

# INTERPOLATION BASED ON SEMANTIC DISTANCE WEIGHTING IN HEDGE ALGEBRA AND ITS APPLICATION

NGUYEN TIEN DUY<sup>1</sup>, VU NHU LAN<sup>2</sup>

<sup>1</sup>*Thai Nguyen University of Technology, Thai Nguyen University*

<sup>2</sup>*Thang Long University*

<sup>1</sup>*duy.infor@tnut.edu.vn; <sup>2</sup>vnlan@ioit.ac.vn*



**Abstract.** When solving the approximation problems proposed by a Linguistic Rule Base System (LRBS), the choice of reasoning method has a great influence on the result. The chosen reasoning method should ensure the correctness of the variable relationship between the outputs versus the inputs. According to the Hedge Algebra's approach, we can convert the LRBS to a real "hyper surface" in semantic space, called the Quantified Semantic Rule Base System (QRBS), and use an interpolative method for reasoning. In this paper, we propose the use of the interpolation to ensure the correctness mentioned above. It is interpolation based on the semantic distance weighting of the input semantic values versus the semantic value of the elements appearing in the rule. In the input semantic value vector, for each component we define its distance weight to the semantic value of the corresponding language class that appears in the rule. This distance-weighted value is used to join the output value. With this proposal, we have designed and simulated the controller that control rules are given by LRBS for the resistance furnace. The results show that the controller works well under the control requirements.

**Keywords.** Hedge algebras, approximate reasoning, bi-linear interpolation, linguistic rule base system, fuzzy control.

## 1. INTRODUCTION

For the past few years, the theory of hedge algebras (HA) [9, 10] has been used to solve problems in many different fields, including the field of control [4, 5, 6, 12, 13, 16]. According to the HA approach, the design of approximately reasoning set working as a controller whose control rules are given by the LRBS has many advantages [8]. The way to represent knowledge in real "hyper surface" in semantic space reflects the LRBS correctly, in which it is the semantic order of the language values that is preserved. Solving the problem of inference only by the method of interpolation on this "hyper surface" [16]. Among a few of interpolation methods, they have a small number of calculations. This is an important advantage for systems that require real-time response as the system in [16]. In the structure of the reasoning set, there are not many influencing parameters, that is, the parameters of the HA's fuzzy parameters should be convenient for adjustment and optimization.

Consider a control rule set MISO given by LRBS:

$$\begin{aligned}
&\text{If } \mathcal{X}_1 = A_{11} \text{ and } \dots \text{ and } \mathcal{X}_m = A_{1m} \text{ then } \mathcal{Y} = B_1 \\
&\text{If } \mathcal{X}_1 = A_{21} \text{ and } \dots \text{ and } \mathcal{X}_m = A_{2m} \text{ then } \mathcal{Y} = B_2 \\
&\dots \\
&\text{If } \mathcal{X}_1 = A_{n1} \text{ and } \dots \text{ and } \mathcal{X}_m = A_{nm} \text{ then } \mathcal{Y} = B_p
\end{aligned} \tag{1}$$

with  $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_m$  and  $\mathcal{Y}$  are linguistic variables, each linguistic variable  $\mathcal{X}_j$  belongs to the base space  $U_j$  and the linguistic variable  $\mathcal{Y}$  belongs to the base space  $V$ ;  $A_{ij}, B_k$  ( $i = 1..n, j = 1..m, k = 1..p$ ) are the linguistic values that belong to corresponding background space. By approaching the HA theory, we construct the HA structure for linguistic variables and use the semantically quantifying mapping function SQMs [8] to convert the linguistic values with a real “hyper surface”  $S_{real}^{m+1}$  in semantic space  $[0, 1]^{m+1}$ . At that time, it is possible to consider this hyperreal  $S_{real}^{m+1}$  as the mathematical representation of the LRBS.

Suppose that real inputs belonging to the corresponding base space are the input values of the controllers  $x_{01}, x_{02}, \dots, x_{0m}$ , the Normalization of those values in the value domain of HA is used to obtain  $x_{01s}, x_{02s}, \dots, x_{0ms}$  respectively. The solution of the problem of approximation by the interpolation method on  $S_{real}^{m+1}$  is carried out. The interpolation value received in the domain  $[0, 1]$  is the the semantically quantifying value of the output linguistic variable  $\mathcal{Y}$  that is transferred to the real variable domain (base space of variable  $\mathcal{Y}$ ) of the control variable at the output by Denormalization.

The diagram of the approximate reasoning set based on the HA approach is described as in Figure 1.

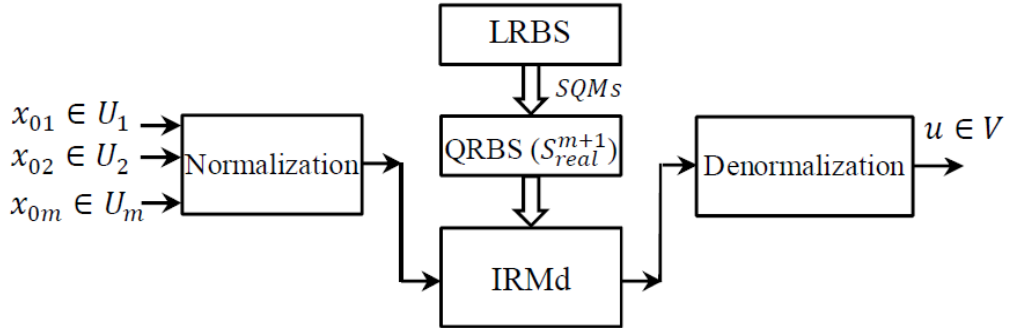


Figure 1. The diagram of the approximate reasoning set based on the HA approach

where:

- LRBS: Linguistic rule based system of the controller;
- QRBS: Quantifying rule based system of linguistic values which is computed by mapping function SQM ( $S_{real}^{m+1}$ );
- Normalization: standardize values of the variables in the semantic domain;
- IRMd (Interpolation Reasoning Method): Interpolative method on the “hyper surface”  $S_{real}^{m+1}$ ;

- Denormalization: convert semantic control value to the domain of variable real value of the output variable.

As can be seen in Figure 1, choosing an IRMd plays an important role in determining the output result of the inference. The chosen interpolation method should ensure a variable relationship between the output and the inputs.

In the last studies, authors could use the PRODUCT instead of the “and” in the rule to aggregate  $m$  inputs into a single input [3, 14]. Another way to aggregate inputs is to use weighted connection [7, 14]. At that time,  $S_{real}^{m+1}$  surface will be transformed into  $C_{real}^2$  in 2-D space, which makes the interpolation simpler. However, the access may cause loss of information. The curve  $C_{real}^2$  will no longer fully describe the semantics of the original LRBS, and may even cause inconsistencies between the rules. This results in inaccurate inferences. Direct interpolation on the semantic surface [1, 2] or using bi-linear interpolation [11] is a good solution to overcome the disadvantages of the above connection. However, there is a drawback that this method is only computable in the case of interpolation. When the input value is outside the spatial domain of  $S_{real}^{m+1}$ , the computation of bi-linear interpolation value becomes extrapolated. Basing on analysis, we find that in extrapolation, the calculated value does not guarantee the variable correlation between the output and the inputs based on LRBS. Suppose that there is a QRBS of a given controller as shown in Table 1. The semantic surface of  $S_{real}^3$  corresponds to the data in Figure 2a.

Table 1. QRBS of the controller

$e \backslash ce$	0.000	0.125 ( $v_{1ce}$ )	0.375 ( $v_{2ce}$ )	0.500	0.625	0.875	1.000
0.000							
0.125 ( $v_{1e}$ )		$u_{11} = 0.125$	$u_{12} = 0.125$	0.250	0.375	0.500	
0.375 ( $v_{2e}$ )		$u_{21} = 0.125$	$u_{22} = 0.250$	0.375	0.500	0.625	
0.500		0.250	0.375	0.500	0.625	0.750	
0.625		0.375	0.500	0.625	0.750	0.875	
0.875		0.500	0.625	0.750	0.875	0.875	
1.000							

where, the semantic value of the input variables  $e, ce = \{0.125, 0.375, 0.5, 0.625, 0.875\}$ . The semantic values of the output variable  $u$  corresponding each rule are the cells in the table.

According to QRBS in Table 1, we find that the variation of the output of  $u$  is “covariated” with the input components  $e$  and  $ce$ . The yellow and green squares in Table 1 are the lack of knowledge from the rule system ( $e, ce < 0.125$  or  $e, ce > 0.875$ ). For real values at the input of this domain, the inference value must be calculated by other knowledge values in the table. In that case, the computation based on bi-linear interpolation turns to extrapolation. Suppose there is an input vector  $v_1 (e_1 = 0.1, ce_1 = 0.1) < v_2 (e_2 = 0.125, ce_2 = 0.125)$ .

According to the variable rule of “covariated” of the rule table, there must have the output  $u_1 < u_2$ . However, the calculation based on the interpolation formula:

- For the input vector  $v_1$ , the interpolative steps are computed as follows (Figure 2b):
 
$$\text{B1: } t_1 = u_{11} + \frac{(e_1 - v_{1e})(u_{21} - u_{11})}{v_{2e} - v_{1e}} = u_{11} = 0.125$$

$$\text{B2: } t_2 = u_{12} + \frac{(e_1 - v_{1e})(u_{22} - u_{12})}{(v_{2e} - v_{1e})} = 0.125 + \frac{(0.1 - 0.125)(0.25 - 0.125)}{(0.375 - 0.125)} = 0.1125$$

$$\text{B3: } u_1 = t_1 + \frac{(ce_1 - v_{1ce})(t_2 - t_1)}{(v_{2ce} - v_{1ce})} = 0.125 + \frac{(0.1 - 0.125)(0.1125 - 0.125)}{(0.375 - 0.125)} = 0.12625$$
- For the input vector  $v_2$ , the interpolative steps are computed similarly to give  $u_2 = 0.125$ .

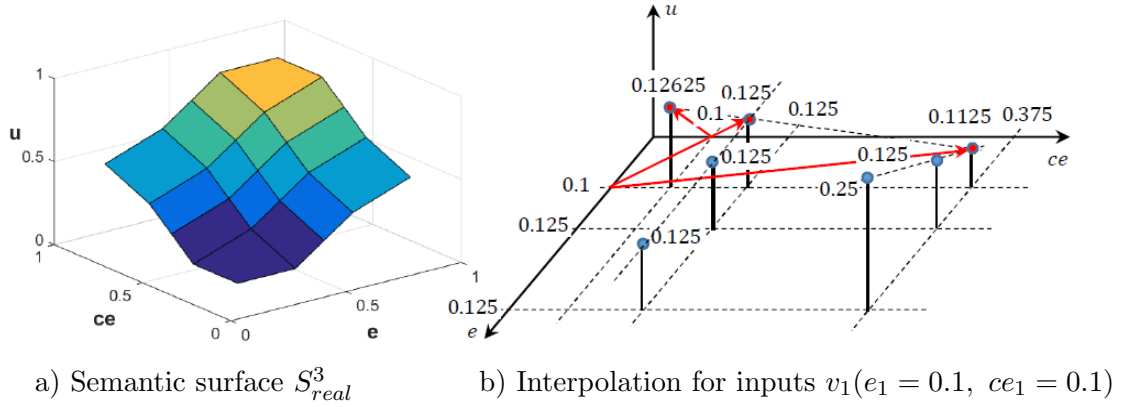


Figure 2. Semantic surface corresponding Table 1 and extrapolative computation

It can be seen that  $(u_1 = 0.12625) > (u_2 = 0.125)$ . This shows that the bi-linear interpolative method does not guarantee the correlation between the output and the inputs in the case of extrapolative computation. The inference value loses its correctness in some areas of the input/output relationship surface. To overcome this problem, we propose an interpolation method based on the semantic weighted distance of the input values versus the semantic values of the linguistic terms appearing in the rule. After applying the suggestion for resistor furnace controller, the simulation results show that the controller works well according to the control requirements.

## 2. INTERPOLATION BASED ON THE SEMANTIC WEIGHTING DISTANCE

In this section, we present a proposal for an interpolative method based on the semantic weighting distance of the input values versus the semantic value of the terms appearing in the rule.

In reality, for each linguistic rule, the output conclusions (the value of the concluding statement) only depend on the input (the value of the conditional clause). In the domain of semantic values, it is assumed that each input semantic value only defines a degree of satisfaction for conditional clauses. They have semantic values that are upper and lower bounds

of the semantic distance containing it. Figure 3 describes the distribution of the semantic values of the language classes  $x_i, x_j, \dots, x_k, x_m$  which belong to the language variable  $X, \dots$

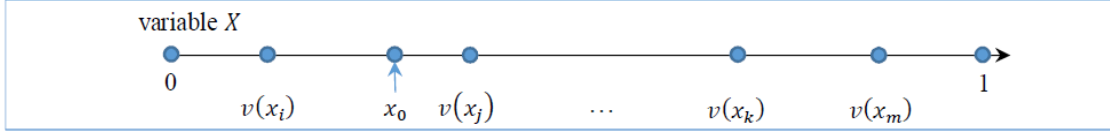


Figure 3. Input  $x_0$  will satisfy the conditional rules contained  $v(x_i), v(x_j)$

As can be seen in Figure 3, the input semantic value  $x_0$  is in the range from  $v(x_i)$  to  $v(x_j)$ , which are semantically closest to  $x_0$ . Thus only the clauses  $x = x_i$  and  $x = x_j$  are defined as a degree of satisfaction. Each clause will have a degree of satisfaction  $0 \leq w \leq 1$ . The closer the semantic values at the input of the semantic rule to the semantic value of the linguistic category in the conditional clause are, the greater the degree of satisfaction of the clause will be. We call this degree of satisfaction the “weight” of the clause.

Suppose that  $d = |x_0 - v(x_i)|$  is the semantic distance from the value  $x_0$  to  $v(x_i)$ . The smaller  $d$  is, the closer the distance between  $x_0$  and  $v(x_i)$  is, the greater the weight of the clause  $x = x_i$ . If  $d = 0$ ,  $w = 1$ . The bigger the  $w$  rule is, the bigger the conclusion for the output will be.

**Definition 1.** The semantic weighting distance function of the clause  $w_i, w_j$  :

$$x_0 \in [v(x_i), v(x_j)] \rightarrow [0, 1].$$

Suppose  $x_0$  is a semantic value

1)  $x_0 \in [v(x_i), v(x_j)]$ , the semantic weighting distance of the clause  $x = x_i$  is

$$w_i = 1 - |x_0 - v(x_i)| / |v(x_j) - v(x_i)|. \quad (2)$$

And its of clause  $x = x_j$  is

$$w_j = 1 - |x_0 - v(x_j)| / |v(x_j) - v(x_i)|. \quad (3)$$

2)  $x_0 \in [0, v(x_i)]$ , the semantic weighting distance of the clause  $x = x_i$  is

$$w_i = 1 - |x_0 - v(x_i)| / (v(x_i)). \quad (4)$$

3)  $x_0 \in [v(x_j), 1]$ , the semantic weighting distance of the clause  $x = x_j$  is

$$w_j = 1 - |x_0 - v(x_j)| / (1 - v(x_j)). \quad (5)$$

The output semantic value of the rule is computed by  $y = W_r * v(y_j)$ , where  $W_r$  is the weight of the rule or the degree of satisfaction of the left member in the rule.

For the rule *SISO*: If  $x = x_i$  then  $y = y_j$

Through the semantic value  $x_0$  of the input variable, we can compute the weight  $w_i$ , which represents the degree of satisfaction of the clause (with the conditional clause it is the weight of  $W_r$ ). Output:  $y = w_i * v(y_j)$ .

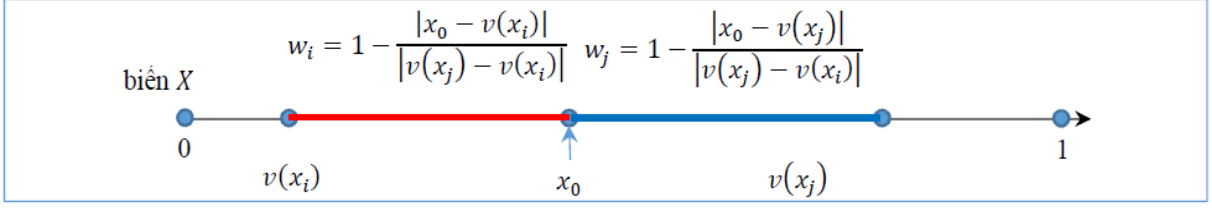


Figure 4. Degree of satisfaction of the clause

For the rule MISO: If  $x_1 = x_{1i}$  and  $x_2 = x_{2i} \dots$  and  $x_m = x_{mi}$  then  $y = y_j$

Through the input vector  $(x_{01}, x_{02}, \dots, x_{0m})$ , we can compute the weighting vector  $(w_1, w_2, \dots, w_m)$  (Figure 5). In view of the fact that the output conclusion does not exceed the input condition (or the weight of the concluding statement is no greater than the weight of the conditional clauses). Thus we can:

- Use “and = min”, we have  $W_r = \min(w_1, w_2, \dots, w_m)$  or
- Use “and = prod”, we have  $W_r = (w_1 * w_2 * \dots * w_m)$ .

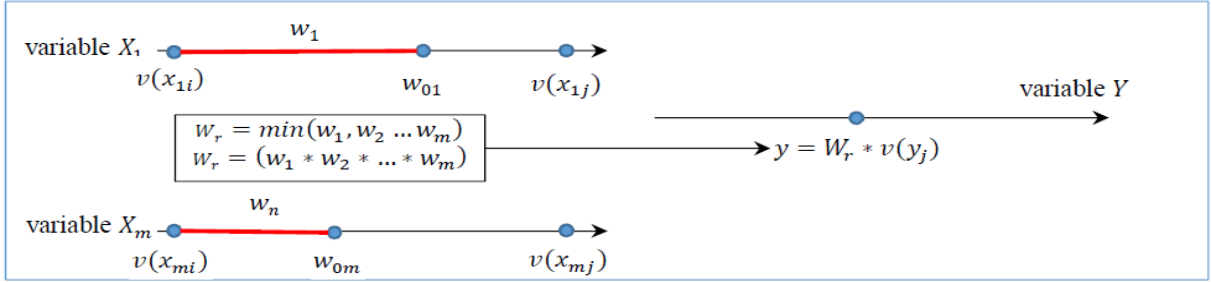


Figure 5. Reasoning implementation of the rule MISO

When there are many degrees of satisfying more than zero, the output  $y^*$  is calculated by

$$y^* = \frac{(W_{1r} * y_1 + W_{2r} * y_2 + \dots + W_{nr} * y_n)}{W_{1r} + W_{2r} + \dots + W_{nr}}. \quad (6)$$

In case QRBS includes of 2 input components, 1 output component is represented by the  $S_{real}^3$ . Computing interpolation/ extrapolation is demonstrated as follows.

- In Figure 6a it is the interpolation:

$$\text{Given } S = (e_2 - e_1)(ce_2 - ce_1),$$

$$W_{11} = \left(1 - \frac{(e_0 - e_1)}{(e_2 - e_1)}\right) \left(1 - \frac{(ce_0 - ce_1)}{(ce_2 - ce_1)}\right) = \frac{(e_2 - e_0)(ce_2 - ce_0)}{(e_2 - e_1)(ce_2 - ce_1)} = \frac{S_{11}}{S},$$

$$W_{12} = \left(1 - \frac{(e_2 - e_0)}{(e_2 - e_1)}\right) \left(1 - \frac{(ce_0 - ce_1)}{(ce_2 - ce_1)}\right) = \frac{(e_0 - e_1)(ce_2 - ce_0)}{(e_2 - e_1)(ce_2 - ce_1)} = \frac{S_{12}}{S},$$

$$W_{21} = \left(1 - \frac{(e_0 - e_1)}{(e_2 - e_1)}\right) \left(1 - \frac{(ce_2 - ce_0)}{(ce_2 - ce_1)}\right) = \frac{(e_2 - e_0)(ce_0 - ce_1)}{(e_2 - e_1)(ce_2 - ce_1)} = \frac{S_{21}}{S},$$

$$W_{22} = \left(1 - \frac{(e_2 - e_0)}{(e_2 - e_1)}\right) \left(1 - \frac{(ce_2 - ce_0)}{(ce_2 - ce_1)}\right) = \frac{(e_0 - e_1)(ce_0 - ce_1)}{(e_2 - e_1)(ce_2 - ce_1)} = \frac{S_{22}}{S},$$

$$u = \frac{W_{11}u_{11} + W_{12}u_{12} + W_{21}u_{21} + W_{22}u_{22}}{W_{11} + W_{12} + W_{21} + W_{22}} = \frac{S_{11}u_{11} + S_{12}u_{12} + S_{21}u_{21} + S_{22}u_{22}}{S}.$$

- In Figure 6b:

$$\text{Given } S = (e_2 - e_1)ce_1,$$

$$W_{11} = \left(1 - \frac{(e_0 - e_1)}{(e_2 - e_1)}\right) \left(1 - \frac{(ce_1 - ce_0)}{ce_1}\right) = \frac{(e_2 - e_0)ce_0}{(e_2 - e_1)ce_1} = \frac{S_{11}}{S},$$

$$W_{12} = \left(1 - \frac{(e_2 - e_0)}{(e_2 - e_1)}\right) \left(1 - \frac{(ce_1 - ce_0)}{ce_1}\right) = \frac{(e_0 - e_1)ce_0}{(e_2 - e_1)ce_1} = \frac{S_{12}}{S},$$

$$u = \frac{W_{11}u_{11} + W_{12}u_{12}}{W_{11} + W_{12}} = \frac{S_{11}u_{11} + S_{12}u_{12}}{S_{11} + S_{12}}.$$

- In Figure 6c:

$$\text{Given } S = e_1(ce_2 - ce_1),$$

$$W_{11} = \left(1 - \frac{(e_1 - e_0)}{e_1}\right) \left(1 - \frac{(ce_0 - ce_1)}{(ce_2 - ce_1)}\right) = \frac{e_0(ce_2 - ce_0)}{e_1(ce_2 - ce_1)} = \frac{S_{11}}{S},$$

$$W_{21} = \left(1 - \frac{(e_1 - e_0)}{e_1}\right) \left(1 - \frac{(ce_2 - ce_0)}{(ce_2 - ce_1)}\right) = \frac{e_0(ce_0 - ce_1)}{e_1(ce_2 - ce_1)} = \frac{S_{21}}{S},$$

$$u = \frac{W_{11}u_{11} + W_{21}u_{21}}{W_{11} + W_{21}} = \frac{S_{11}u_{11} + S_{21}u_{21}}{S_{11} + S_{21}}.$$

- In Figure 6d:

$$\text{Given } S = e_1ce_1,$$

$$W_{11} = \left(1 - \frac{(e_1 - e_0)}{e_1}\right) \left(1 - \frac{(ce_1 - ce_0)}{ce_1}\right) = \frac{e_0ce_0}{e_1ce_1} = \frac{S_{11}}{S},$$

$$u = \frac{W_{11}u_{11}}{W_{11}} = \frac{S_{11}u_{11}}{S_{11}} = u_{11}.$$

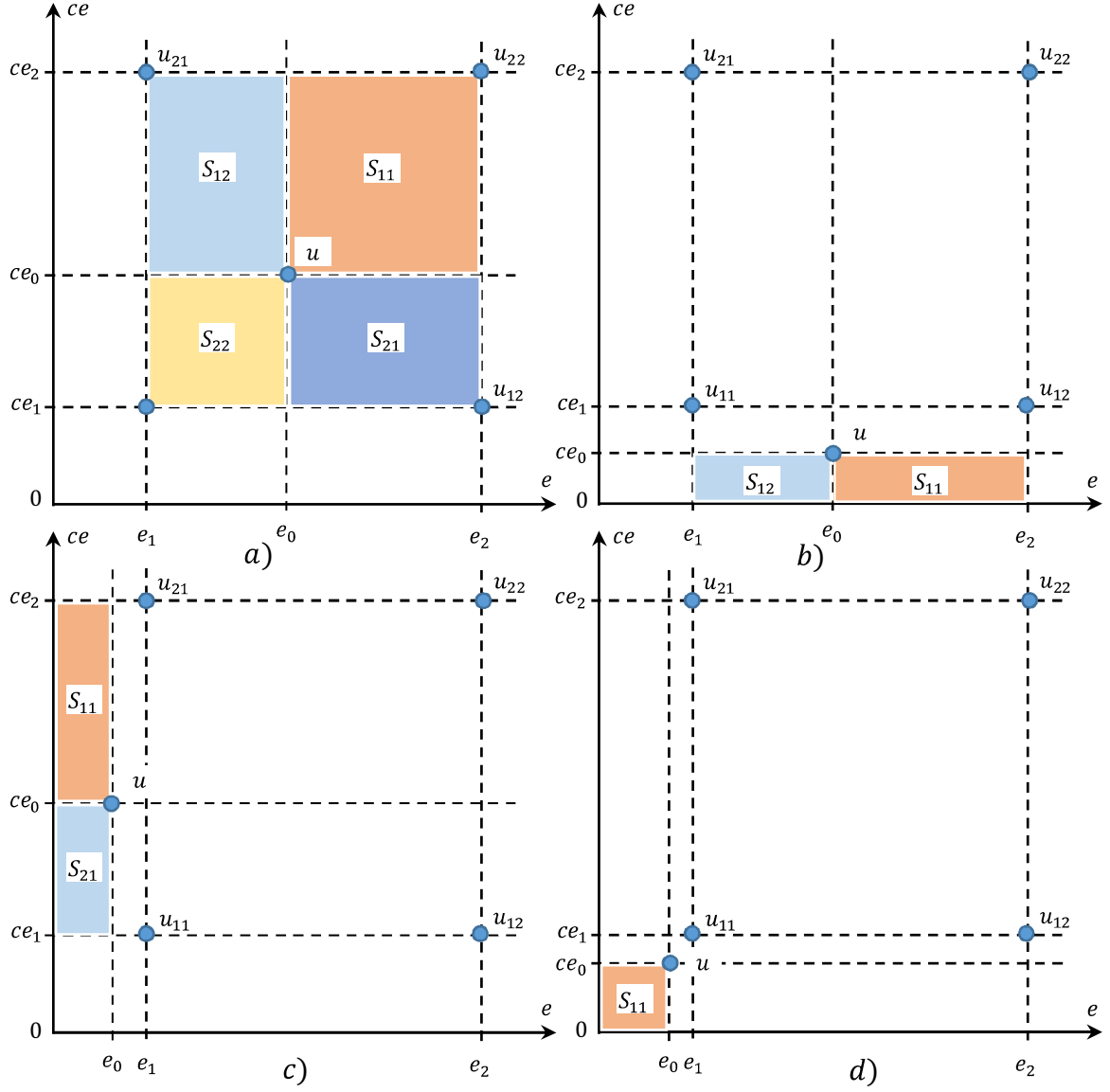


Figure 6. Interpolation/extrapolation on  $S_{real}^3$  based on semantic weighting distance

Back to the problem with QRBS as in Table 1 and input vectors  $v_1 (e_1 = 0.1, ce_1 = 0.1) < v_2 (e_2 = 0.125, ce_2 = 0.125)$ , using the 'prod' for 'and' operator, we can compute the interpolation value on the semantic distance weight.

- With the input vector  $v_1$ , the interpolation steps are computed as follows:

$$+ \text{Apply (4), } w_{1e} = w_{1ce} = 1 - \frac{|0.1 - 0.125|}{|0.125|} = 0.8,$$

$$+ W_r = w_{1e} * w_{1ce} = 0.64,$$



+ The output,  $y = \frac{0.64 * 0.125}{0.64} = 0.125$ .

• For the input vector  $v_2$ :

+ Apply (2) and (3):  $w_{2e} = w_{2ce} = 1 - \frac{|0.125 - 0.125|}{|0.375 - 0.125|} = 1$ ,

$$w_{3e} = w_{3ce} = 1 - \frac{|0.125 - 0.375|}{|0.375 - 0.125|} = 0,$$

+  $W_r = w_{2e} * w_{2ce} = 1$ ,

+ The output is calculated according to (6):  $y^* = \frac{1 * 0.125}{1} = 0.125$ .

### 3. APPLICATION PROBLEM

#### 3.1. Introduction to the resistor heaters

Heating equipment is an object which is widely used in industry, medicine and engineering civil. In the industry it is often used for heat treatment, melting ferrous and non-ferrous metals. Industrial furnaces often use metal wire. In other areas such as medical or civil, they mainly focus on kilns. Temperature is the quantity that needs to be adjusted. Controlling of the furnace temperature control is usually done by controlling the power supply.

Consider a resistance furnace with a power  $P = 1KW$ . Heating bar SiC, with a temperature range of  $25 - 250^\circ C$ . The furnace has an approximate transfer function (received through process of object recognition), which is the most inertial step of delay:

$$W(s) = K \frac{e^{-\tau s}}{1 + Ts}, \quad (7)$$

where,

Amplification coefficient  $K = 10^\circ C$ ,

Constant time (second)  $T = 1300s$ ,

Time delay (second)  $\tau = 30s$ .

$$W(s) = \frac{10e^{-30s}}{1 + 1300s}. \quad (8)$$

#### 3.2. Design of controller used hedge algebra

In this section, we apply a new method of interpolation based on the semantic distance weights described in Section 2. The object to be controlled is the thermocouple furnace. The controller here is the approximation set that the control rule is given by the LRBS. The structure of the controller is shown in Figure 1. The controller design steps are as follows.

**Step 1:** Identify input/output variables, their variation domains, and control rules with language classes in HA. Through the LRBS, we define the input/output variables and their variance domain is determined by the survey as follows:

- The controller inputs have two variables:  
 $e$  (*error*) – control error, variance in range  $[-e_{\max}, +e_{\max}] = [-4, 4]$ .  
 $ce$  (*changeerror*) – indicates the variable speed of  $e$ . Variation in the range  
 $[-ce_{\max}, +ce_{\max}] = [-50, 50]$ .
- The output of the controller is the control quantity  $u$  to control the voltage of the power source, varying in the range  $[-u_{\max}, +u_{\max}] = [-4.5, 4.5]$ .

The control rule is an LRBS given in Table 2.

Table 2. LRBS of the controller

$e \backslash ce$	$VN$	$LN$	$ZE$	$LP$	$VP$
$VN$	$VN$	$VN$	$N$	$LN$	$ZE$
$LN$	$VN$	$N$	$LN$	$ZE$	$LP$
$ZE$	$N$	$LN$	$ZE$	$LP$	$P$
$LP$	$LN$	$ZE$	$LP$	$P$	$VP$
$VP$	$ZE$	$LP$	$P$	$VP$	$VP$

The input/output language variables include the following language values:

$$e, ce = \{VN < LN < ZE < LP < VP\}; u = \{VN < N < LN < ZE < LP < P < LP\}$$

where,  $VN = \text{VeryNegative}$ ,  $LN = \text{LittleNegative}$ ,  $ZE = \text{Zero}$ ,  $LP = \text{LittlePositive}$ , ...

Rules in the table are understood as follows:

If  $e = VN$  and  $ce = VN$  then  $u = VN$ ; If  $e = VN$  and  $ce = LN$  then  $u = VN$ ; ...

**Step 2:** Select the structures  $\mathcal{AX}_{e,ce}$  and  $\mathcal{AY}$  for the variables  $\mathcal{X}_i$  and  $\mathcal{Y}$ . Determine the fuzzy parameter of the generating elements and the hedges.

- Set of generating elements  $G = \{N < P\}$ ; Set of hedges choosen:  $H^- = \{L\}$  and  $H^+ = \{V\}$ .
- According to the HA structure for the variables constructed above, we need to choose the fuzzy measure of the negative generating elements  $fm(c^-) = fm(N)$  ( $fm(c^+) = 1 - fm(c^-) = fm(P) = 1 - fm(N)$ ) and the fuzzy measurement of negative hedges  $\alpha = \mu(L)$  ( $\beta = \mu(V) = 1 - \alpha$ ). The fuzzy parameters are initially chosen as intuitively as in Table 3.

Table 3. Fuzzy parameter of HAs

	$e$	$ce$	$u$
$fm(N)$	0.5	0.5	0.5244
$\alpha = \mu(L)$	0.5	0.5	0.5

- The sign of the generating elements, the hedges and the sign relationship between the hedges are determined based on the semantic nature of the language terms. For example, we have  $sgn(N) = -1$ ,  $sgn(P) = 1$ . Also, it can be seen that  $VVN < VN \Rightarrow sgn(V, V) = 1$ .  $LVN > VN \Rightarrow sgn(L, V) = -1$ . Similarly, for other linguistic elements, we define the sign relation as in Table 4.

Table 4. Sign relation

	$V$	$L$	$N$	$P$
$V$	+	+	-	+
$L$	-	-	+	-

**Step 3:** Compute semantically quantifying values for language labels in the rule set. Build up the input/output relationship in semantic space  $S_{real}^3$ .

With fuzzy parameters selected as in Table 3 and the sign relationship between hedges, between the hedges and the generating elements as in Table 4, use semantically quantifying mapping function SQMs [8], we can compute the semantically quantifying value of the language terms in the rule table as in Table 5.

Table 5. QRBS of the controller

$e \backslash ce$	0.125	0.375	0.5000	0.625	0.875
0.125	0.1311	0.1311	0.2622	0.3933	0.5244
0.375	0.1311	0.2622	0.3933	0.5244	0.6433
0.500	0.2622	0.3933	0.5244	0.6433	0.7622
0.625	0.3933	0.5244	0.6433	0.7622	0.8811
0.875	0.5244	0.6433	0.7622	0.8811	0.8811

Figure 7 is  $S_{real}^3$  corresponding to Table 5. That is the mathematical model that represents the input/output relationship of the controller.

**Step 4:** Select interpolative method: When performing the simulation, we proceed on both bi-linear interpolation and interpolation based on semantic weighting distance. Figure 8 is a simulation model of the system in the Matlab/Simulink environment.

#### 4. COMPUTING RESULT AND SIMULATION

Let  $e, ce$  be variable from 0 to 1,  $step = 0.05$ , apply bi-linear interpolation and interpolate based on semantic weighting distance with  $\text{and} = \text{prod}$ , we get the corresponding interpolation on Figure 9.

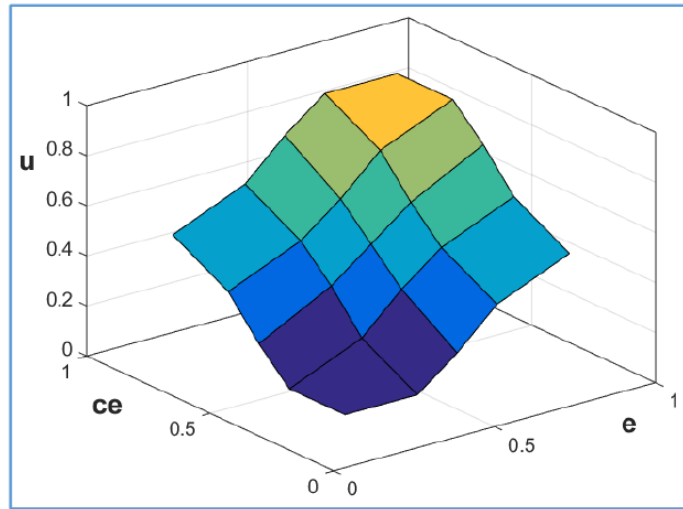


Figure 7. Input/output relationship surface  $S_{real}^3$  in semantic space

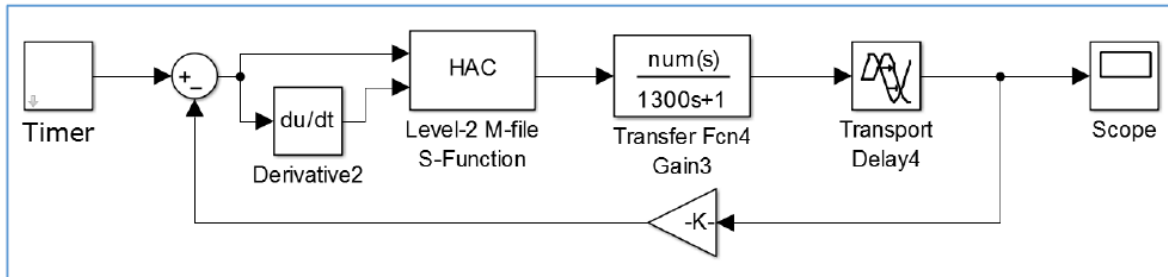


Figure 8. Model simulation of the system

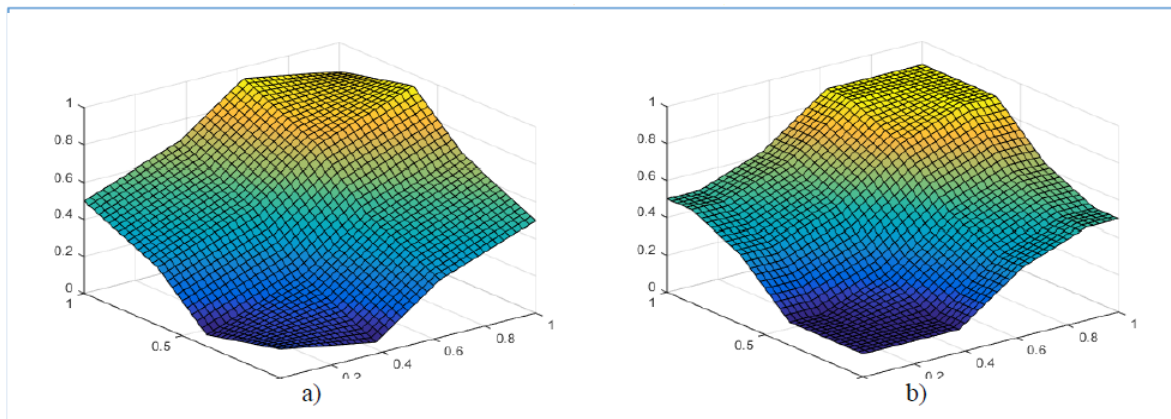


Figure 9. a) Interpolation by bi-linear interpolation      b) Weighted interpolation distances

Based on Figure 9, we find that both interpolation methods give the variable interpolation

homogeneous with the semantic surface  $S_{real}^3$  (Figure 7). On Figure 9a, when the input values  $e < -e_{max}$ ,  $ce < -ce_{max}$  or  $e > +e_{max}$ ,  $ce > +ce_{max}$ , the extrapolation value is incorrect according to the variable rule of the rule set. When deviations  $e$  and  $ce$  are large, the controller should give a large value  $u$ . However, according to bi-linear interpolation, the value of  $u$  is small. This is opposed to the control objective. With the method of interpolation based on semantic weighting distance when extrapolation has overcome this limitation (Figure 8b). This is due to the addition of constraints (Definition 1) that allow for normal bi-linear interpolation to the process of appropriate inference for LRBS.

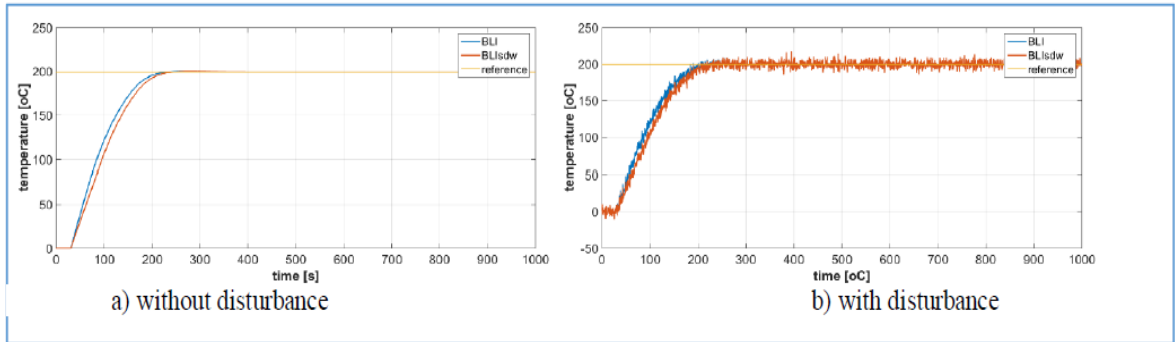


Figure 10. Output response when simulating with the controller

Simulating the system with  $time = 1000s$ , the reference value is  $t_{ref} = 200^{\circ}C$ , when there is no interference and there is no load disturbance at the output (interference amplitude  $N = 5\%$  of the reference value). We get the result shown in Figure 10. The line BLI is the response of the controller to the bi-linear interpolation, BLIsdw is the response of the controller to the interpolation method based on the semantic weighting distance. Observing the responses on the graph, we can see that even if there is no interference or interference at the output of the system, the HAC controller always responds well in terms of response time, over-tuning and time accuracy which is set up in both methods of interpolation. In the same condition, the response of the controller with interpolation based on semantic distance weights is a bit better than bi-linear interpolation. It can be explained that the difference in the calculated values between the two methods is only in the range of extrapolation, where

Table 6. Simulation results

	Bi-linear interpolation		Interpolation based on semantic distance weights	
	<i>Without Noise</i>	Noise	<i>Without Noise</i>	Noise
Rise time [s]	172.6	163	161.087	151
Overshoot [%]	0.196%	0.021%	0.2%	0.021%
Settling time [%]	172.6	163	161.1	151
ISE	2.4021e+03	2418.2	2177.4	2.1937e+03

$e$  and  $ce$  are large. When  $e$  and  $ce$  have entered the interpolation area of surface  $S_{real}^3$ , the interpolation values of the two methods are the same.

Some computational values are summarized in Table 6. Numerical results also show that interpolation based on semantic distance weights gives smaller values than normal bi-linear interpolation.

## 5. CONCLUSION

In this paper, we have proposed the use of interpolation based on the semantic distance weight of the input semantic values versus the semantic values of the terms appearing in the rule. Specifically, we introduce constraints for the two-dimensional linear interpolation method (Definition 1). Under the limited constraints in this definition, the extrapolation value only depends on the semantic values of the adjacent elements in the LRBS. Thus weighted output values are consistent with human reasoning based on the rule of the linguistic element. We also carried out the design and applied this interpolation method in controllers whose control rules were given by LRBS for resistor furnace. The simulated results are compared and evaluated using normal bi-linear interpolation. The output response when the controller works with the interpolation method based on the semantic distance weight is better than the two-dimensional linear interpolation method. The quality of control is not very great, as it only occurs in a few cases when extrapolating but it demonstrates the correctness of the method when simulating LRBS's variation rules and approximately reasoning processes. In the near future, we will test this proposal for controllers with LRBS and more complex objects to confirm the effectiveness of this method.

## REFERENCES

- [1] N. D. Anh, B. H. Le, V. N. Lan, and T. D. Trung, "Application of hedge algebra-based fuzzy controller to active control of a structure against earthquake," *Structural Control and Health Monitoring*, vol. 20, pp. 483–495, 2013. [Online]. Available: DOI:10.1002/stc.508.
- [2] N. D. Duc, V. N. Lan, T. D. Trung, and B. H. Le, "A study on the application of hedge algebras to active fuzzy control of a seism-excited structure," *Journal of Vibration and Control*, vol. 18, no. 14, pp. 2186–2200, 2011.
- [3] N. C. Ho and V. N. Lan, "Hedge algebras: An algebraic approach to domains of linguistic variables and their applicability," *ASEAN Journal on Science & Technology for Development*, vol. 23, no. 1&2, pp. 1–18, 2006.
- [4] N. C. Ho, V. N. Lan, and N. D. Anh, "Fuzzy controllers using hedge algebra based semantics of vague linguistic terms," *Fuzzy Control Systems*, pp. 135–192, 2013.
- [5] N. C. Ho, V. N. Lan, N. T. Duy, and P. V. Thiem, "Study the ability of replacing fuzzy and pi controllers with the hedge - algebras - based controller for dc motor," *Journal of Science and Technology*, vol. 52, no. 1, pp. 35–48, 2014.
- [6] N. C. Ho, V. N. Lan, and L. X. Viet, "Optimal hedge-algebras-based controller: Design and application," *Fuzzy Sets and Systems*, vol. 159, pp. 968–989, 2008.

- [7] —, “Optimal hedge-algebras-based controller: Design and application,” *Fuzzy Sets and Systems*, vol. 159, pp. 968–989, 2008.
- [8] N. C. Ho and N. V. Long, “Fuzziness measure on complete hedge algebras and quantifying semantics of terms in linear hedge algebras,” *Fuzzy Sets and Systems*, vol. 158, no. 4, pp. 452–471, 2007.
- [9] N. C. Ho and W. Wechler, “Hedge algebra: An algebraic approach to structures of sets of linguistic truth values,” *Fuzzy Sets and Systems*, vol. 35, pp. 281–293, 1990.
- [10] —, “Extended hedge algebras and their application to fuzzy logic,” *Fuzzy Sets and Systems*, vol. 52, pp. 259–281, 1992.
- [11] B. L. Hoang, F. Luo, N. T. Duy, and N. H. Phuong, “Dissolved oxygen control of the activated sludge wastewater treatment process using hedge algebras control,” *The 2014 7<sup>th</sup> International Congress on Image and Signal Processing*, pp. 1298–1303, 2014, 14-16 October, Daian, China.
- [12] B. H. Le, N. C. Ho, V. N. Lan, and N. C. Hung, “General design method of hedge-algebras-based fuzzy controllers and an application for structural active control,” *Applied Intelligence*, vol. 43, no. 2, pp. 251–275, 2015.
- [13] —, “General design method of hedge-algebras-based fuzzy controllers and an application for structural active control,” *Applied Intelligence*, vol. 43, no. 2, pp. 251–275, 2015. [Online]. Available: DOI10.1007/s10489-014-0638-6
- [14] B. H. Le, T. D. Trung, and V. N. Lan, “Optimal fuzzy control of an inverted pendulum,” *Journal of Vibration and Control*, vol. 18, no. 14, pp. 2097–2110, 2011.
- [15] —, “Optimal fuzzy control of an inverted pendulum,” *Journal of Vibration and Control*, vol. 18, no. 14, pp. 2097–2110, 2011.
- [16] D. Vukadinovic, M. Basic, N. C. Ho, V. N. Lan, and N. T. Duy, “Hedge-algebra-based voltage controller for a self-excited induction generator,” *Control Engineering Practice*, vol. 30, no. 1, pp. 78–908, 2014.

*Received on May 20, 2017*

*Revised on July 30, 2017*