., Paz, V.: Rule base reduction on a self-learning fuzzy controller. <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.145.5507>
33. Ciliz, M.: Rule base reduction for knowledge-based fuzzy controllers with application to a vacuum cleaner. *Expert Syst. Appl.* **28**, 175–184 (2005)
34. Kosko, B.: *Fuzzy engineering*. Prentice-Hall, Upper Saddle River (1997)