Hindawi
Journal of Control Science and Engineering
Volume 2019, Article ID 3590527, 17 pages
https://doi.org/10.1155/2019/3590527

Research Article

Hedge-Algebra-Based Phase-Locked Loop for Distorted Utility Conditions

Dinko Vukadinović ,[1] Tien Duy Nguyen,[2] Cat Ho Nguyen,[3] Nhu Lan Vu,[4] Mateo Bašić,[1] and Ivan Grgić[1]

[1] *Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Ruđera Boškovića 32, 21000 Split, Croatia*
[2] *Faculty of Electronics Engineering, Thai Nguyen University of Technology, 666, 3/2 street, Tich Luong ward, Thai Nguyen, Vietnam*
[3] *Institute of Theoretical and Applied Research, Duy Tan University, Quang Trung 03, Da Nang, Vietnam*
[4] *Faculty of Mathematics and Informatics, Thang Long University, Nghiem Xuan Yem Street, Hoang Mai District, Hanoi, Vietnam*

Correspondence should be addressed to Dinko Vukadinović; dvukad@fesb.hr

Received 25 October 2018; Accepted 10 January 2019; Published 3 March 2019

Academic Editor: Radek Matušů

This paper presents the first application of the hedge-algebra theory in the field of grid synchronization. For this purpose, an optimized hedge-algebra controller (HAC) is developed and incorporated within the three-phase phase-locked loop (PLL) with moving average filters (MAFs) inside its feedback loop. Optimized fuzziness parameters and linguistic rule base of the HAC are obtained by a genetic algorithm using the integral of absolute error as the performance index during optimization. Calculated optimal parameter values of the HAC depend on the most frequently occurring disturbance in the electric grid. Two different PLL structures are proposed, depending on the types of disturbances occurring in the electric grid. The first structure is the conventional synchronous reference frame PLL with the nonadaptive MAF (i.e., MAF without order adjustment), but with the PI/PID controller in the phase loop replaced by the developed HAC. Such PLL structure is suitable for all analyzed disturbance types, expect for step-changes in the grid frequency. The second PLL structure introduces the adaptive MAF (i.e., MAF with order adjustment) and a new feedback signal in the output stage of the controller to achieve zero steady-state error in the case of step-changes in the grid frequency. The disturbance rejection capability of the two developed PLLs with the HAC (HAC-PLLs) is tested separately and compared experimentally with the PID- and fuzzy-controller-based PLLs.

1. Introduction

The synchronization of power electronics converters with a distributive electric grid presents an intensive research area in the world. In three-phase systems, phase-locked loops (PLLs) present one of the most used synchronization methods. The conventional synchronous reference frame PLL (SRF-PLL) is shown in Figure 1.

The main parts of the conventional SRF-PLL are as follows: the phase detector, the low-pass filter, and the voltage controlled oscillator. The phase detector transforms the phase voltages u_a, u_b, u_c to synchronous direct and quadrature voltage components u_d and u_q, respectively. The direct voltage component u_d is equal to the phase voltages amplitude U for nondistorted phase voltages. Usually, the filter in Figure 1 is realized as the PI or PID controller, with the zero-voltage reference. Hence, the term controller is used below. The SRF-PLL outputs the estimated phase angle of the three-phase system $\hat{\vartheta}$. This type of PLL shows its disadvantages in the case of an unbalanced phase voltage system, i.e., when there are negative- and zero-sequence components along with a positive-sequence component. In this case, a disturbance component whose frequency is equal to twice the fundamental grid frequency (e.g., 100 Hz when the nominal value is 50 Hz) appears in the direct and quadrature voltage components u_d and u_q. Consequently, a power electronics inverter, which performs DC to AC inversion, is not properly synchronized with the electric grid.

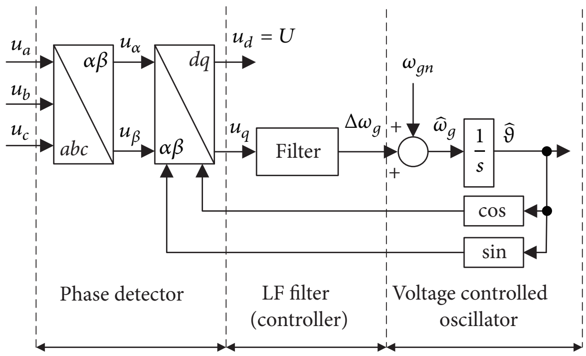

Figure 1: Block diagram of the conventional SRF-PLL.

To deal with unbalanced phase voltage systems, many PLLs with added filters have been proposed [1]. The PLLs with filters in the phase loop can be classified as follows: moving average filter-based PLL (MAF-PLL) [2–4], cascaded delayed signal-cancellation (CDSC) operator [5], notch filter-based PLL [2, 6], variable sampling period filter-based PLL [7], multiple reference frame-based PLL [8], and in-loop complex-coefficient filter-based PLL [9].

On the other hand, the PLLs with filters outside of the phase loop are often called PLLs with a prefiltering stage. This prefiltering stage significantly attenuates disturbance components before the phase loop input. Some of the PLLs with the prefiltering stage are the PLL with a multiple complex-coefficient filter [10], the PLL with the recursive discrete Fourier transform [11], the PLL with the sliding Goertzel discrete Fourier transform [12], the PLL with the moving average filter-based (MAF) prefiltering [13], the PLL with the dual second-order generalized integrators [14], and the delay signal cancelation-based PLL [5, 15].

Scientific papers dealing with the PLLs in which artificial intelligence is applied are extremely rare. In review paper [1], there is no single example of artificial intelligence use in PLLs. In [16–18] the fuzzy logic theory is applied in the PLL. In [16], the particle swarm optimization algorithm is used to tune values of the fuzzy controller (FC) membership functions. In [17], the authors propose application of the same algorithm for both tuning of the FC membership functions and finding the optimal values of the low-pass filter. It is very important to point out that, in these papers, there is a nonzero phase error when unbalanced grid conditions appear. In [18], a FC in the three-phase PLL is applied, but only with the phase voltages containing harmonics or a DC component. The phase error is not shown in this paper.

Generally, advantages that FCs have in nonlinear control systems over the classical PI/PID controller are numerous. For example, FCs do not require knowledge of a detailed mathematical model of the control system and also they are expressed in linguistic terms used in natural language. In addition, FCs have the capability of handling noisy signals and usually lead to better results compared to the classical PI/PID controller in terms of the control system performance [19].

Hedge-algebra theory was first developed in 1990 [20] to model the order-based semantics of the terms in term-domains of linguistic variables. Hedge algebras form a formalism to immediately handle linguistic words and linguistic rule bases (LRBs) and their computational (compt.-) semantics instead of their fuzzy sets or fuzzy set expressions representing their inherent semantics. So, this theory is based on mapping of a few fuzziness parameters of each linguistic variable instead of using fuzzy sets. In [21], the authors point out that the structure of the hedge-algebra-based controller (HAC) is like the conventional FC, but it is simpler and more convenient. As a result, the HAC was applied in optimal fuzzy control of an inverted pendulum [22], in active vibration control of building structures subjected to seismic excitations [23–26], and in voltage control of a self-excited induction generator [27]. However, it was not yet applied in the field of grid synchronization.

This paper introduces three-phase PLLs with the MAF inside the phase loop and the HAC used as a controller. The first proposed PLL is based on the MAF-PLL reported in [3], but with the PI/PID controller replaced by the HAC. For this purpose, the MAF with no order adjustment (nonadaptive MAF) is proposed. This PLL is suitable for all analyzed disturbance types, expect for step-changes in the grid frequency. The second PLL structure introduces the adaptive MAF (i.e., MAF with order adjustment) and a new feedback signal in the output stage of the controller to achieve zero steady-state error in the component u_q, in the case of step-changes in the grid frequency. For this PLL type, an application of the MAF is proposed with its order adjusted according to the linear interpolation method [4]. Both types of the PLLs are briefly named HAC-PLL. In addition, it is proposed that both the structure and parameters of the HAC-PLL are determined depending on the most frequently occurring disturbance in the electric grid. This approach requires that the power quality monitoring is carried out over one week [28]. Optimized fuzziness parameters and linguistic rule base of the HAC are obtained for each considered disturbance by means of a genetic algorithm. The integral of absolute error (IAE) is used as a performance index for finding these parameters and rules.

In this paper, the following disturbances are considered: unbalanced voltage sag with/without harmonics distortion of the phase voltages, unbalanced low-frequency transients in the phase voltages, and step-changes in the grid frequency. It is important to point out that the HAC-PLL with parameters and linguistic rules optimized for a specific disturbance works stable with other analyzed disturbances as well. Finally, a comparison is shown between the HAC-PLL and two other MAF-PLLs: with the PID controller and with the FC.

2. Phase-Locked Loops with the Moving Average Filter

A MAF belongs to the group of finite impulse response filters which eliminate all harmonics whose frequency is multiplier of the reciprocal value of its window width. The MAFs are suitable for embedded computer systems because of their

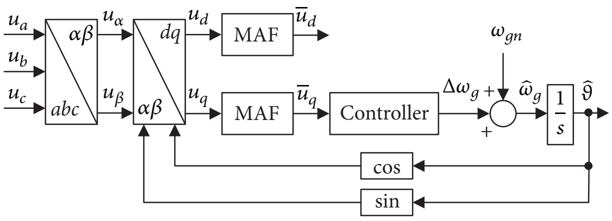

Figure 2: PLL with the nonadaptive MAF and a controller.

low computational requirements. On the other hand, their filtering capability decreases when the ratio of the window width and the sampling period is not an integer. In that case, the window width needs to be varied and adapted to the circumstances. Such MAFs are called hereinafter *the adaptive MAFs*, whereas *MAF* only refers to the nonadaptive type, unless otherwise indicated.

Figure 2 shows the PLL with the nonadaptive MAF and a controller inside the phase loop [3]. The controller can be of a PID, fuzzy or HAC type, as described later.

The MAF with the input signal x and the output signal $\overline{x}$ can be described by the following equations in the continuous-time and discrete-time domain, respectively:

$$\overline{x}(t) = \frac{1}{T_w}\int_{t-T_w}^{T_w} x(t)\,dt \tag{1}$$

$$\overline{x}(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(k-n) \tag{2}$$

where $T_w = NT_s$ is the MAF window width, T_s is the sampling time, and N is the number of samples within the window width.

Using (1) and (2), the transfer functions in the s-domain and z-domain, respectively, are defined as

$$G(s) = \frac{\overline{x}(s)}{x(s)} = \frac{1-e^{-T_w s}}{T_w s} \tag{3}$$

$$G(z) = \frac{\overline{x}(z)}{x(z)} = \frac{1}{N}\sum_{n=0}^{N-1} z^{-N} = \frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}}. \tag{4}$$

By replacing s with $j\omega$ in (3), the magnitude and phase characteristics in the s-domain are given as

$$G(j\omega) = \left|\frac{\sin(\omega T_w/2)}{\omega T_w/2}\right| e^{-j\omega T_w/2} \tag{5}$$

Similarly, by replacing z with $e^{j\omega T_s}$ in (5), the magnitude and phase characteristics in the z-domain are given as

$$G\left(e^{j\omega T_s}\right) = \left|\frac{\sin(\omega N T_s/2)}{N\sin(\omega T_s/2)}\right| e^{-j\omega(N-1)T_s/2} \tag{6}$$

Equation (5) shows that the MAF input signals whose angular frequency is a multiple of the reciprocal window width $1/T_w$ ($f = n/T_w$, $n = 1, 2, 3, \ldots$) will be completely blocked. So, it is often said that the MAF acts as an ideal low-pass filter. In addition, (6) takes on the form of (5) if the sampling time $T_s \longrightarrow 0$.

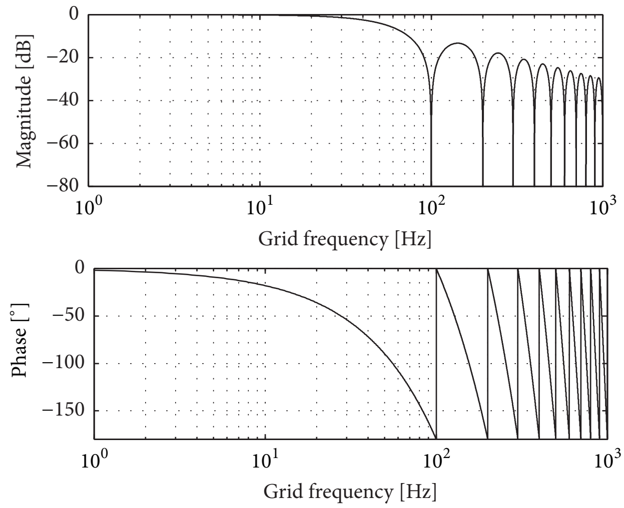

Figure 3: MAF Bode plots obtained with $T_w = 0.01$ s and $N = 100$.

As mentioned before, the analyzed disturbances in the phase voltage quadrature component u_q include a harmonic component whose frequency is 100 Hz when the grid frequency is 50 Hz. So, it is chosen $T_w = 0.01$ s to completely remove the 100 Hz harmonic and all its multiples from u_q. This selection for the MAF window width is recommended when the grid harmonic content is unknown and the DC-offset may be present in the phase voltages. To determine the sampling time T_s, the fastest time-varying disturbance needs to be considered. In this paper, low-frequency transients in the phase voltages are considered, with an oscillation frequency less than 5 kHz, as the fastest time-varying disturbance [29]. The sampling time of 100 μs is, in this sense, sufficient, whereas the number of samples within the window width N is equal to 100.

Figure 3 shows the Bode plots of the MAF obtained using (5) and with $T_w = 0.01$ s and $N = 100$.

Equation (4) presents the nonadaptive MAF because its parameter N is constant. Such MAF is suitable when no significant frequency variation of the electric grid is expected. However, the problem arises when the grid frequency varies. In such cases, the nonadaptive MAF does not completely block the harmonic whose frequency is equal to twice the changed grid frequency. Therefore, some of the MAF variants with the variable parameter N should be applied [3]. The best filtering capabilities are achieved by using MAFs with the linear interpolation method [3, 4]. Using this approach, the output signal of the MAF in the k^{th} sample instant $\overline{x}(k)$ is defined as

$$\overline{x}(k) = \frac{T_s}{T_w}\left(\sum_{i=0}^{N_f-1} x(k-i) + \alpha\left((1-\alpha)\,x\left(k-N_f+1\right) + \alpha x\left(k-N_f\right)\right)\right) \tag{7}$$

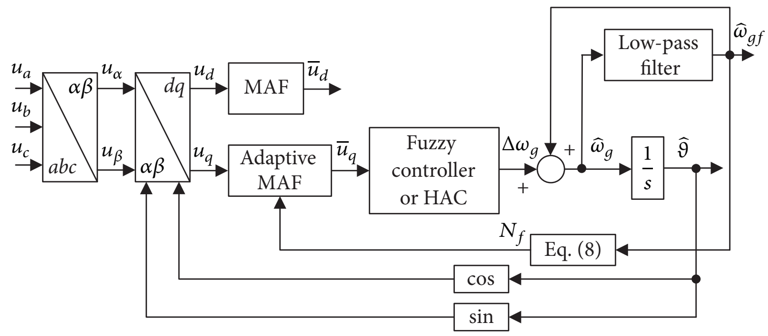

Figure 4: PLL for rapid changes in grid frequency with the fuzzy controller or the HAC and the adaptive MAF.

where N_f is the greatest integer less than or equal to T_w/T_s and $\alpha = T_w/T_s - N_f$.

The parameter N_f is calculated using the *floor* function as follows:

$$N_f = floor\left(\frac{1}{T_s \cdot 2\widehat{f}_{gf}}\right) = floor\left(\frac{\pi}{T_s \cdot \widehat{\omega}_{gf}}\right) \tag{8}$$

where $\widehat{\omega}_{gf}$ is the estimated angular grid frequency, filtered by a low-pass filter described in Section 2.2.

By using (7), the MAF becomes adaptive with regard to the grid frequency (i.e., the window width is variable). In this way, better filtering performance is achieved, especially for low-frequency components.

2.1. MAF-Based PLL with the PID Controller. The first considered PLL is the one with the nonadaptive MAF and PID controller inside the phase loop shown in Figure 2. In this paper, this PLL is used for performance evaluation of the developed PLL, which is described in Section 3.

The PID controller is written in the following form [3]:

$$G(s) = k_p \frac{1 + \tau_i s}{\tau_i s} \frac{1 + \tau_d s}{1 + \beta \tau_d s} \tag{9}$$

where k_p is the proportional gain, τ_i is the integral time constant, and τ_d is the derivative time constant.

The parameter $\beta = 0.1$ prevents the derivation action for the frequencies higher than $1/(\beta\tau_\mathrm{d})$. The PID parameters which ensure the phase margin of 45°, the damping factor $\zeta = 0.707$, and the natural frequency $\omega_n = 2\pi 20$ rad/s can be obtained as

$$\tau_i = \frac{2\zeta}{\omega_n},$$
$$\tau_d = \frac{T_w}{2}, \tag{10}$$
$$k_p = \frac{2\zeta\omega_n}{U_1^+}$$

where U_1^+ is the amplitude of the positive-sequence component of the phase voltage.

More details about the described MAF-based PLL with the PID controller can be found in [3].

2.2. MAF-Based PLL for Rapid Changes in Grid Frequency. In the case of rapid changes in grid frequency, the FC within the PLL structure as in Figure 2 is unable to achieve zero steady-state error in the component u_q. To deal with this problem, the new adaptive-MAF-based PLL structure with the FC or the HAC is proposed, as shown in Figure 4. In comparison with Figure 2, here it was necessary to add a new feedback signal in the output stage of the controller in order to obtain stable operation. Also, by using the adaptive MAF instead of the nonadaptive MAF it is possible to obtain lower IAE values. First, the FC is considered for the PLL in Figure 4, whereas the HAC is dealt with in more detail in Section 3.

In this study, the Mamdani fuzzy inference system is selected for the FC because it is more widely accepted and allows a more intuitive approach to design in comparison to the Sugeno type. Figure 5 shows the membership functions of the Mamdani-type FC. These functions were obtained by the trial-and-error, whereas the IAE was used as a performance index. In Figure 5, N, P, and ZE stand for *negative*, *positive,* and *zero*, respectively, whereas L, V, and VV stand for *little*, *very,* and *very very*, respectively. Hence, for example, LN is interpreted as *little negative*. In addition, e is the error signal calculated as the difference between the voltage u_q and its zero reference, ce is the change in error calculated as the difference between two consecutive values of the error signal e, and $c\Delta\omega_g$ is the adjustment signal for the change in the estimated grid frequency. Variables e and ce represent the inputs to the FC, whereas $c\Delta\omega_g$ represents the output variable in per unit notation. Such notation, however, requires scaling of the input and output signals. The scaling factors were selected according to the analyzed disturbance. For voltage sags (with and without harmonics) the scaling factors equal to 0.1, 3.5, and 85 were chosen for e, ce, and $c\Delta\omega_g$, respectively, again using the IAE as the performance index. These factors are equal to 0.25, 2.8, and 0.1 for an unbalanced low-frequency transient, and, finally, they are equal to 0.093, 4, and 105 for rapid changes in grid frequency.

The FC has 25 fuzzy rules, shown in Table 1, whereas the centroid method was selected for defuzzification.

The low-pass filter in Figure 4 outputs the feedback signal $\widehat{\omega}_{gf}$, which presents the estimated grid frequency passed through the second-order filter. This filter has the damping factor $\zeta = 0.9$ and the natural frequency $\omega_n = 2\pi 35$ rad/s. The signal $\widehat{\omega}_{gf}$, in comparison with the estimated grid

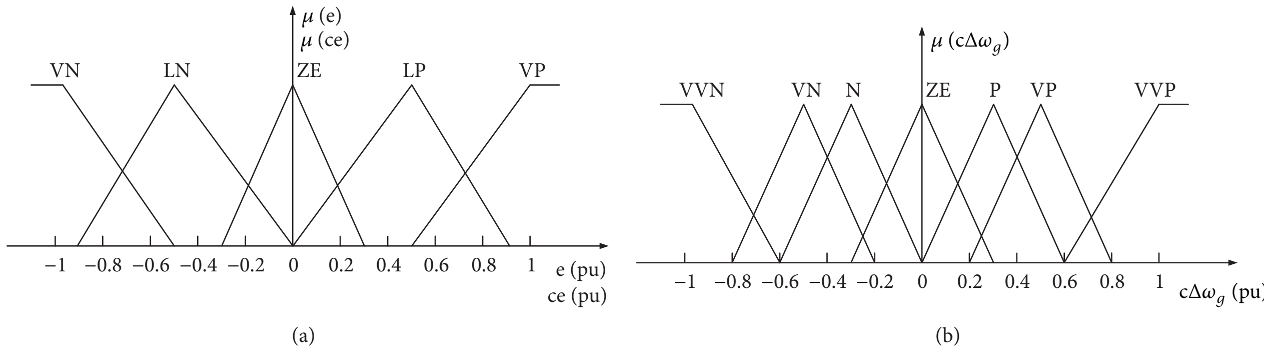

Figure 5: Membership functions of the developed Mamdani-type FC.

Table 1: Rule base for FC.

		ce				
		VN	**LN**	**ZE**	**LP**	**VP**
	VN	VVN	VVN	VN	N	ZE
e	**LN**	VVN	VN	N	ZE	P
	ZE	VN	N	ZE	P	VP
	LP	N	ZE	P	VP	VVP
	VP	ZE	P	VP	VVP	VVP

frequency $\widehat{\omega}_g$, has slower dynamics and it helps the FC to reach its steady-state setpoint. In addition, the parameter N_f, which is the input parameter of the adaptive MAF, is calculated according to (8), whereby the estimated angular grid frequency is replaced by its filtered value.

The rule base summarized in Table 1 and the membership functions shown in Figure 5 also apply to the FC in Figure 4.

3. Hedge-Algebra-Based Phase-Locked Loop with Genetic Optimized LRBs

In this study, the LRBs are converted from the respective fuzzy (linguistic label) rule bases (FRBs) established by the experience of human domain experts. Though these domain-experts' FRBs are very important, they are not always appropriate for specific application problems, particularly for complex problems. The construction of the FC component of the MAF-based PLL shown in Figures 2 and 4 is considered as such a problem.

A genetic method to design an optimized HAC is developed in order to replace the FC in the MAF-based PLL shown in Figures 2 and 4. This one is called the optimized HAC (*op*HAC) developed based on the hedge-algebra-formalism (HA-formalism). It is important and novel that the main component of the HAC, its LRB, is also to be optimized. However, as an LRB is genetically established by computer instead of human experts, this problem must only be solved in a formalism that is able to immediately handle linguistic words with their own real-world-semantics. Another distinguished advantage of the HA-formalism is its ability to ensure the real-world-semantics (RWS-) interpretability of the designed HACs, which means that the expected behaviors of the designed HACs are actually the same as described by their designers in terms of their LRBs [29]. It is also required that its approximate reasoning method must be RWS-interpretable as well. So, it can be seen that the RWS-interpretability of a designed HAC is very essential and important to ensure the high performance of not only the designed HACs but also the designed fuzzy systems. Also, it is for the first time applied to design RWS-interpretable HACs to solve application problems. Finally, their LRBs, with their own semantics, are for the first time genetically constructed.

3.1. The Role of Human Language in Designing RWS-Interpretable Fuzzy Systems. In the fuzzy set framework, words assigned to the fuzzy sets are considered as linguistic labels and there is no formal basis to use words and linguistic sentences of human natural language with their own *inherent* semantics. So, there exists a gap between the fuzzy representations representing the inherent semantics of their respective words and of their respective LRBs. The main aim of the fuzzy control is to utilize human capability in handling words of the natural language. If there was no formal bridge established across the gap, one would not be able to indeed make use of human expert linguistic knowledge. The paper [30] introduces an RWS-approach to the interpretability of fuzzy systems in which the RWS-interpretability of the human language is utilized to design RWS-interpretable fuzzy systems. Assuming the RWS-interpretability of the natural language, it implies that the LRB equipped for a fuzzy system is able to properly describe the reality with which the system interfaces. Based on this feature, the ultimate goal of the RWS-approach is to establish a methodology to design RWS-interpretable fuzzy systems, whose RWS-interpretability means that their behavior is ensured to be compatible with the expected one described by the given LRB.

The main specific features of the RWS-approach examined in [30] are the following:

(i) It is able to utilize the RWS-interpretability of the human language. In the fuzzy control, they usually describe *monotonic dependences* between two RW-variables.

(ii) Its formalism is able to immediately handle words and sentences of human expert languages *with their own RW-semantics* as well as their compt.-semantics.

These features are very crucial for any method to design a fuzzy (linguistic) controller to ensure its expected behavior when it interacts with its RW-counterpart.

3.2. Hedge Algebras of Variables, Their Quantification, and Their RWS-Interpretability. In order to ensure the RWS-interpretability of the designed HAC-PLLs, it is required to utilize the RWS-interpretability of human language to establish a mathematical formalization of word-domains for developing the HAC-PLLs. To do this, it is observed that the word-domain of every (linguistic) variable possesses the following two specific features:

(i) The word-domain can be generated from its atomic words, e.g., 'small' and 'large' of the current intensity variable, using linguistic hedges, e.g., 'very' or 'extremely'. Then, every word x is of the form $h_n \ldots h_1 c$, where $c \in \{small, large\}$ and h_j's are (linguistic) hedges and, hence, they can be considered as operations of an algebra.

(ii) There exists a linear or partial semantic order relation on this word-domain induced by the *inherent* semantics of its words.

Thus, hedges with their own order-based semantics defined in their word-domain play an essential role in discovering order-based structures of the word-domains of variables. However, a large number of studies consider words only as linguistic labels of the constructed fuzzy sets, whereas there are relatively few studies that *immediately* deal with linguistic hedges with their own modification functionality, called hedge operators. For instance, such hedge operators are examined and applied in fuzzy logic and approximate reasoning [31, 32], in contrast intensification [33], and in fuzzy control [34]. In particular, it is actively examined in formal concept analysis of a dataset based on studying formal concept lattice with hedges, called truth-stressing hedges [35–38]. In these studies, hedges are utilized to modify concept-forming operators to control the number of conceptual clusters extracted from a given dataset represented by formal concepts of a concept lattice of the dataset. However, the inherent semantics of words, which may contain hedge occurrences, is still not taken into consideration and, therefore, 'hedge operators' were developed in these studies independently from the semantics of words and hedges, which in practice must be determined in the context of other words of their variable.

Starting from the two aforementioned specific features of word-domains, HAs were proposed and developed in turn in [20, 30, 39, 40]. As argued in [30], all theories developed in an axiomatic way, whose axioms actually represent structural semantics of their respective RW-counterparts (e.g., math-theories or theoretical physics theories), are shown to be RWS-interpretable; in this section, we briefly explain that the theories of HAs and their quantification, which form the above formalism, are also RWS-interpretable.

3.2.1. A Description of an Axiomatization of Word-Domains and Their Quantification. HAs introduced in [20, 30, 39, 40] establish an algebraic approach based on a formalization of their word-domains of variables in an axiomatic way to immediately handle the inherent order-based semantics of the words. It is natural that there are comparability elements in the human language, the presence of which is required by the human decision-making activities in which the comparability of decision criteria values is crucial. So, there exists a semantic order relation on every variable word-domain. For instance, considering the RW-variable $\mathscr{I}_{RW}$ of the current intensity, this RW-variable may be examined on the standpoint either of a numeric formalism, in which its respective numeric variable is denoted by $\mathscr{I}_N$, or of a linguistic formalism, in which its respective linguistic variable is denoted by $\mathscr{I}_L$. For every hedge h of $\mathscr{I}_L$, say '*very*' (V), the pair of the words '*very large*' and '*large*' or the pair of the words '*very small*' and '*small*' of $\mathscr{I}_L$ are always comparable. It is essential that the order of these words is *compatible* with the order of the numeric values of the current intensity assigned to their respective words and, hence, the word-domain of $\mathscr{I}_L$ must also be linearly ordered.

For general case, let us consider an arbitrary linguistic variable denoted by X without its subscript L, for simplicity. As analyzed above, the word-domain of X becomes an order-based structure, denoted by $AX = (\text{Dom}(X), G, C, H, \leq)$, where $G = \{c^-, c^+\}$ – considered as the generators of AX, where $c^- < c^+$, c^+ is called the *positive* atomic word and c^- – the *negative* one; C is the set of constants $C = \{\mathbf{0}, W, \mathbf{1}\}$, which are, respectively, the least, neutral, and greatest words of $\text{Dom}(X)$; $H = H^- \cup H^+$ is the set of the hedges of X, where H^- (or H^+) is the set of the negative (or positive) hedges; e.g., $H^- = \{$R (Rather), L (Little)$\}$, and $H^+ = \{$M (More), V (Very)$\}$; and $\leq$ is the semantic order relation. Syntactically, $\text{Dom}(X)$ is the set of all the words of X of the form $x = h_n \ldots h_1 c$, $c \in G$, whose length is denoted by $|x|$; i.e., $|x| = n + 1$. Many properties of the inherent semantics of the words of X can be modeled in the order-based structure of $\text{Dom}(X)$ and, more importantly, as the human language is RWS-interpretable, they are RWS-interpretable as well. To formalize the word-domains in an axiomatic way, it is necessary to discover few properties of the order-based semantics of words and hedges that can easily be verified based on their natural meaning so that they can be taken as axioms and their math-models and the remaining properties can be derived from them. Restricted to the purpose of the study, a very short description of this formalization is given, including the necessary knowledge for the study. More details can be found in [20, 30, 39, 40].

(i) Words and hedges of X have their own "algebraic" signs determined in this order-based structure as follows.

(a) For atom words, $\text{sign}(c^-) = -1$ and $\text{sign}(c^+) = +1$; for any hedge h, $\text{sign}(h) = +1$ (or, $\text{sign}(h) = -1$) if $h \in H^+$ (or, $h \in H^-$) which is identified by $hc^+ \geq c^+$ (or, $hc^+ \leq c^+$).

(b) The *relative sign* of h with respect to k, denoted by $\text{sign}(h, k)$, is computed by $\text{sign}(h, k) = -1$ iff $(\exists c \in G)(kc \leq c \Longrightarrow kc \leq hkc)$ and $\text{sign}(h, k) = +1$ iff $(\exists c \in G)(kc \leq c \Longrightarrow hkc \leq kc)$. For illustration, the validity of Table 2 can be checked.

Table 2: The relative sign of a hedge in a row with respect to a hedge in a column.

	V	M	R	L
V	+	+	-	+
M	+	+	-	+
R	-	-	+	-
L	-	-	+	-

(c) Now, the sign of $x = h_n \ldots h_1 c$ is computed by sign (x) = sign $(h_n, h_{n-1}) \ldots$ sign (h_1) sign $(c) \in \{-1, +1\}$. There is an important fact that sign (hx) =–1 (or, +1) $\Longleftrightarrow$ $hx \leq x$ (or, $x \leq hx$). For example, for L, R, M, and V stand for *little*, *rather*, *more,* and *very*, respectively, it may be verified, for instance, that sign (VR_small) = sign (V,R)sign (R) × sign (small) = –1, which results in VR_small ≤ R_small.

(ii) By the aforementioned functionality of the hedges of X, for every hedge $h \in H$ and every word $x \in \mathrm{Dom}(X)$, the word hx inherits the semantics of his parent x, called *hedge-inheritance*, that can be formalized as follows: $c^- \leq c^+ \Longrightarrow H(c^-) \leq H(c^+)$ and $hx \leq kx \Longrightarrow H(hx) \leq H(kx)$, $\forall h, k$, where $H(z) = \{\sigma z\colon \sigma = h_n \ldots h_1 \in H*, \varnothing \in H*\}$. It states that any strings of hedges σ and σ' applying to hx and kx cannot change their inherent semantics defined by the inequality direction of $hx \leq kx$; i.e., $\sigma hx \leq \sigma' kx$ holds or, equivalently, $H(hx) \leq H(kx)$. By this, $H(x)$ can be considered in [39] as the fuzziness model of x. The set of the fuzziness models, $\{H(x)\colon x \in \mathrm{Dom}(X)\}$, has many properties representing the RW-semantics of the words which can be proved from the axioms of the HA of X [20]:

(a) The inclusion $H(hx) \subseteq H(x)$ represents the generality-specificity relation of x and hx: x is more general than hx and, conversely, hx is more specific than x. Moreover, we have

$$\begin{gathered}\{\mathbf{0}\} \cup H\left(c^-\right) \cup \{W\} \cup H\left(c^+\right) \cup \{\mathbf{1}\} = \mathrm{Dom}\left(X\right) \\ H\left(\mathrm{L}x\right) \cup H\left(\mathrm{R}x\right) \cup H\left(\mathrm{M}x\right) \cup H\left(\mathrm{V}x\right) = H\left(x\right) \\ \bigcup_{h \in X_k} H\left(hx\right) = H\left(x\right), \\ \text{where } X_k = \{x \in \mathrm{Dom}\left(X\right); |x| = k\}\end{gathered} \tag{11}$$

(b) The order of the fuzziness models of words is the same as the order of their words:

$$\begin{gathered}\{\mathbf{0}\} \leq H\left(c^-\right) \leq \{W\} \leq H\left(c^+\right) \leq \{\mathbf{1}\} \\ H\left(\mathrm{L}x\right) \leq H\left(\mathrm{R}x\right) \leq \{x\} \leq H\left(\mathrm{M}x\right) \leq H\left(\mathrm{V}x\right), \\ \text{for sign}\left(\mathrm{V}x\right) = +1 \\ H\left(\mathrm{L}x\right) \geq H\left(\mathrm{R}x\right) \geq \{x\} \geq H\left(\mathrm{M}x\right) \geq H\left(\mathrm{V}x\right), \\ \text{for sign}\left(\mathrm{V}x\right) = -1\end{gathered} \tag{12}$$

3.2.2. Quantification of the Words of Variable-Domains [39]. In nature, the compt.-semantics, including *numeric semantics*, *interval semantics,* and *fuzziness measure*, of the words of $\mathrm{Dom}(X)$ must be determined from their inherent order-based semantics presented above. Hence, they are closely defined. The axiomatic way to quantify the words is described as follows:

(1) Numeric Semantics of the Words. A mapping $f : \boldsymbol{X} \longrightarrow [0,1]$, where $[0,1]$ is the normalized universe of X, is said to be a numeric semantics interpretation if it is an *order-isomorphism*, whose image $f(\boldsymbol{X})$ is *dense in* $[0,1]$. It is called a *Semantically Quantifying Mapping* (SQM) and its values can be interpreted as the *numeric word semantics*.

(2) Interval Semantics and Fuzziness Measure of the Words. An SQM f maps each $H(x)$ to a subinterval of $[0,1]$, which is the smallest subinterval including the image of $H(x), f(H(x))$. It is called the *fuzziness interval* or an *interval semantics* of x and is denoted by $\mathfrak{I}(x) = [f(H(x))]$ and its length is called *fuzziness measure* of x, denoted by $fm(x)$, i.e., $fm : \boldsymbol{X} \longrightarrow [0,1]$.

(3) Strict Relationships of the Compt.-Semantics of the Words. It can be seen that, given an SQM f, as f preserves the order of the words, by (11) and (12), it follows by induction that

(i) for $x \in \{\mathbf{0}, c^-, W, c^+, \mathbf{1}\}$, their numeric semantics, i.e., their f-values, and the related fuzziness intervals are exactly located in $[0,1]$ as represented in Figure 6(a);

(ii) for x with $|x| = k > 1$, assume that its fuzziness interval $\mathfrak{I}(x)$ is already located in $[0,1]$. If sign $(\mathrm{V}x) = -1$ (sign $(\mathrm{V}x) = +1$), its numeric semantics $f(x)$ and its related fuzziness intervals are arranged as given in Figure 6(b) (are arranged so that a structure of the words generated from the atom c and the structure obtained from the former one by replacing c with the other atom word c' are mirror symmetrical).

From these, it follows that

$$\begin{gathered}f\left(\mathbf{0}\right) \leq \mathfrak{I}\left(c^-\right) \leq f\left(W\right) \leq \mathfrak{I}\left(c^+\right) \leq f\left(\mathbf{1}\right) \\ \mathfrak{I}\left(c^-\right) \cup \mathfrak{I}\left(c^+\right) = [0,1] \\ \mathfrak{I}\left(\mathrm{L}x\right) \leq \mathfrak{I}\left(\mathrm{R}x\right) \leq f\left(x\right) \leq \mathfrak{I}\left(\mathrm{M}x\right) \leq \mathfrak{I}\left(\mathrm{V}x\right), \\ \textit{if } \text{sign}\left(Vx\right) = +1 \\ \mathfrak{I}\left(\mathrm{V}x\right) \leq \mathfrak{I}\left(\mathrm{M}x\right) \leq f\left(x\right) \leq \mathfrak{I}\left(\mathrm{R}x\right) \leq \mathfrak{I}\left(\mathrm{L}x\right), \\ \textit{if } \text{sign}\left(Vx\right) = -1 \\ \mathfrak{I}\left(\mathrm{L}x\right) \cup \mathfrak{I}\left(\mathrm{R}x\right) \cup \mathfrak{I}\left(\mathrm{M}x\right) \cup \mathfrak{I}\left(\mathrm{V}x\right) = \mathfrak{I}\left(x\right), \\ \textit{for } x \text{ with } |x| \geq 1\end{gathered} \tag{13}$$

From (13) it follows that the fuzziness measure fm of X has the following properties:

($fm1$) $fm(c^-) + fm(c^+) = 1$;

($fm2$) $fm(\mathrm{L}x) + fm(\mathrm{R}x) + fm(\mathrm{M}x) + fm(\mathrm{V}x) = fm(x)$;

($fm3$) From ($fm2$) is inferred $fm(\mathrm{L}x)/fm(x) + fm(\mathrm{R}x)/fm(x) + fm(\mathrm{M}x)/fm(x) + fm(\mathrm{V}x)/fm(x) = 1$.

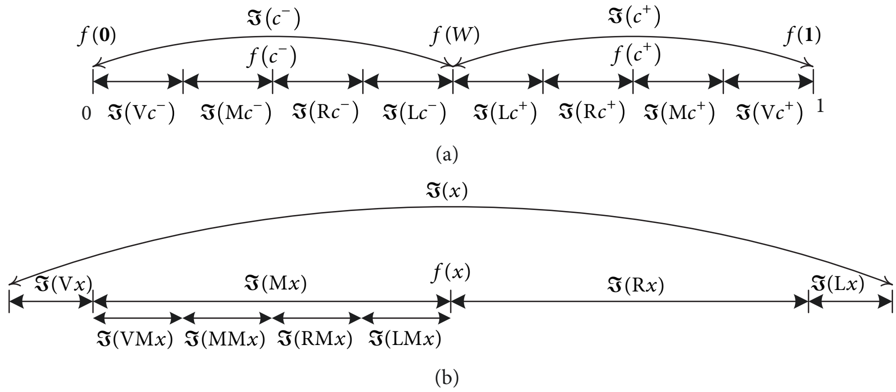

Figure 6: The structure of SQM-values and their related fuzziness intervals of the words of $C \cup \{c^-, c^+\}$ that is given in (a) and of the word x that is given in (b).

Assume that the fraction $fm(hx)/fm(x)$ does not depend on particular x; it can be considered as the fuzziness measure of h, denoted by $\mu(h)$, $h \in H$. Thus, it holds

($fm4$) $fm(hx) = \mu(h) fm(x)$ and $\mu(\mathrm{L})+\mu(\mathrm{R})+\mu(\mathrm{M})+\mu(\mathrm{V}) = \alpha + \beta = 1$,

where $\alpha = \mu(\mathrm{L}) + \mu(\mathrm{R})$ and $\beta = \mu(\mathrm{M}) + \mu(\mathrm{V})$, which are the sums of the fuzziness measure of, respectively, the negative hedges and the positive ones.

It is important that when the values of $fm(c^-)$, $\mu(\mathrm{R})$, $\mu(\mathrm{M})$, and $\mu(\mathrm{V})$ are given, $fm(x)$ can be calculated, for any word x. These values are called *independent* fuzziness parameters.

Now, interpreting ($fm1$) - ($fm3$) as axioms, a quantification theory is founded, in which the numeric semantics and the interval semantics of the words are easily computed [30, 39, 41], referring to Figure 6. This illustrates that the quantification of HAs can also developed in an axiomatic way and, hence, the quantification theory of HAs is *RWS-interpretable*.

3.3. Genetic Design of HAC-PLL. Two of the main components of HAC are its control LRB and approximate reasoning method (ARM) running on the graphical representation of its LRB. The HAC-PLL designed in this study has two new features: it is RWS-interpretable and its LRB is genetically constructed. Usually, the LRBs of FCs are formulated by human experts and, in general, they are very difficult to optimize. In this study, the LRBs of the designed HAC-PLLs are for the first time genetically optimally constructed, including the fuzziness parameters of their variables, while the RWS-interpretability of the designed HAC-PLLs is guaranteed. So, first, a method is examined to construct RWS-interpretable compt.-representations of LRBs and an RWS-interpretable ARM working on their compt.-representations constructed by this method. These concepts are first examined in [30] and improved in [40].

3.3.1. A Method to Construct RWS-Interpretable Graphical Representations of LRBs. Consider an LRB B consisting of n rules of m input variables and one output variable

$$(r_i)\ \text{IF } X_{1,L} \text{ is } x_{i,1} \& \ldots \& X_{mL} \text{ is } x_{i,m}, \text{THEN } X_{m+1,L} \text{ is } x_{i,m+1}; i = 1 \text{ to } n \tag{14}$$

where $X_{j,L}$'s denote the linguistic variables of the respective RW-variables X_j's; similarly, $X_{j,N}$'s denote the numeric variables of the respective RW-variables X_j's and their universes are denoted, respectively, by $U(X_{j,N})$'s.

As this LRB is generally assumed to be consistent, it describes a *linguistic* functional dependence of $X_{m+1,L}$ on the remaining ones $X_{j,L}$'s defined in the Cartesian product $\mathrm{Dom}(X_{1L}) \times \ldots \times \mathrm{Dom}(X_{m+1,L})$. Methodologically, the method to construct compt.-representations of LRBs, written shortly as RMd, is crucial because of the technical requirement for studying the RWS-interpretability not only of the LRBs but also of the ARMs working on them. In addition, when methods to genetically design LRBs are examined, they must deal not only with the individual compt.-representations but also with the method itself. Thus, the functionality of an RMd is able to construct from any given LRB B a compt.-representation, denoted by $\mathscr{R}_{\mathrm{RMd}}(B)$, defined in $U(X_{1,N}) \times \ldots \times U(X_{m+1,N})$. At the same time, as human language is RWS-interpretable, the linguistic function defined by B does describe an RW-function f_{RW} of X_{m+1} on the remaining X_j's. Denote by f_N the numeric function modeling the same f_{RW}; the both functions must be *compatible* with each other. This leads to the following constraint on RMd's interpreted as a test condition whether an RMd is RWS-interpretable.

Constraint on RMd: the output $\mathscr{R}_{\mathrm{RMd}}(B)$ of RMd must be a numeric function defined in the Euclid space $U(X_{1,N}) \times \ldots \times U(X_{m+1,N})$, denoted by $f_{N,\mathrm{RMd}}(B)$, and if the linguistic function B is increasingly or decreasingly monotonic function, then so is the $f_{N,\mathrm{RMd}}(B)$.

This compt.-representation $f_{N,\mathrm{RMd}}(B)$ of the LRB B is called *graphic* representation of B. It can be seen that it is an aggregation of the elementary linguistic predicates given in B. It should be emphasized that, in accordance with our knowledge, no aggregation operators aggregating these (fuzzy) linguistic predicates can preserve such an

RW-semantics represented by the LRB B like this graphic representation. In relation to this, it is observed that practical applications of the math-theories and the theoretical physics theories show that the graphic representations of functions in Euclid space applied in these theories are RWS-interpretable.

In this section, an RMd developed in the HA-formalism to construct graphic representation of LRBs, called graphical RMd and denoted by M, is shortly described as follows:

(i) Determine HAs of $\mathrm{Dom}(X_{j,L})$, $j = 1$ to $m+1$, by selecting their negative and positive hedges and determining their relative signs and their fuzziness parameter values.

(ii) With these, the SQMs $f_{Xj,L}$'s are defined and map the n linguistic points determined by the LRB of B in $\mathrm{Dom}(X_{1,L}) \times \ldots \times \mathrm{Dom}(X_{m+1,L})$ into n points in the Euclid space $[0,1]^{m+1}$ that forms a grid of points, denoted by $G_M(B)$.

To examine the RWS-interpretability of $G_{\mathrm{RMd}}(B)$ some notations are needed. Let $\boldsymbol{a} = (a_1, \ldots, a_m, a_{m+1})$ be a point in $[0,1]^{m+1}$. The project of $\boldsymbol{a}$ on the Cartesian space indicated by $X_{i1,N} \times \ldots \times X_{ik,N}$; i.e., $\boldsymbol{a}|_{X_{i1}\times\ldots\times X_{ik}} = (a_{i1}, \ldots, a_{ik})$ is denoted by $\boldsymbol{a}|_{X_{i1}\times\ldots\times X_{ik}}$. For any RMd, a point of $G_{\mathrm{RMd}}(B)$ corresponding to a rule r_i is denoted by $G_{\mathrm{RMd}}(r_i)$. By the constraint above, the RWS-interpretability of an RMd can be defined as follows, noting that the LRBs in many control applications are monotonic.

Definition 1. A given method RMd to produce a compt.-representation in a Cartesian product S^{m+1} from a given LRB is said to be RWS-interpretable provided that if B is increasingly (decreasingly) monotonic, then

(i) $G_{\mathrm{RMd}}(B) \subseteq U(X_{1,N}) \times \ldots \times U(X_{m+1,N})$ must define a function. That is, for every rule r_i of B, $G_{\mathrm{RMd}}(r_i)$ is of the form $\boldsymbol{a}_i = (a_{i,1}, \ldots, a_{i,m}, a_{i,+1})$, for $i = 1, \ldots, n$, and for any such two vectors $\boldsymbol{a}_j = (a_{j,1}, \ldots, a_{j,m}, a_{j,m+1})$ and $\boldsymbol{a}_{j'} = (a_{j',1}, \ldots, a_{j',m}, a_{j',m+1})$, the equality $\boldsymbol{a}_j|_{X_1\times\ldots\times X_{m+1}} = \boldsymbol{a}_{j'}|_{X_1\times\ldots\times X_m}$ implies that $\boldsymbol{a}_j|_{X_{m+1}} = \boldsymbol{a}_{j'}|_{X_{m+1}}$.

(ii) For B being an increasingly (or decreasingly) monotonic function and its i-th rule r_i is denoted by $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,m}, x_{i,m+1})$, it holds $\boldsymbol{x}_i|_{X_1\times\ldots\times X_m} \le \boldsymbol{x}_j|_{X_1\times\ldots\times X_m} \Longrightarrow G_{\mathrm{RMd}}(r_i)|_{X_{m+1}} \le G_{\mathrm{RMd}}(r_j)|_{X_{m+1}}$ (or, $\boldsymbol{x}_i|_{X_1\times\ldots\times X_m} \le \boldsymbol{x}_j|_{X_1\times\ldots\times X_m} \Longrightarrow G_{\mathrm{RMd}}(r_i)|_{X_{m+1}} \ge G_{\mathrm{RMd}}(r_j)|_{X_{m+1}}$).

For the RMd M, as the determined SQMs are isomorphisms, it can easily be verified that the monotonicity of a given B is preserved by M; i.e., the $G_M(B)$ has the same monotonicity property as B. So, the following proposition holds:

Proposition 2. *The RMd M proposed as above is RWS-interpretable.*

3.3.2. The RWS-Interpretability of Approximate Reasoning Method. Each ARM should be developed accompanied with a given RMd to solve an application problem and it also plays a key role to construct an RWS-interpretable fuzzy system. It strongly depends on a given LRB B as well as on a given RMd. Therefore, to define the RWS-interpretability of a given ARM, $\mathbb{R}$, it is assumed that the given RMd accompanied with $\mathbb{R}$ is RWS-interpretable. Let B be an LRB of the form (14). For any input vector $\boldsymbol{a} = (a_1, \ldots, a_m)$ in $X_{1,N} \times \ldots \times X_{m,N}$, denote by $\mathbb{R}(G_{\mathrm{RMd}}(B))(\boldsymbol{a})$ the output value of $\boldsymbol{a}$ in $\mathrm{Dom}(X_{m+1,N})$ produced by $\mathbb{R}$ working on $G_{\mathrm{RMd}}(B)$. Then, the following definition may be introduced which is a modified version of the respective one examined in [30].

Definition 3. Let be given an RMd to produce compt.-representations of any given LRBs and let an ARM $\mathbb{R}$ be developed to work on the compt.-representations produced by the RMd. $\mathbb{R}$ is said to be *RWS-interpretable* provided that for any LRB B, which is increasingly (or decreasingly) monotonic, $\mathbb{R}$ should satisfy the following condition:

$$(\forall \boldsymbol{a}, \boldsymbol{b}) \{[\boldsymbol{a} < \boldsymbol{b} \Longrightarrow \mathbb{R}(G_{\mathrm{RMd}}(B))(\boldsymbol{a}) < \mathbb{R}(G_{\mathrm{RMd}}(B)) \cdot (\boldsymbol{b})]\}$$
$$(\text{or}, (\forall \boldsymbol{a}, \boldsymbol{b}) \{[\boldsymbol{a} < \boldsymbol{b} \Longrightarrow \mathbb{R}(G_{\mathrm{RMd}}(B))(\boldsymbol{a}) > \mathbb{R}(G_{\mathrm{RMd}}(B))(\boldsymbol{b})]\}). \tag{15}$$

The conditions in this definition are natural but very strong for the ARMs developed in the fuzzy set framework. In this study, the so-called semantically weighted interpolation-extrapolation reasoning method proposed in [40] is applied and described as follows.

(i) Semantically Weighted Interpolation-Extrapolation Reasoning Method.

(A) Interpolation: consider a graphic representation $G_{\mathrm{RMd}}(B)$ of B.

(1) Define semantic weight of the closeness of a given input vector falling in a grid-mesh of $G_{\mathrm{RMd}}(B)$ to each grid-mesh point: a restriction is applied to the case of two input variables which is compatible to the HAC-PLL examined in this study. Assume that a mesh of the 2D-grid graphically representing B is exhibited in Figure 7, in which four points of the mesh are calculated from four respective linguistic points (x_{1i}, x_{2j}), $(x_{1(i+1)}, x_{2j})$, $(x_{1i}, x_{2(j+1)})$, and $(x_{1(i+1)}, x_{2(j+1)})$ and denoted by (a_{1i}, a_{2j}), $(a_{1(i+1)}, a_{2j})$, $(a_{1i}, a_{2(j+1)})$, and $(a_{1(i+1)}, a_{2(i+1)})$. That is, $a_{k,l} = f_{Xk}(x_{k,l})$, $k = 1, 2$ and $l \in \{i, i+1, j, j+1\}$. The $f_{X_{m+1}}$-values at these points are denoted by u_{11}, u_{12}, u_{21}, and u_{22}. To simplify the presentation, these values are used to name the respective points (a_{1i}, a_{2j}), $(a_{1(i+1)}, a_{2j})$, $(a_{1i}, a_{2(j+1)})$, and $(a_{1(i+1)}, a_{2(i+1)})$ and they indicate the rectangle represented in Figure 7. It is denoted by $(u_{11}, u_{12}, u_{21}, u_{22})$ and its area being denoted by S_{ij}. Consider an input vector $a = (a_1, a_2)$, whose components fall, respectively, into $[a_{1i}, a_{1(i+1)}]$ and $[a_{2j}, a_{2(j+1)}]$. The expected value u of the input vector is computed by the weighted average of u_{11}, u_{12}, u_{21}, and u_{22} whose semantic weights are defined based on how close to each mesh point it is: the closer to a point, the larger its semantic weight. The point (a_1, a_2) decomposes the mesh $(u_{11}, u_{12}, u_{21}, u_{22})$ into four rectangles. The area of any of the rectangles determined by any point $c = (c_1, c_2)$ is denoted by the following rule: if it is located opposite to the position of

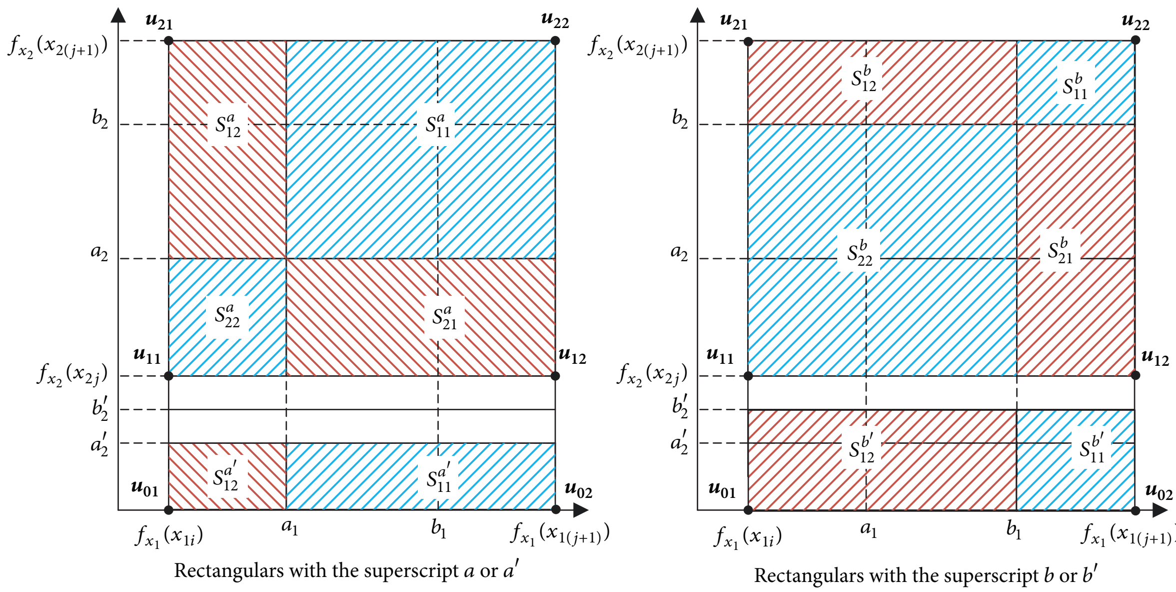

Figure 7: Semantically weighted inter-extrapolative reasoning method on $G(B)$.

the value u_{kl}, it is denoted by S_{kl}^c. In Figure 7, S_{11}^a is opposite to u_{11}, S_{21}^a is opposite to u_{21}, $S_{12}^{a'}$ and $S_{12}^{b'}$ are opposite to u_{12}, and so on.

The semantic weight of the closeness of a to a mesh point (a_{1k}, a_{2l}), whose $f_{X_{m+1}}$-value is u_{kl}, denoted by $w_a(u_{kl}) = w_a(a_{1k}, a_{2l})$, is defined by the following product:

$$\begin{aligned} w_a(a_{1k}, a_{2l}) &= \left(1 - \frac{|a_1 - a_{1k}|}{a_{1(i+1)} - a_{1i}}\right)\left(1 - \frac{|a_2 - a_{2l}|}{a_{2(j+1)} - a_{2j}}\right) \\ &= \frac{\left(\left(a_{1(i+1)} - a_{1i}\right) - |a_1 - a_{1k}|\right)\left(\left(a_{2(j+1)} - a_{2j}\right) - |a_2 - a_{2l}|\right)}{S_{ij}} \end{aligned} \quad (16)$$

The product in the numerator of (16) is just the area of the rectangle S_{kl}^a opposite to the position of u_{kl}, where no confusion occurs if the area of a rectangle S_{ij} is denoted also by S_{ij}.

(2) Compute the control value using the semantically weighted average: for the given input vector $\boldsymbol{a}$, u is computed as the semantically weighted average as follows:

$$u^a = \frac{S_{11}^a u_{11} + S_{12}^a u_{12} + S_{21}^a u_{21} + S_{22}^a u_{22}}{S_{ij}} \quad (17)$$

(B) Extrapolation: assume that a' falls outside of the 2D-grid $G_{\text{RMd}}(B)$. There are two cases. The first one is that a' falls into a rectangle which have exactly two vertices belonging to $G_{\text{RMd}}(B)$, say the rectangle $(u_{01}, u_{02}, u_{11}, u_{12})$ in Figure 7. Similar as above, u is computed by

$$u^{a'} = \frac{S_{11}^{a'} u_{11} + S_{12}^{a'} u_{12}}{S_{11}^{a'} + S_{12}^{a'}} \quad (18)$$

For a' falling into a rectangle outside of $G_{\text{RMd}}(B)$ having exactly one common vertex with $G_{\text{RMd}}(B)$, e.g., the vertex u_{11} of the rectangle domain of the product $e \times ce$, similar as above, u is computed by

$$u^{a'} = \frac{S_{11}^{a'} u_{11}}{S_{11}^{a'}} = u_{11} \quad (19)$$

Theorem 4. *For the graphical RMd M, the semantically weighted interpolation-extrapolation (SWIE) reasoning method working with M is RWS-interpretable; i.e., it preserves the monotonicity property of the graphic representation $G_{\text{RMd}}(B)$, for any LRB B.*

Proof. Let B be increasing; thus, so is $G_{\text{RMd}}(B)$. In the case of interpolation, it is sufficient to consider the case where two input vectors $a = (a_1, a_2)$ and $b = (b_1, b_2)$ in question satisfy $a \leq b$ and all fall into the same mesh of the grid $G_{\text{RMd}}(B)$. Based on these vectors, the rectangle $(u_{11}, u_{12}, u_{21}, u_{22})$ is divided into nine rectangles indicated by S_{22}^a, $S_{22}^b \cap S_{21}^a$, $S_{21}^b \cap S_{21}^a$, $S_{22}^b \cap S_{12}^a$, $S_{22}^b \cap S_{11}^a$, $S_{21}^b \cap S_{11}^a$, $S_{12}^b \cap S_{12}^a$, $S_{12}^b \cap S_{11}^a$ and S_{11}^a, as computed and shown in Figure 7. By the method SWIE, the outputs u_a and u_b are computed by

$$\begin{aligned} u^a &= \frac{S_{11}^a u_{11} + S_{12}^a u_{12} + S_{21}^a u_{21} + S_{22}^a u_{22}}{S_{ij}} \\ \text{and } u^b &= \frac{S_{11}^b u_{11} + S_{12}^b u_{12} + S_{21}^b u_{21} + S_{22}^b u_{22}}{S_{ij}} \end{aligned} \quad (20)$$

Examine now the sign of the following quantity:

$$\begin{aligned} S_{ij}\left(u^b - u^a\right) = {} & \left(S_{11}^b - S_{11}^a\right) u_{11} + \left(S_{12}^b - S_{12}^a\right) u_{12} \\ & + \left(S_{21}^b - S_{21}^a\right) u_{21} + \left(S_{22}^b - S_{22}^a\right) u_{22} \end{aligned} \quad (21)$$

Considering the above nine rectangles as elementary ones, it can be seen that

$$\begin{aligned}
S^a_{11} &= S^b_{22} \cap S^a_{11} + S^b_{21} \cap S^a_{11} + S^b_{12} \cap S^a_{11} + S^b_{11} \\
S^b_{12} &= S^b_{12} \cap S^a_{12} + S^b_{12} \cap S^a_{11} \\
S^a_{21} &= S^b_{21} \cap S^a_{21} + S^b_{21} \cap S^a_{11} \\
S^b_{22} &= S^a_{22} + S^b_{22} \cap S^a_{21} + S^b_{22} \cap S^a_{12} + S^b_{22} \cap S^a_{11}
\end{aligned} \tag{22}$$

By (22), the quantity (21) can be reduced to

$$\begin{aligned}
S_{ij}\left(u^b - u^a\right) = {} & \left(S^b_{12} \cap S^a_{11}\right)\left(u_{12} - u_{11}\right) \\
& + \left(S^b_{21} \cap S^a_{11}\right)\left(u_{21} - u_{11}\right) \\
& + \left(S^b_{22} \cap S^a_{11}\right)\left(u_{22} - u_{11}\right) \\
& + \left(S^b_{22} \cap S^a_{12}\right)\left(u_{22} - u_{12}\right) \\
& + \left(S^b_{22} \cap S^a_{21}\right)\left(u_{22} - u_{21}\right)
\end{aligned} \tag{23}$$

By the increasing monotonicity of $G_{\mathrm{RMd}}(B)$, it holds $u_{11} \le u_{12} \le u_{22}$ and $u_{11} \le u_{21} \le u_{22}$ that leads to $S_{ij}(u^b - u^a) \ge 0$.

In the case of extrapolation, assume for example that the input vectors $a' = (a_1, a_2')$ and $b' = (b_1, b_2')$ fall into the rectangle $(u_{01}, u_{02}, u_{11}, u_{12})$. To simplify and to easily read the proof expressions, it is imagined that this rectangle is linearly transformed into the unit-rectangle whose vertex u_{01} locates at the center of the coordinate system. If it is proved that $u^{a'} \le u^{b'}$ in the unit-rectangle, then, by the linear transformation, this respective inequality is also valid in the original rectangle. So, for simplicity, imagine that a and b are already linearly transformed in the unit-rectangle; i.e., $u_{01} = (0, 0)$, $u_{02} = (0, 1)$, and $u_{11} = (0, 1)$. By (18), it holds

$$\begin{aligned}
u^{b'} - u^{a'} &= \frac{S^{b'}_{11}u_{11} + S^{b'}_{12}u_{12}}{S^{b'}_{11} + S^{b'}_{12}} - \frac{S^{a'}_{11}u_{11} + S^{a'}_{12}u_{12}}{S^{a'}_{11} + S^{a'}_{12}} \\
&= \frac{(1-b_1)\,b_2'u_{11} + b_1b_2'u_{12}}{1 \times b_2'} - \frac{(1-a_1)\,a_2'u_{11} + a_1a_2'u_{12}}{1 \times a_2'} \\
&= \frac{a_2'\,(1-b_1)\,b_2'u_{11} + a_2'b_1b_2'u_{12} - (1-a_1)\,a_2'b_2'u_{11} + a_1a_2'b_2'u_{12}}{a_2' \times b_2'} \\
&= \frac{a_2'b_2'\,(b_1-a_1)\,(u_{12}-u_{11})}{a_2' \times b_2'} \ge 0
\end{aligned} \tag{24}$$

As the remaining cases are proved in a similar way, it can be concluded that the method SWIE preserves the monotonicity property of the given LRB B. By Definition 3, the method SWIE is RWS-interpretable. □

3.3.3. Genetic Algorithm to Design Optimized HAC for MAF-PLL. In this section, it is described how to develop a genetic algorithm (GA), denoted HAC-GA, to design *op*HAC for the MAF-PLL with HAC. The HAC-GA has three tasks: (i) to optimize LRBs, (ii) to optimize the variable fuzziness parameters, and (iii) to optimize the variable ranges. So, the HAC-GA has a functionality to simultaneously produce optimized LRB, optimized fuzziness parameters, and optimized variables ranges. To develop such a HAC-GA the following tasks should be performed.

Table 3: An Example of an LRB *B*.

		ce				
		VN	**LLN**	**ZE**	**LLP**	**VP**
	VN	VVN	VVN	VN	N	ZE
e	**LLN**	VVN	VN	N	ZE	P
	ZE	VN	N	ZE	P	VP
	LLP	N	ZE	P	VP	VVP
	VP	ZE	P	VP	VVP	VVP

Task 1. Encode the Solution of the Optimization Problem of HACs. (a) Solutions of the optimization problem of HACs: each solution comprises three groups: LRB, fuzziness parameters, and ranges of the variables *e*, *ce,* and *u*. Assume that the HAs of these variables are syntactically the same; i.e., they have the same set of atomic words $G = \{$ N (negative), P (positive) $\}$ and the same set of hedges $H = \{$ L, V $\}$. Hence, Dom(e) = Dom(ce) = {VN, LLN, ZE, LLP, VP} and Dom(u) = {VVN, VN, LVN, N, LLN, LN, VLN, ZE, VLP, LP, LLP, LVP, VP, VVP}, which are *ordered from left to right*. The limitation of Dom(e) and Dom(ce) of having only five words aims to be compatible with their five fuzzy sets of the FCs for comparative study in Section 4. A representation of an LRB of the HAC is given in the form of Table 3. It is assumed that this table is *antonym* in the sense defined as follows.

For every X, two words $x = h_n \ldots h_1 c$ and $y = h'_m \ldots h'_1 c'$ are said to be contradictory or antonyms if $h_n \ldots h_1 = h'_m \ldots h'_1$ and c, c' belonging to $\{ c^-, c^+ \}$ are antonym. For example, *LLP* and *LLN* are antonym. A cell of Table 3 can be written as (x, y, z), $x \in$ Dom(e), $y \in$ Dom(ce) and $z \in$ Dom(u). Two cells (x, y, z) and (x', y', z') are said to be antonym if the pairs of the words (x, x'), (y, y') and (z, z') are antonym. Then, an LRB represented by Table 3 is said to be antonym if for any two its cells (x, y, z) and (x', y', z') whose pairs (x, x') and (y, y') are antonym, then so is the pair (z, z'). As a consequence, the words of the cells lying on the secondary diagonal of the table are the same and equal to the neutral words (ZE) and, when the upper secondary diagonal of the table is known, the table of rules is completely determined. It can be seen that the LRB given in Table 3 is antonym.

(b) Individual coding: Assume that the *real* coding is applied. To genetically design *op*HAC, the following should be coded: its LRB B of the form given in Table 3, the independent fuzziness parameters, and the ranges of the variables *e*, *ce*, and *u*. For their fuzziness parameters and their range values, by the symmetricity of the variable ranges, it is assumed that $fm_e(c^-) = fm_{ce}(c^-) = fm_u(c^-) = 0.5$ and, hence, the GA to be developed needs to seek for optimized values of μ_e(L), μ_{ce}(L), and μ_u(L). So, the GA needs only three genes to represent the fuzziness parameters and only three genes to represent the variable ranges. For the words of the variable u in Table 3, it is required to know only ten words below the

right diagonal of the table which are arranged from *left to right* and from *top to bottom*. For example, the LRB B given in Table 3 is represented by a chromosome of ten genes: (P P VP P VP VVP P VP VVP VVP). Each solution is encoded by an individual of 16 genes: the first three genes represent the values $\mu_e(\mathrm{L})$, $\mu_{ce}(\mathrm{L})$, and $\mu_u(\mathrm{L})$, the next ten genes represent the ten above-mentioned words of u uniquely determining B, and the three remaining genes represent three positive variable range values. Assume that the words of $\mathrm{Dom}(u)$ = {VVN, VN, LVN, N, LLN, LN, VLN, ZE, VLP, LP, LLP, P, LVP, VP, VVP} are increasingly arranged and encoded by their positions, i.e., by the integers 0, 1, 2, . . ., 14. For example, three values $\mu_e(\mathrm{L})$, $\mu_{ce}(\mathrm{L})$, and $\mu_u(\mathrm{L})$, three positive variable ranges, rg_X's, and the B given in Table 3 are represented by the solution individual: ($\mu_e(\mathrm{L})$, $\mu_{ce}(\mathrm{L})$, $\mu_u(\mathrm{L})$, 11, 11, 13, 11, 13, 14, 11, 13, 14, 14, rg_e, rg_{ce}, rg_u).

Task 2. Construct a HAC, an Alternative Component for the FC

(1) Determine the inputs of a HAC

They comprise the following:

(i) the positive and negative hedges, their relative signs, fuzziness parameter values and the positive variable range values;
(ii) an LRB B given in the form of Table 3;
(iii) an ARM— a selected interpolative method working on the surface in $[0,1]^3$, which is computed using the SQMs f_e, f_{ce}, and f_u;
(iv) input vector of the control problem.

These input values are needed in each iteration, noting that the values listed in (i) and (ii) are provided by an individual under consideration of a population generated by the HAC-GA.

(2) Construct components of a HAC of the HAC-PLL

Its main components are shown in Figure 8 which comprises the following:

(i) the normalization to convert the universes of e, ce, and u to their respective unit intervals $[0,1]$'s;
(ii) a given LRB B and its quantification to translate B into a graphic representation approximately modeling a surface S^3_{real};
(iii) construct an RWS-interpretable interpolation-extrapolation reasoning method. In this study, the developed method SWIE is applied;
(iv) the denormalization to convert the numeric output of u computed by the second component of HAC to its respective real value of the variable u.

Task 3. Construct a GA, Denoted by GA(opHAC), to Produce the Optimized LRB B_{op}, the Optimized Fuzziness Parameters, and the Optimized Variable Ranges of the Desired opHAC. The HAC developed in *Task 1* is a kernel-component of the desired $GA(op\mathrm{HAC})$.

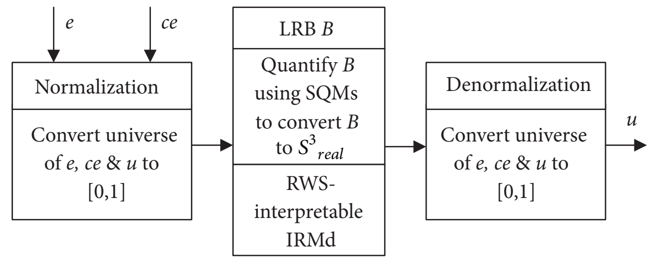

FIGURE 8: The components of the HAC.

(1) Inputs of $GA(op\mathrm{HAC})$:

(i) The criterion $J_1 = \int_0^\delta |e(t)|dt$ (or $= \sum_{k=1}^{n} |e(k)|$, $n = \delta/T_s$), called the performance index (Prf-index) in this study, and the time period of simulation δ for computing the Prf-index J_1
(ii) GA running parameters: population size; generation number; disturbance type

(2) Outputs of $GA(op\mathrm{HAC})$:

(i) The optimized values of the fuzziness parameters $\mu_e(\mathrm{L})$, $\mu_{ce}(\mathrm{L})$, and $\mu_u(\mathrm{L})$ and the optimized variable ranges $(-rg_e, rg_e)$, $(-rg_{ce}, rg_{ce})$, and $(-rg_u, rg_u)$
(ii) The optimized LRB B_{op}

(3) General description of $GA(op\mathrm{HAC})$:

(i) Population of solutions: $GA(op\mathrm{HAC})$ will initialize an initial population. Every individual of a population consists of 16 genes, 3 genes for the fuzziness parameters, 3 genes for positive variable ranges values, and 10 genes for the words to determine LRB of the form in Table 3. Genetic operations like crossovers and mutation will be applied to generate the next population of the same size, in which only individuals with better fitness values are retained.
(ii) Method to compute the Prf-index J_1 of an individual of a given population: the procedure of $GA(op\mathrm{HAC})$ to compute J_1 of each individual of the population in question, denoted by δ-$\mathrm{Loop}_{op\mathrm{HAC}}$, is described by the scheme given in Figure 9.

(4) Implementation of $GA(op\mathrm{HAC})$ in Matlab (see Box 1)

4. Results and Discussion

In this section, the performance of the proposed HAC-PLL is validated through experimental studies with dSpace DS1103 R&D board. For this purpose, all the PLLs considered in this study were programmed in the MATLAB/Simulink environment. The comparison of the results obtained by the developed HAC-PLL with those obtained by both the PID-based MAF-PLL and the FC-based MAF-PLL is also shown.

```
GA(opHAC)
Input: (i) Simulation model 'HACPLLfinal.slx' suitable type for the expected disturbance;
        (ii) J_1(I) = Σ_{e(k)∈Set_δ(e(k))} |e(k)|; // I is an individual of the population under consideration
Output: The optimized HAC: 'opHAC';
  Method:
    Initialize parameters of the simulation model;
    Initialize parameters of the GA:
        num_var = number of variables;
        pop_size = population size;
        num_gen = number of generation;
        time_limit = time value; // to limit the running time of the algorithm GA(opHAC)
    Initialize an initial population;
        m = 1; t = 0;
        while ((m ≤ num_gen) and (t ≤ time_limit))
          for each individual I_m in the population
            call simulation model: 'HACPLLfinal.slx';
            compute J_1(I_m) = Σ_{e(k)∈Set_δ(e(k))} |e(k)|;
          end_for
          m = m+1;
          Perform genetic operators;
          auto_update(t);
        end_while
End_GA(opHAC)
```

Box 1

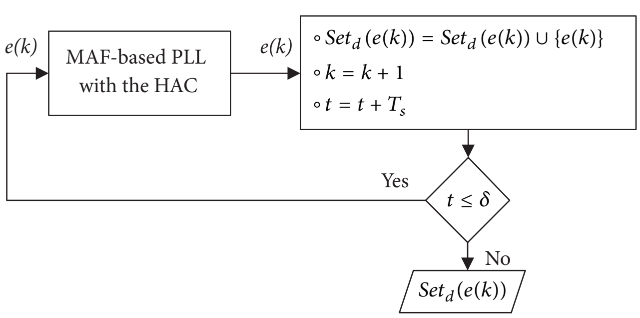

Figure 9: δ-Loop$_{\text{opHAC}}$ of the simulation model.

In addition to real-time execution of the analyzed PLL algorithms, the R&D board was used for grid emulation as well. In this regard, the D/A converters of the R&D board were utilized for generation of the phase voltages, but with reduced amplitude. The amplitude of the nondistorted phase voltages has been selected to cover ±10 V output range of the D/A converters when disturbances occur. The phase voltages were then sampled at 40 kHz sampling frequency by using the A/D converters of the same board. The sampling frequency of the analyzed PLLs was set to 10 kHz taking into consideration that the fastest time-varying disturbance analyzed in this paper has an oscillation frequency less than 5 kHz.

Performance of the analyzed MAF-based PLL was tested under the following four conditions:

(1) Unbalanced voltage sag

(2) Unbalanced voltage sag with harmonics distortion (i.e., positive 5th and 7th order harmonics have been added with the amplitudes equal to 8% of the 1st phase voltage harmonic)

(3) Unbalanced transient in the phase voltages with the oscillation frequency of 500 Hz

(4) Step change of the grid frequency of +5 Hz and back again

For the disturbances (1), (2), and (4), the amplitude of the nondistorted phase voltages has been selected 8.6 V, whereas for the disturbance (3), it has been selected 1.7 V.

Parameters of the FC and the HAC are obtained in simulations as mentioned in Sections 2.2 and 3.3. Parameters of the analyzed disturbances are in accordance with the IEC classification of electromagnetic phenomena [42]. Distorted phase voltages and responses of the analyzed PLLs are shown in Figures 10–13. In this paper, it is selected that each disturbance lasts 200 ms exactly.

Figure 10 shows the recorded responses under the unbalanced voltage sag in phases a and c. The amplitudes of the faulted phases a and c are 80% and 92% of the nondistorted phase voltages, respectively. Figures 10(b), 10(c), and 10(d) show traces of the phase error, the filtered quadrature voltage component u_q, and the IAE of the controllers, respectively. The HAC controller ensures the lowest IAE value - 4% lower compared to the FC and 44% lower compared to the PID.

The phase voltages with harmonics under the unbalanced voltage sag in phases a and c are considered in Figure 11.

The traces of the phase error (Figure 11(b)), the quadrature voltage component u_q (Figure 11(b)), and the IAE (Figure 11(c)) are very similar to the corresponding traces shown in Figure 10. This means that the influence of the

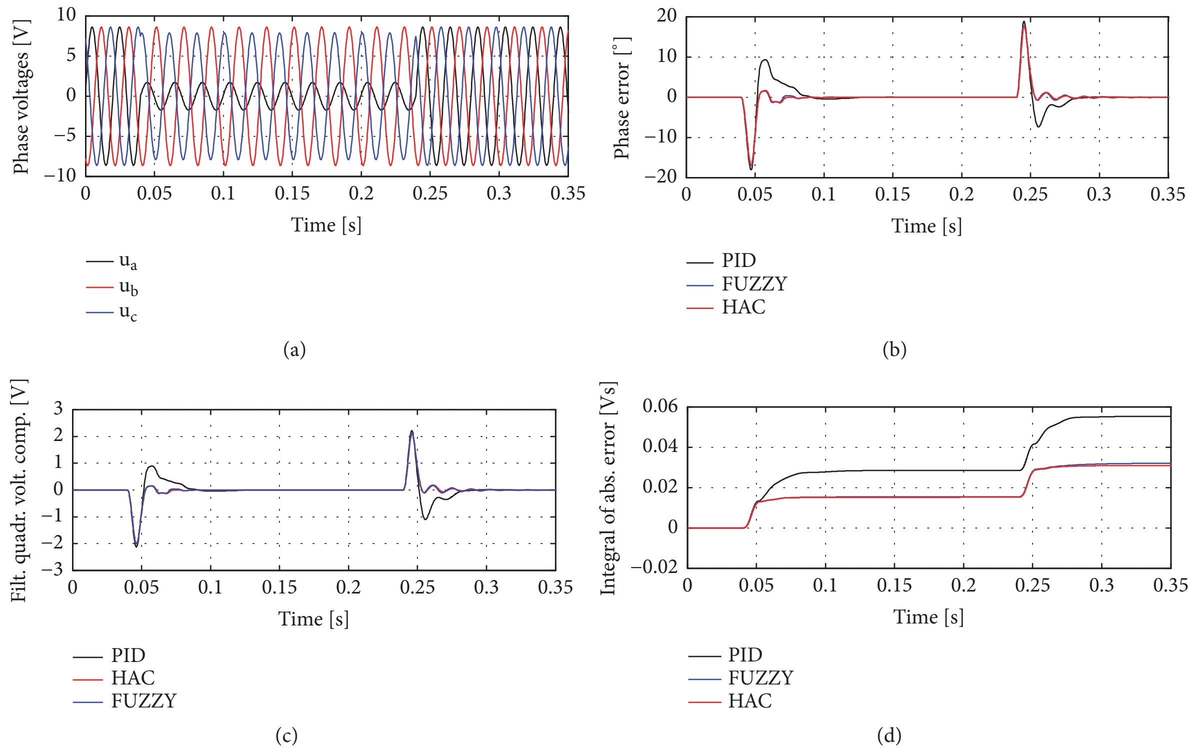

Figure 10: Phase voltages (a), phase error (b), filtered quadrature voltage component $\overline{u}_q$ (c), and integral of absolute error (d) for the unbalanced voltage sag in phases *a* and *c*.

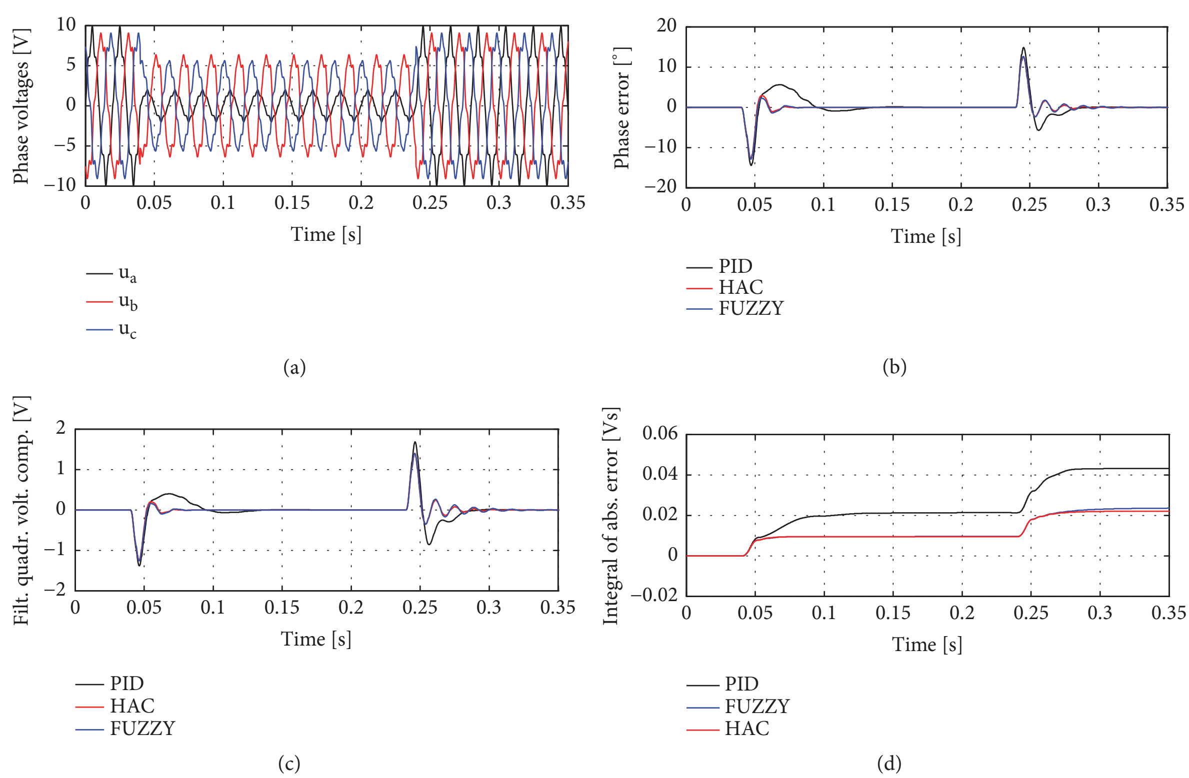

Figure 11: Phase voltages (a), phase error (b), filtered quadrature voltage component $\overline{u}_q$ (c), and integral of absolute error (d) for the unbalanced voltage sag in phases *a* and *c* with harmonics distortion.

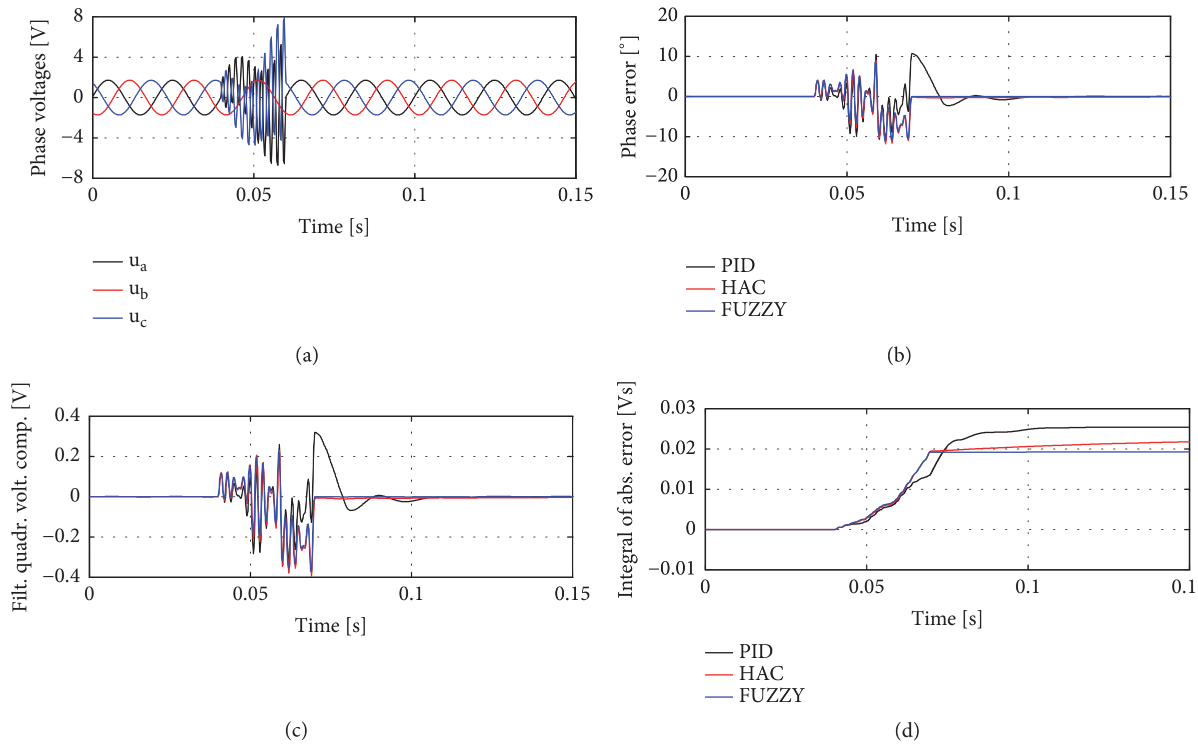

Figure 12: Phase voltages (a), phase error (b), filtered quadrature voltage component $\overline{u}_q$ (c), and integral of absolute error (d) for the unbalanced low-frequency transient in phase voltages.

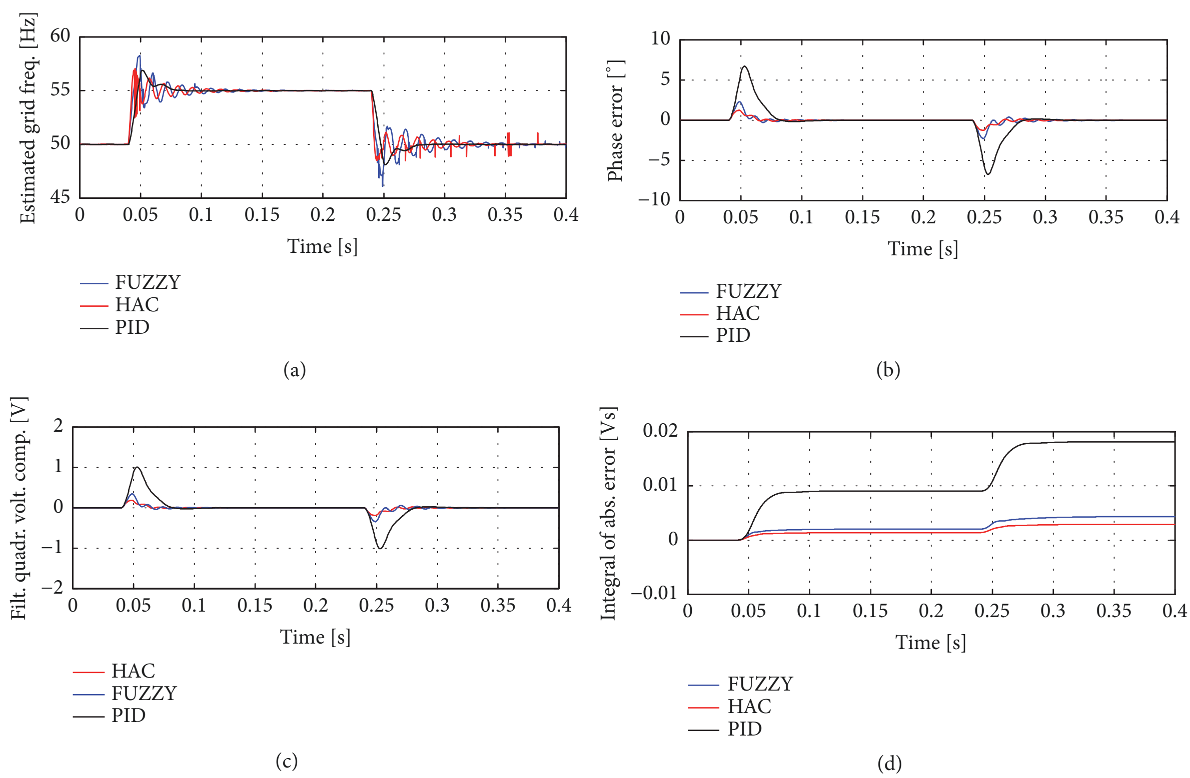

Figure 13: Estimated grid frequency (a), phase error (b), filtered quadrature voltage component $\overline{u}_q$ (c), and integral of absolute error (d) for the step-changes in frequency of ±5 Hz.

harmonics is negligible. Again, the HAC controller ensures the lowest IAE value.

The PLL response to the unbalanced transient in phase voltages with the oscillation frequency of 500 Hz is considered in Figure 12. In this case, and in this case only, the HAC does not ensure the lowest IAE value—it is about 12% higher in comparison with the one obtained by the FC, but it is still lower compared to the PID controller.

Figure 13 shows the PLL response to step-changes in the grid frequency. Figure 13(a) shows the estimated grid frequency obtained by the nonadaptive MAF-PLL (Figure 2) with the PID controller and the estimated grid frequency obtained with the proposed PLL configuration (Figure 4) and with the HAC and FCs. Note that the nonadaptive MAF was here used in combination with the PID controller instead of the adaptive MAF because it was observed that the latter results in slightly higher IAE value.

Choosing the settling time with a 2% criterion (i.e., the time after which the estimated frequency reaches and remains within the limits 49.9 Hz-50.1 Hz), 2.1 times shorter settling time is obtained in the PLL with the PID controller in comparison with the HAC-based PLL (Figure 13(a)). However, the maximum phase error obtained by the HAC is 5.4 times lower than the one obtained by the PID controller (Figure 13(b)). On the other hand, when compared with the FC-based MAF-PLL, the HAC-based PLL has approximately the same settling time (Figure 13(a)) but 1.9 times lower maximum phase error (Figure 13(b)). Once more, the lowest IAE value is achieved by using the HAC-based PLL.

For all analyzed PLLs, the possible occurrence of a DC-offset in phase voltages causes fundamental frequency oscillations in steady state in all estimated variables, causing incorrect operation of the proposed PLL. This effect will be the subject of a future research. On the other hand, a small DC component (up to 50 mV) caused by the A/D conversion process was present in the experiments, but its influence on the PLL performance was shown to be negligible.

5. Conclusion

In this paper, the new MAF-based PLLs with the HAC are successfully applied for the three-phase distorted signals. They can handle different types of disturbances and ensure zero steady-state error. Optimized fuzziness parameters and linguistic rule base of the HAC are successfully obtained by means of the genetic algorithm for each analyzed disturbance. The HAC with its parameters optimized for a certain disturbance can handle other analyzed disturbances as well, although with somewhat slower dynamic responses.

The experimental comparison of the developed HAC-PLLs with both the PID-based MAF-PLL and the FC-based MAF-PLL has been carried out. It is shown that the HAC and the FC provide better dynamic performance in comparison to the PID controller, for all analyzed disturbance types. In addition, the HAC ensures the lowest IAE value for all of the analyzed disturbance types except for the unbalanced low-frequency transients, when the FC achieves the lowest IAE value (12% lower than the HAC). However, in this case, the maximum difference between the phase errors obtained by the HAC and FCs is less than 3°, which is considered negligible.

The proposed PLLs do not work properly when there is a DC-offset present in phase voltages and this will be the subject of a future research. On the other hand, the influence of a small DC component (up to 50 mV) caused by the A/D conversion process is shown to be negligible.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work has been fully supported by Croatian Science Foundation under the project IP-2016-06-3319.

References

[1] S. Golestan, J. M. Guerrero, and J. C. Vasquez, "Three-phase plls: a review of recent advances," *IEEE Transactions on Power Electronics*, vol. 32, no. 3, pp. 1897–1907, 2017.

[2] F. D. Freijedo, J. Doval-Gandoy, Ó. López, and E. Acha, "Tuning of phase-locked loops for power converters under distorted utility conditions," *IEEE Transactions on Industry Applications*, vol. 45, no. 6, pp. 2039–2047, 2009.

[3] S. Golestan, M. Ramezani, J. M. Guerrero, F. D. Freijedo, and M. Monfared, "Moving average filter based phase-locked loops: Performance analysis and design guidelines," *IEEE Transactions on Power Electronics*, vol. 29, no. 6, pp. 2750–2763, 2014.

[4] L. Wang, Q. Jiang, L. Hong, C. Zhang, and Y. Wei, "A novel phase-locked loop based on frequency detector and initial phase angle detector," *IEEE Transactions on Power Electronics*, vol. 28, no. 10, pp. 4538–4549, 2013.

[5] Y. F. Wang and Y. W. Li, "Analysis and digital implementation of cascaded delayed-signal-cancellation PLL," *IEEE Transactions on Power Electronics*, vol. 26, no. 4, pp. 1067–1080, 2011.

[6] K.-J. Lee, J.-P. Lee, D. Shin, D.-W. Yoo, and H.-J. Kim, "A novel grid synchronization PLL method based on adaptive low-pass notch filter for grid-connected PCS," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 292–301, 2014.

[7] I. Carugati, S. Maestri, P. G. Donato, D. Carrica, and M. Benedetti, "Variable sampling period filter PLL for distorted three-phase systems," *IEEE Transactions on Power Electronics*, vol. 27, no. 1, pp. 321–330, 2012.

[8] S. Golestan, M. Monfared, and F. D. Freijedo, "Design-oriented study of advanced synchronous reference frame phase-locked loops," *IEEE Transactions on Power Electronics*, vol. 28, no. 2, pp. 765–778, 2013.

[9] K. W. Martin, "Complex signal processing is not complex," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 9, pp. 1823–1836, 2004.

[10] X. Guo, W. Wu, and Z. Chen, "Multiple-complex coefficient-filter-based phase-locked loop and synchronization technique for three-phase grid-interfaced converters in distributed utility

networks," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 4, pp. 1194–1204, 2011.

[11] S. Chandrasekaran and K. Ragavan, "Adaptive sampling period adjusted sliding DFT for synchronous reference frame PLL," in *Proceedings of the 2014 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pp. 1–4, Mumbai, India, 2014.

[12] Y. Cao, J. Yu, Y. Xu, Y. Li, and J. Yu, "An Efficient Phase-Locked Loop for Distorted Three-Phase Systems," *Energies*, vol. 10, no. 3, 2017.

[13] S. Golestan, J. M. Guerrero, A. Vidal, A. G. Yepes, and J. Doval-Gandoy, "PLL with MAF-based prefiltering stage: small-signal modeling and performance enhancement," *IEEE Transactions on Power Electronics*, vol. 31, no. 6, pp. 4013–4019, 2016.

[14] P. Rodriguez, R. Teodorescu, I. Candela, A. Timbus, M. Liserre, and F. Blaabjerg, "New positive-sequence voltage detector for grid synchronization of power converters under faulty grid conditions," in *Proceedings of the 37th IEEE Power Electronics Specialists Conference*, pp. 1–7, Jeju, Korea, 2006.

[15] F. A. S. Neves, M. C. Cavalcanti, H. E. P. De Souza, F.-C. Bradaschia, E. J. Bueno, and M. Rizo, "A generalized delayed signal cancellation method for detecting fundamental-frequency positive-sequence three-phase signals," *IEEE Transactions on Power Delivery*, vol. 25, no. 3, pp. 1816–1825, 2010.

[16] S. Beheshtaein, "Fuzzy phase locked loop for three-phase power converters," in *Proceedings of the Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE*, 2013.

[17] S. Beheshtaein, M. Savaghebi, and J. M. Guerrero, "A fuzzy-based hybrid PLL scheme for abnormal grid conditions," in *Proceedings of the 41th Annual Conference of IEEE Industrial Electronics Society, IECON 2015*, Yokohama, Japan, 2015.

[18] G. Gowribanu and S. Anbumalar, "Dynamic performance improvement of a moving average filter based-PLL using PI and fuzzy controller," in *Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, India, 2016.

[19] H. Zhang and D. Liu, *Fuzzy Modeling and Fuzzy*, Birkhäuser Basel, 2006.

[20] N. Ho and W. Wechler, "Hedge algebras: an algebraic approach to structure of sets of linguistic truth values," *Fuzzy Sets and Systems*, vol. 35, no. 3, pp. 281–293, 1990.

[21] H.-L. Bui, C.-H. Nguyen, V.-B. Bui, K.-N. Le, and H.-Q. Tran, "Vibration control of uncertain structures with actuator saturation using hedge-algebras-based fuzzy controller," *Journal of Vibration and Control*, vol. 23, no. 12, pp. 1984–2002, 2017.

[22] H.-L. Bui, D.-T. Tran, and N.-L. Vu, "Optimal fuzzy control of an inverted pendulum," *Journal of Vibration and Control*, vol. 18, no. 14, pp. 2097–2110, 2012.

[23] N. D. Anh, H.-L. Bui, N.-L. Vu, and D.-T. Tran, "Application of hedge algebra-based fuzzy controller to active control of a structure against earthquake," *Structural Control and Health Monitoring*, vol. 20, no. 4, pp. 483–495, 2013.

[24] N. D. Duc, N. Vu, D. Tran, and H. Bui, "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. 4, pp. 2186–2200, 2011.

[25] H. L. Bui, T. A. Le, and V. B. Bui, "Explicit formula of hedge-algebras-based fuzzy controller and applications in structural vibration control," *Applied Soft Computing*, vol. 60, pp. 150–166, 2017.

[26] H.-L. Bui, C.-H. Nguyen, N.-L. Vu, and C.-H. Nguyen, "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.

[27] D. Vukadinović, M. Bašić, C. H. Nguyen, N. L. Vu, and T. D. Nguyen, "Hedge-algebra-based voltage controller for a self-excited induction generator," *Control Engineering Practice*, vol. 30, pp. 78–90, 2014.

[28] M. H. J. Bollen and I. Y.-H. Gu, *Signal Processing of Power Quality Disturbances*, IEEE Press, New York, NY, USA, 2006.

[29] M. H. J. Bollen, *Understanding Power Quality Problems: Voltage Sags and Interruptions*, John & Wiley Sons, Inc, Hoboken, NJ, USA, 2000.

[30] C. H. Nguyen and J. M. Alonso, "Looking for a Real-World-Semantics-Based Approach to the Interpretability of Fuzzy Systems," in *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Naples, Italy, 2017.

[31] B. Bouchon-Meunier, "Linguistic hedges and fuzzy logic," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, SD, USA, 1992.

[32] V. Novak, "A horizon shifting model of linguistic hedges for approximate reasoning," in *Proceedings of the IEEE 5th International Fuzzy Systems*, NO, USA, 1996.

[33] C. Y. Chen, C. Y. Huang, and B. D. Liu, "Current-mode fuzzy linguistic hedge circuit - contrast intensification," in *proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 511–513, Atlanta, GA, USA, 1996.

[34] B. D. Liu, C. Y. Chen, and J. Y. Tsao, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 1, pp. 32–53, 2001.

[35] R. Belohlavek and V. Vychodil, "Fuzzy concept lattices constrained by hedges," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 11, no. 6, pp. 536–545, 2007.

[36] R. Belohlavek and V. Vychodil, "Formal concept analysis and linguistic hedges," *International Journal of General Systems*, vol. 41, no. 5, pp. 503–532, 2012.

[37] J. Konecny, J. Medina, and M. Ojeda-Aciego, "Multi-adjoint concept lattices with heterogeneous conjunctors and hedges," *Annals of Mathematics and Artificial Intelligence*, vol. 72, no. 1-2, pp. 73–89, 2014.

[38] L. Antoni, S. Krajči, and O. Krídlo, "Constraint heterogeneous concept lattices and concept lattices with heterogeneous hedges," *Fuzzy Sets and Systems*, vol. 303, pp. 21–37, 2016.

[39] C. H. Nguyen and V. L. Nguyen, "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.

[40] T. A. Nguyen and C. H. Nguyen, "On the real-world-semantics interpretability of fuzzy rule based systems under fuzzy set approach and hedge algebra approach," *Journal of Computer Science and Cybernetics*, vol. 33, no. 1, pp. 89–109, 2017.

[41] C. H. Nguyen, N. L. Vu, and X. V. Le, "Optimal hedge-algebras-based controller: design and application," *Fuzzy Sets and Systems*, vol. 159, pp. 968–989, 2008.

[42] R. C. Dugan, M. F. Mc Granaghan, S. Santoso, and H. W. Beaty, *Electrical Power Systems Quality*, McGraw-Hill, 2nd edition, 2002.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
Advances in Multimedia
Advances in Civil Engineering
Hindawi
Submit your manuscripts at
www.hindawi.com
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration