Properties and Data-driven Design of Perceptual Reasoning Method Based Linguistic Dynamic Systems

LI Cheng-Dong, ZHANG Gui-Qing, WANG Hui-Dong, REN Wei-Na

Abstract

The linguistic dynamic systems (LDSs) based on type-1 fuzzy sets can provide a powerful tool for modeling, analysis, evaluation, and control of complex systems. However, as pointed out in earlier studies, it is much more reasonable to take type-2 fuzzy sets to model the existing uncertainties of linguistic words. In this paper, the LDS based on type-2 fuzzy sets is studied, and its reasoning process is realized through the perceptual reasoning method. The properties of the perceptual reasoning method based LDS (PR-LDS) are explored. These properties demonstrated that the output of PR-LDS is intuitive and the computation complexity can be reduced when the consequent type-2 fuzzy numbers in the rule base satisfy some conditions. Further, a data driven method for the design of the PR-LDS is provided. At last, the effectiveness and rationality of the proposed data-driven method are verified by an example.

Key words

Computing with words, linguistic dynamic system, type-2 fuzzy, perceptual reasoning

Citation


DOI

10.3724/SP.J.1004.2014.02221

In complex systems, problems are usually described and analyzed using natural language. Humans are used to reasoning and calculating on the basis of the premises expressed in natural language. In order to solve the modeling, analysis, control and evaluation problems in complex systems, Wang [1–5] proposed the methodology of linguistic dynamic systems (LDSs). The LDSs aim to dynamically utilize the language level information to handle the issues related to complex systems and adopt fuzzy sets (FSs) to represent the information granularity [3]. In recent years, different theoretical aspects of the LDSs have been profoundly explored, e.g., the fixed points [8], the stability analysis [6–7] and the orbits of the LDS [8]. The key part for the realization of the LDSs is computing with words (CWW). CWW was firstly proposed by Zadeh in 1996 in [9] and is mainly used to deal with perceptual information. In CWW, the operation objects are linguistic variables, i.e., the value of the variable is the word or sentence expressed in natural language.

Since the appearance of CWW, lots of papers and books have focused on this topic [10–18]. In recent years, Mendel [19] and Türkýen [20] pointed out that the linguistic words should be modeled by type-2 fuzzy sets (T2 FSs), as “words mean different things to different people”. Since then, the T2 FSs based CWW has attracted wide attention. In [21–23], Mendel et al. proposed the framework of perceptual computer (PC), and explored the functions of different parts of the PC. In order to solve the reasoning problem in PC, in [24–25], linguistic weighted average (LWA) algorithm was given. Based on this algorithm, in [26–29], a perceptual reasoning (PR) method was proposed. For the T2 FSs based LDSs, Mo et al. [30–31] and Zhao [32–33] have studied the LDS trajectories and their stability issues.

Till now, the CWW methods for LDSs can be divided into two categories. One is based on the extension principle [6–7] and the other is on the basis of the Mamdani reasoning method [32–33]. However, for some complex systems, the extension principle is difficult to use, especially when T2 FSs are adopted. Both kinds of methods suffer another drawback—it is difficult to interpret the reasoning output words from CWW, i.e., the shapes of the output words may be irregular. This will cause the difficulty to understand the output words. On the other hand, when the CWW is applied to LDSs, the computation complexity should be considered. Most existing algorithms are on the basis of α-cuts. This method has one shortcoming that the computation time is long and the efficiency is low.

In order to solve the intuition and computation complexity issues of LDSs, this paper presents the perceptual reasoning method based LDS (PR-LDS). The properties of the PR-LDS are then explored. We will prove that the output words of the PR-LDS are intuitive, and trapezoidal fuzzy numbers can be generated by the PR-LDS when the T2 FSs in the rule base satisfy some conditions. Moreover, we will show how to reduce the computation complexity of the PR-LDS.

To realize the application of the LDS, besides theoretical analysis, there exists another important issue—modeling of the LDS—that needs to be solved. Previous studies [3, 6–8, 31] mainly have paid attention to extending the numeric dynamic systems to LDS through the extension principle. However, when the numeric dynamic model of the system is unknown, the LDS model can not be obtained by the existing method. On the other hand, the data observed from the system can reflect its dynamic characteristics. In this case, we can utilize the observed data to construct the LDS model. To the best of our knowledge, there exist no discussion on this topic. In this study, a data-driven method for the
modeling of LDS is provided. The presented method consists of four steps: word library construction, projection from data to words, decision table construction, and rule generation using the rough set method. Also, simulations are given to verify the proposed data driven method and the PR-LDS. From simulation results, we can see that the linguistic trajectory of the LDS constructed by the proposed method can reflect the dynamic characteristics of the system being identified. The limited loss of precision is acceptable when we want or have to describe the system in a more concise way, especially when we face big data.

The rest of this study is organized as follows. Section 1 briefly reviews the definitions of T1 FSs, T2 FSs, fuzzy numbers and their α-cuts. Section 2 presents the inference procedure and properties of the PR-LDS. Section 3 gives the data-driven design of PR-LDS. Section 4 provides a simulation to show the reasonableness of the proposed method. Finally, Section 5 concludes the paper.

1 Some definitions

1.1 Type-2 and type-1 FSs and fuzzy numbers

Fuzzy sets whose elements have degrees of membership were first introduced by Zadeh[34] in 1965 to depict linguistic words or perceptions. The FS with a crisp membership function (MF) is called type-1 (T1) FS. Commonly, a T1 FS, denoted as $\tilde{A}$, is represented as

\[
\tilde{A} = \int_X \mu_{\tilde{A}}(x)/x
\]

where $X$ is the universe of discourse, $\mu_{\tilde{A}}(x)$ is the crisp MF grade of a generic element $x$. $\int$ denotes union over all admissible $x$.

A type-1 fuzzy number, which is an extension of a real number, is a special case of a convex, normalized T1 FS $\tilde{X}$ in the real space $\mathbb{R}^{35}$. Usually, let $\tilde{Y}$ be a fuzzy number, whose membership function $\mu_{\tilde{Y}}(x)$ can generally be defined as

\[
\mu_{\tilde{Y}}(x) = \begin{cases} 
\tilde{Y}_L(x), & a < x \leq b \\
1, & b < x \leq c \\
\tilde{Y}_R(x), & c < x \leq d \\
0, & \text{else}
\end{cases}
\]

where $\tilde{Y}_L(x)$ and $\tilde{Y}_R(x)$ are two strictly monotonic and continuous mappings from $\mathbb{R}$ to the closed interval $[0,1]$. If the membership function $\mu_{\tilde{Y}}(x)$ is piecewise linear, then $\tilde{Y}$ is referred to as a trapezoidal fuzzy number and is usually denoted by $\tilde{Y} = (a, b, c, d)$.

Recently, in [19, 36], Mendel claimed that words mean different things to different people and T2 FS should be used as a model of a word. A T2 FS is an FS with fuzzy MF. The T2 FS $\tilde{A}$ can be characterized as

\[
\tilde{A} = \int_X \mu_{\tilde{A}}(x)/x = \int_X \int_{u \in J_s} f_s(u)/u / x, J_s \subseteq [0,1]
\]

where $\mu_{\tilde{A}}(x)$ is the fuzzy MF grade of a generic element $x$. $\int \int$ denotes union over all admissible $x$ and $u$, $f_s(u)$ is the secondary MF and $J_s$ is the primary membership of $x$ which is the domain of the secondary MF. When the secondary MFs of T2 FS become interval sets, the T2 FS turns to interval type-2 (IT2) FS that can be characterized as

\[
\tilde{A} = \int_{x \in X} \left[ \left[ \int_{u \in J_s} 1/u \right] / x, J_s \subseteq [0,1] \right]
\]

where the secondary grades of $\tilde{A}$ all equal 1. In this paper, without detailed specification, T2 FS means IT2 FS.

IT2 FS $\tilde{A}$ can be completely described by its lower MF (LMF) $\mu_{\tilde{A}}(x)$ and upper MF (UMF) $\mu_{\tilde{A}}(x)$, i.e.

\[
\mu_{\tilde{A}}(x) = [\mu_{\tilde{A}}(x), \mu_{\tilde{A}}(x)]
\]

Type-2 fuzzy numbers\(^1\) are special T2 FSs. The general type-2 fuzzy numbers adopted in this paper are shown in Fig. 1 (a). We can still depict type-2 fuzzy numbers using their LMFs and UMFs. A type-2 fuzzy number $\tilde{Y}$ can be denoted by its LMF and UMF as

\[
\tilde{Y}(x) = \begin{cases} \tilde{Y}_L(x), & \pi < x \leq \tilde{b} \\
1, & \tilde{b} < x \leq \pi \\
\tilde{Y}_R(x), & \pi < x \leq d \\
0, & \text{else}
\end{cases}
\]

where $\tilde{Y}_L(x)$ and $\tilde{Y}_R(x)$ are two strictly monotonic and continuous mappings from $\mathbb{R}$ to the closed interval $[0,1]$. If the LMF and UMF are piecewise linear, then $\tilde{Y}$ is referred to as a trapezoidal type-2 fuzzy number which is widely used and shown in Fig. 1 (b).

\[
\tilde{Y}_L(x) = \begin{cases} \tilde{Y}_L(x), & \pi < x \leq \tilde{b} \\
1, & \tilde{b} < x \leq \pi \\
\tilde{Y}_R(x), & \pi < x \leq d \\
0, & \text{else}
\end{cases}
\]

\[
\tilde{Y}_R(x) = \begin{cases} \tilde{Y}_L(x), & \pi < x \leq \tilde{b} \\
1, & \tilde{b} < x \leq \pi \\
\tilde{Y}_R(x), & \pi < x \leq d \\
0, & \text{else}
\end{cases}
\]

\(^1\)Because LDS is an extension of the classical dynamic system whose universe of discourse is usually the real space $\mathbb{R}$, in this study, FSs (including both type-1 and type-2 cases) are also limited in $\mathbb{R}$. We use type-1 and type-2 fuzzy numbers which are extensions of real numbers as the models of linguistic words in $\mathbb{R}$.
1.2 \( \alpha \)-cuts of type-2 and type-1 fuzzy numbers

The \( \alpha \)-cut of a type-2 fuzzy number is depicted in Fig. 2. For the type-2 fuzzy number \( \tilde{Y} \) expressed in (6) and (7), its \( \alpha \)-cut consists of four end-points, which are denoted respectively as \( \tilde{Y}_{L}(\alpha), \tilde{Y}_{R}(\alpha), \tilde{Y}_{L'}(\alpha) \) and \( \tilde{Y}_{R'}(\alpha) \).

![Fig. 2 \( \alpha \)-cuts of type-2 fuzzy numbers](image)

When \( \alpha = 0 \), we have \( \tilde{Y}_{L}(0) = \pi, \tilde{Y}_{R}(0) = \overline{d}, \tilde{Y}_{L'}(0) = \overline{d} \) and \( \tilde{Y}_{R'}(0) = d \).

When \( \alpha = 1 \), we have \( \tilde{Y}_{L}(1) = \overline{d}, \tilde{Y}_{R}(1) = \pi, \tilde{Y}_{L'}(1) = \overline{b} \) and \( \tilde{Y}_{R'}(1) = b \).

Similarly, for the type-1 fuzzy number \( \tilde{Y} \) as expressed in (2), its \( \alpha \)-cut consists of two end-points, which are denoted respectively as \( \tilde{Y}_{L}(\alpha) \) and \( \tilde{Y}_{R}(\alpha) \). And, \( \tilde{Y}_{L}(0) = a, \tilde{Y}_{R}(0) = d, \tilde{Y}_{L}(1) = b \) and \( \tilde{Y}_{R}(1) = c \).

2 Inference procedure and properties of PR-LDS

2.1 PR-LDS

Suppose that the following type-2 fuzzy rule base is adopted in the LDS:

\[
\begin{align*}
R^i: & \ x(k) = \tilde{A}_{1}, \ldots, x(k-s+1) = \tilde{A}_{s}, u(k) = \tilde{U}_{1}, \ldots, \\ & u(k-t+1) = \tilde{U}_{t} \ \Rightarrow \ x(k+1) = \tilde{Y}^i \bigwedge_{i=1}^{M}
\end{align*}
\]

where \( \tilde{A}_{1}, \ldots, \tilde{A}_{s}, \tilde{Y}^i \) are linguistic words modeled by T2 FSs in the universe of discourse of the state variable \( x \). \( \tilde{U}_{1}, \ldots, \tilde{U}_{t} \) are linguistic words modeled by T2 FSs in the universe of discourse of the control variable \( u \). \( s+t = p \). \( \tilde{Y}^i \) is the output word of rule \( R^i \) and \( M \) is the number of fuzzy rules in the LDS.

Assume that at time \( k \), the previous system states and control words are respectively \( (\tilde{X}_0, \tilde{X}_1, \ldots, \tilde{X}_{k-1}, \tilde{X}_k) \) and \( (\tilde{U}_0, \tilde{U}_1, \ldots, \tilde{U}_{k-1}, \tilde{U}_k) \). To reflect the system characteristic, we need to calculate the next state \( \tilde{X}_{k+1} \), i.e., the output word \( (T_2 \text{FS}) \) \( \tilde{X}_{k+1} \) of the LDS should be computed when \( x(k) = \tilde{X}_k, \ldots, x(k-s+1) = \tilde{X}_{k-s+1}, u(k) = \tilde{U}_k, \ldots, u(k-t+1) = \tilde{U}_{k-t+1} \).

Different methods can be adopted to realize this dynamic inference, e.g. the Mamdani reasoning method used in [32–33]. However, as pointed in the introduction, existing methods have their drawbacks. From the results in [32–33] (see Fig. 3 for example), we can see that the Mamdani reasoning method based LDS gives irregular T2 FSs which are difficult to be interpreted. In this study, we utilize a new reasoning method named perceptual reasoning to achieve the intuition of the output word. Below, we will discuss this issue in detail.

\[
\text{Step 1. Computing the firing strength of each rule}
\]

Once the linguistic vector \( (\tilde{X}_k, \ldots, \tilde{X}_{k+s-1}, \tilde{U}_k, \ldots, \tilde{U}_{k+t-1}) \) is input to the CWW engine at time \( k \), then, the firing strength of Rule \( i \) is an interval \( F^i(k) = [\tilde{Y}^i(k), \tilde{T}^i(k)] \), which can be calculated as

\[
\begin{align*}
\tilde{Y}^i(k) &= \left[ \bigwedge_{j=1}^{N} \left( \sup_{x \in X} \mu_{\tilde{X}_{k+j-1}}(x) \wedge \mu_{\tilde{A}_{j}}(x) \right) \right] \wedge \\
\tilde{T}^i(k) &= \left[ \bigwedge_{j=1}^{N} \left( \sup_{u \in U} \mu_{\tilde{U}_{k+j-1}}(u) \wedge \mu_{\tilde{U}_{j}}(u) \right) \right]
\end{align*}
\]

\[
\text{Step 2. Aggregating the fired rules by the linguistic weighted average method}
\]

In the perceptual reasoning method [26–28], the linguistic weighted average algorithm [24–25] is adopted to aggregate the consequents of the fired rules to obtain the output word \( (T_2 \text{FS}) \), i.e.,

\[
\tilde{X}_{k+1} = \frac{\sum_{i=1}^{M} F^i(k) \tilde{Y}^i}{\sum_{i=1}^{M} F^i(k)}
\]

where \( F^i(k) \) is an interval, \( \tilde{Y}^i \) is the consequent T2 FS of Rule \( i \). This equation can be computed through the \( \alpha \)-cuts method. Fig. 2 shows us an example on the \( \alpha \)-cuts of the T2 FS \( \tilde{Y} \). The \( \alpha \)-cut of its UMF can be represented as \( [\tilde{Y}_{L}(\alpha), \tilde{Y}_{R}(\alpha)] \), while the \( \alpha \)-cut of its LMF can be represented as \( [\tilde{Y}_{L}(\alpha), \tilde{Y}_{R}(\alpha)] \).

From the results in [24–29], the left-end and right-end points of the \( \alpha \)-cuts of the output T2 FS can be computed as
\[
(\tilde{X}_{k+1})_{L}(\alpha) = \min_{\forall f' \in [L(k),T(k)]} \sum_{i=1}^{M} \tilde{Y}_{i}^{L}(\alpha)f'(k)
\] (12)

\[
(\tilde{X}_{k+1})_{R}(\alpha) = \max_{\forall f' \in [L(k),T(k)]} \sum_{i=1}^{M} \tilde{Y}_{i}^{R}(\alpha)f'(k)
\] (13)

\[
(\tilde{X}_{k+1})_{L}(\alpha) = \min_{\forall f' \in [L(k),T(k)]} \sum_{i=1}^{M} \tilde{Y}_{i}^{L}(\alpha)f'(k)
\] (14)

\[
(\tilde{X}_{k+1})_{R}(\alpha) = \max_{\forall f' \in [L(k),T(k)]} \sum_{i=1}^{M} \tilde{Y}_{i}^{R}(\alpha)f'(k)
\] (15)

where \( \alpha \in [0,1] \). We can use the Karnik-Mendel algorithm\(^{36}\) to compute these equations.

For the type-1 case, the output word can be similarly computed as

\[
(\tilde{X}_{k+1})_{L}(\alpha) = \frac{\sum_{i=1}^{M} \tilde{Y}_{i}^{L}(\alpha)f'(k)}{\sum_{i=1}^{M} f'(k)}
\] (16)

\[
(\tilde{X}_{k+1})_{R}(\alpha) = \frac{\sum_{i=1}^{M} \tilde{Y}_{i}^{R}(\alpha)f'(k)}{\sum_{i=1}^{M} f'(k)}
\] (17)

where \( \alpha \in [0,1] \) and \( f'(k) \) can be calculated as

\[
f'(k) = \left[ \bigwedge_{j=1}^{N} \left( \sup_{x \in X} \mu_{X_{k-j+1}}(x) \wedge \mu_{X_{j}}(x) \right) \right] \wedge
\left[ \bigwedge_{j=1}^{N} \left( \sup_{u \in U} \mu_{U_{k-j+1}}(u) \wedge \mu_{U_{j}}(u) \right) \right]
\] (18)

\[\text{2.3 Properties of the PR-LDS}\]

For the PR-LDS, we have the following properties which show us the reasonableness and the advantages of this LDS.

\[\text{2.3.1 Intuition of the reasoning output word}\]

For CWW, especially for the LDS, the intuition of the reasoning output word is quite important, because the intuition represents the interpretability of the output word. For the PR-LDS, we have the following property which can assure the intuition of the reasoning output word.

**Theorem 1.** If the consequent T2 FISs \( \tilde{Y}^{1}, \tilde{Y}^{2}, \ldots, \tilde{Y}^{M} \) in the type-2 fuzzy rule base are all type-2 fuzzy numbers, then the output word \( \tilde{X}_{k+1} \) of the PR-LDS is still a type-2 fuzzy number.

**Proof.** In order to prove that the output word \( \tilde{X}_{k+1} \) is still a type-2 fuzzy number, we need to prove the following:

1) The eight vertexes \( \bar{\pi}^{k+1}, \bar{b}^{k+1}, \bar{a}^{k+1}, \bar{d}^{k+1}, \bar{a}^{k+1}, \bar{d}^{k+1}, \bar{a}^{k+1}, \bar{d}^{k+1} \) of \( \tilde{X}_{k+1} \) satisfy the vertex arrangement of type-2 fuzzy number.

2) i) \( \overline{\pi}_{X_{k+1}} \) is strictly monotonically increasing in \([\bar{a}^{k+1}, \bar{b}^{k+1}]\), ii) \( \overline{X}_{X_{k+1}} \) equals 1 in \([\bar{b}^{k+1}, \bar{a}^{k+1}]\), iii) \( \overline{\pi}_{X_{k+1}} \) is strictly monotonically decreasing in \([\bar{a}^{k+1}, \bar{d}^{k+1}]\), iv) \( \overline{\mu}_{X_{k+1}} \) is strictly monotonically increasing in \([\bar{a}^{k+1}, \bar{d}^{k+1}]\), v) \( \overline{\mu}_{X_{k+1}} \) equals 1 in \([\bar{b}^{k+1}, \bar{a}^{k+1}]\), vi) \( \overline{\mu}_{X_{k+1}} \) is strictly monotonically decreasing in \([\bar{a}^{k+1}, \bar{d}^{k+1}]\).

Below, let us prove 1) and 2) respectively.

1) Since \( \tilde{Y}^{i} \) is a type-2 fuzzy number, we have

\[
\tilde{Y}_{L}^{i}(0) \leq \tilde{Y}_{R}^{i}(1) \leq \tilde{Y}_{R}^{i}(0)
\]

So, from (12) and (13), we can derive that

\[
(\tilde{X}_{k+1})_{L}(0) \leq (\tilde{X}_{k+1})_{L}(1) \leq (\tilde{X}_{k+1})_{R}(1) \leq (\tilde{X}_{k+1})_{R}(0)
\]

which means that \( a^{k+1} \leq b^{k+1} \leq c^{k+1} \leq d^{k+1} \).

Similarly, \( a^{k+1} \leq b^{k+1} \leq c^{k+1} \leq d^{k+1} \).

As \( \tilde{Y}_{L}^{i}(0) \leq \tilde{Y}_{R}^{i}(1) \), then from (12) and (14), we have

\[
(\tilde{X}_{k+1})_{L}(0) \leq (\tilde{X}_{k+1})_{L}(0) \leq (\tilde{X}_{k+1})_{R}(0),
\]

which implies that \( a^{k+1} \leq b^{k+1} \leq c^{k+1} \leq d^{k+1} \).

From the above discussion, we can conclude that 1) stands.

2) We just prove i), as the other five cases can be proved in a similar way.

First, note that \( \overline{\pi}_{Y^{i}} \) is strictly monotonically increasing in \([\tilde{Y}_{L}^{i}(0), \tilde{Y}_{R}^{i}(1)]\). Hence, for \( \forall \alpha \in [0,1] \), we have

\[
\tilde{Y}_{L}^{i}(\alpha_{2}) > \tilde{Y}_{L}^{i}(\alpha_{1})
\]

From (12),

\[
(\tilde{X}_{k+1})_{L}(\alpha_{2}) = \min_{\forall f' \in [L(k),T(k)]} \sum_{i=1}^{M} \tilde{Y}_{i}^{L}(\alpha_{2})f'(k)
\]

Without loss of generality, suppose that

\[
(\tilde{X}_{k+1})_{L}(\alpha_{2}) = \frac{\sum_{i=1}^{L} \tilde{T}(k) \tilde{Y}_{L}^{i}(\alpha_{2}) + \sum_{i=L+1}^{M} f'(k) \tilde{Y}_{L}^{i}(\alpha_{2})}{\sum_{i=1}^{L} \tilde{T}(k) + \sum_{i=L+1}^{M} f'(k)}
\]

Equations (19) and (21) imply that

\[
(\tilde{X}_{k+1})_{L}(\alpha_{2}) \geq \frac{\sum_{i=1}^{L} \tilde{T}(k) \tilde{Y}_{L}^{i}(\alpha_{1}) + \sum_{i=L+1}^{M} f'(k) \tilde{Y}_{L}^{i}(\alpha_{1})}{\sum_{i=1}^{L} \tilde{T}(k) + \sum_{i=L+1}^{M} f'(k)}
\]

\[
\min_{\forall f' \in [L(k),T(k)]} \sum_{i=1}^{M} \tilde{Y}_{i}^{L}(\alpha_{1})f'(k) = (\tilde{X}_{k+1})_{L}(\alpha_{1})
\]

As a result, if \( \alpha_{2} \neq \alpha_{1} \), then \( (\tilde{X}_{k+1})_{L}(\alpha_{2}) \neq (\tilde{X}_{k+1})_{L}(\alpha_{1}) \).

Consequently, the function \( (\tilde{X}_{k+1})_{L}(\alpha) \) with respect to variable \( \alpha \) is bijective. Thus, its inverse
function $\Pi_{X_{k+1}} = (\tilde{X}_{k+1})_{L_1}$ is strictly monotonically increasing in $[(\tilde{X}_{k+1})_{L_1}(0), (\tilde{X}_{k+1})_{L_1}(1)]$, i.e., $|a^{k+1}, b^{k+1}|$.

From 1) and 2), we can conclude that the output word of the PR-LDS is still a type-2 fuzzy number.

From this property, we have the following corollary for the type-1 case.

**Corollary 1.** If the consequent T1 FUs $\tilde{Y}^1, \tilde{Y}^2, \ldots, \tilde{Y}^M$ in the type-1 fuzzy rule base are all type-1 fuzzy numbers, then the output word $\tilde{X}_{k+1}$ of the PR-LDS is still a type-1 fuzzy number.

From Theorem 1, we can know that if the consequent T2 FSs in the rule base are all trapezoidal type-2 fuzzy numbers, the output word of the PR-LDS is a type-2 fuzzy number. However, we cannot judge that the output word of the PR-LDS is a trapezoidal type-2 fuzzy number which is the most intuitive compared with other type-2 fuzzy numbers. But, for the type-1 PR-LDS, this is true according to the following theorem.

**Theorem 2.** If the trapezoidal type-1 fuzzy numbers $\tilde{Y}^1 = (a^1, b^1, c^1, d^1), \tilde{Y}^2 = (a^2, b^2, c^2, d^2), \ldots, \tilde{Y}^M = (a^M, b^M, c^M, d^M)$ are adopted in the consequent parts of the type-1 fuzzy rule base, then the PR-LDS outputs trapezoidal type-1 fuzzy number $\tilde{X}_{k+1} = (a^{k+1}, b^{k+1}, c^{k+1}, d^{k+1})$ which can be computed by the following closed-form equations

$$a^{k+1} = \frac{\sum_{i=1}^{M} a^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (23)$$

$$b^{k+1} = \frac{\sum_{i=1}^{M} b^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (24)$$

$$c^{k+1} = \frac{\sum_{i=1}^{M} c^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (25)$$

$$d^{k+1} = \frac{\sum_{i=1}^{M} d^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (26)$$

where $f^i(k)$ can be computed by (18).

**Proof.** For the consequent trapezoidal fuzzy number $\tilde{Y}^i = (a^i, b^i, c^i, d^i)$, the left $\alpha$-cut of $\tilde{Y}^i$ can be computed as

$$\tilde{Y}^i_L(\alpha) = a^i + (b^i - a^i)\alpha \quad (27)$$

Then, using (16), the left $\alpha$-cut of $\tilde{X}_{k+1}$ can be derived as

$$(\tilde{X}_{k+1})_L(\alpha) = \frac{\sum_{i=1}^{M} [a^i + (b^i - a^i)\alpha] f^i(k)}{\sum_{i=1}^{M} f^i(k)} = \frac{\sum_{i=1}^{M} a^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} + \alpha \frac{\sum_{i=1}^{M} (b^i - a^i) f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (28)$$

which means that the left side of the output type-1 word is a straight line, and

$$a^{k+1} = (\tilde{X}_{k+1})_L(0) = \frac{\sum_{i=1}^{M} a^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (29)$$

$$b^{k+1} = (\tilde{X}_{k+1})_L(1) = \frac{\sum_{i=1}^{M} b^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (30)$$

Similarly, we can derive that the right side of the output type-1 word is a straight line, and

$$c^{k+1} = (\tilde{X}_{k+1})_R(1) = \frac{\sum_{i=1}^{M} c^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (31)$$

$$d^{k+1} = (\tilde{X}_{k+1})_R(0) = \frac{\sum_{i=1}^{M} d^i f^i(k)}{\sum_{i=1}^{M} f^i(k)} \quad (32)$$

All these results ensure that the output word is a trapezoidal type-1 fuzzy number.

Hence, this theorem holds.

From this theorem, for the type-1 PR-LDS, we have closed-form equations to compute the output word, and the $\alpha$-cut method is not needed in this case. Therefore, the computational complexity can be reduced and the theoretical analysis can be easier.

**2.3.2 Example**

In this example, we take into account nine linguistic words in the real space. Fig. 4 shows the nine linguistic words, which are “about $-4$", “about $-3$", “about $-2$", “about $-1$", “about $0$", “about $1$", “about $2$", “about $3$", “about $4$". For simplicity, we denote the linguistic word “about $k$" as $\tilde{K}$, where $k \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$. The UMF and LMF of the linguistic word $\tilde{K}$ can be depicted as

$$\Pi_{\tilde{K}}(x) = \begin{cases} \frac{x-k+1}{0.8}, & k-1 \leq x \leq k-0.2 \\ 1, & k-0.2 < x \leq k+0.2 \\ \frac{k+1-x}{0.8}, & k+0.2 < x \leq k+1 \\ 0, & \text{else} \end{cases} \quad (33)$$

$$\mu_{\tilde{K}}(x) = \begin{cases} \frac{x-k+0.5}{0.5}, & k-0.5 < x \leq k \\ \frac{k+0.5-x}{0.5}, & k < x \leq k+0.5 \\ 0, & \text{else} \end{cases} \quad (34)$$
Consider the LDS with the following rules:

\[
\begin{align*}
    x(k) &= -4 \Rightarrow x(k + 1) = -2 \\
    x(k) &= -3 \Rightarrow x(k + 1) = -1 \\
    x(k) &= -2 \Rightarrow x(k + 1) = -1 \\
    x(k) &= -1 \Rightarrow x(k + 1) = 0 \\
    x(k) &= 0 \Rightarrow x(k + 1) = 0 \\
    x(k) &= 1 \Rightarrow x(k + 1) = 0 \\
    x(k) &= 2 \Rightarrow x(k + 1) = 1 \\
    x(k) &= 3 \Rightarrow x(k + 1) = 1 \\
    x(k) &= 4 \Rightarrow x(k + 1) = 2
\end{align*}
\]

(35)

Assume that the initial linguistic word of this LDS is \( X(0) \) which is shown in Fig. 5. Its UMF and LMF are

\[
\begin{align*}
    \mu_{X(0)}(x) &= \mu_{X(0)}(x, 2, 2, 3, 3, 9, 4, 8) \\
    \bar{\mu}_{X(0)}(x) &= \bar{\mu}_{X(0)}(x, 3, 3, 5, 3, 5, 4)
\end{align*}
\]

(36) (37)

in the trajectory are shown in Fig. 5, where we can observe that the output word of the PR-LDS is intuitive. This means that the output word can be interpreted easily.

Theorem 1 does not assure that the output word of the PR-LDS is a trapezoidal type-2 fuzzy number even if trapezoidal type-2 fuzzy numbers are used in the consequent parts of the type-2 fuzzy rule base. However, from this example, we can observe that if the slopes of the consequent trapezoidal type-2 fuzzy numbers satisfy some conditions, the output word of the PR-LDS is still a trapezoidal type-2 fuzzy number. This conclusion is summarized in the following property.

### 2.3.3 Fast calculation property

To begin, let us denote the slopes of the trapezoidal type-2 fuzzy numbers satisfying the following constraints:

\[
\begin{align*}
    k_{Lr}(\tilde{Y}) &= \frac{1}{Y_{Lr}(1) - Y_{Lr}(0)} \\
    k_{Rl}(\tilde{Y}) &= \frac{1}{Y_{Rl}(1) - Y_{Rl}(0)} \\
    k_{Ll}(\tilde{Y}) &= \frac{1}{Y_{Ll}(1) - Y_{Ll}(0)} \\
    k_{Rr}(\tilde{Y}) &= \frac{1}{Y_{Rr}(1) - Y_{Rr}(0)}
\end{align*}
\]

(38) (39) (40) (41)

Then, we have the following property.

**Theorem 3.** If the slopes of the consequent trapezoidal type-2 fuzzy numbers satisfy the following constraints:

\[
\begin{align*}
    k_{Lr}(\tilde{Y}^1) &= k_{Lr}(\tilde{Y}^2) = \cdots = k_{Lr}(\tilde{Y}^M) = k_{Ll}, \\
    k_{Rl}(\tilde{Y}^1) &= k_{Rl}(\tilde{Y}^2) = \cdots = k_{Rl}(\tilde{Y}^M) = k_{Rl}, \\
    k_{Ll}(\tilde{Y}^1) &= k_{Ll}(\tilde{Y}^2) = \cdots = k_{Ll}(\tilde{Y}^M) = k_{Ll}, \\
    k_{Rr}(\tilde{Y}^1) &= k_{Rr}(\tilde{Y}^2) = \cdots = k_{Rr}(\tilde{Y}^M) = k_{Rr},
\end{align*}
\]

and \( k_{Ll}(\tilde{X}) = k_{Lr}(\tilde{X}) = k_{Rl}(\tilde{X}) = k_{Rr}(\tilde{X}) \), then the output word \( \tilde{X}_{k+1} \) of the PR-LDS is still a trapezoidal type-2 fuzzy number, whose slopes are

\[
\begin{align*}
    k_{Ll}(\tilde{X}_{k+1}) &= k_{Ll}, \\
    k_{Rl}(\tilde{X}_{k+1}) &= k_{Rl}, \\
    k_{Lr}(\tilde{X}_{k+1}) &= k_{Lr}, \\
    k_{Rr}(\tilde{X}_{k+1}) &= k_{Rr}.
\end{align*}
\]

**Proof.** If \( \tilde{Y}^i \) is a trapezoidal type-2 fuzzy number, then

\[
\tilde{Y}_{Ll}(\alpha) = [\tilde{Y}_{Ll}(1) - \tilde{Y}_{Ll}(0)]\alpha + \tilde{Y}_{Ll}(0) = \frac{1}{k_{Ll}}\alpha + \tilde{Y}_{Ll}(0)
\]

(42)

Hence,

\[
\begin{align*}
    (\tilde{X}_{k+1})_{Ll}(\alpha) &= \min_{\forall f^i \in [f^i(0), f^i(0)]} \sum_{i=1}^{M} f^i \\
    &= \sum_{i=1}^{M} \min \left[ \frac{1}{k_{Ll}}\alpha + \tilde{Y}_{Ll}(0) \right] f^i
\end{align*}
\]

(43)
adopted to generate the fuzzy rules of the LDS.

As stated previously, LDS provides us a useful tool for these applications. Building the LDS model is one problem that can not be avoided by us. Although we can not construct the LDS model directly sometimes, usually observed data can not been avoided by us. Although we can not construct a method to construct the LDS model based on these data pairs. The procedure of the presented method is shown in Fig. 6. This data-driven method includes four main steps which will be given in detail below.

\[ \min \left\{ \frac{1}{k_{LI}} \alpha + \min_{f \in [f \hat{(k)}, T \hat{(k)}]} \left\{ \frac{1}{M} \sum_{i=1}^{M} \tilde{y}_{L}(0) f^i \right\} \right\} = \]

\[ \frac{1}{k_{LI}} \alpha + \min \left\{ \frac{1}{M} \sum_{i=1}^{M} \tilde{y}_{L}(0) f^i \right\} = \]

\[ \frac{1}{k_{LI}} \alpha + b_{LI} \]

where \( b_{LI} = \min_{f \in [f \hat{(k)}, T \hat{(k)}]} \left\{ \frac{1}{M} \sum_{i=1}^{M} \tilde{y}_{L}(0) f^i \right\} \) is a constant.

From this equation, we know that the left side of the output word \( \tilde{x}_{L+1} \) is a straight line, whose slope can be computed as follows

\[ k_{LI}(\tilde{x}_{L+1}) = \frac{\alpha_2 - \alpha_1}{(\tilde{x}_{L+1} \hat{i} L)(\alpha_2) - (\tilde{x}_{L+1} \hat{i} L)(\alpha_1)} = \]

\[ \frac{\alpha_2 - \alpha_1}{(\tilde{x}_{L+1} \hat{i} L)(\alpha_2) + b_{LI} - (\tilde{x}_{L+1} \hat{i} L)(\alpha_1) + b_{LI})} = \]

\[ k_{LI} \]

Similarly, we can prove that

\[ k_{Ri}(\tilde{x}_{L+1}) = k_{Ri} \]

\[ k_{Li}(\tilde{x}_{L+1}) = k_{Li} \]

\[ k_{Ri}(\tilde{x}_{L+1}) = k_{Ri} \]

\[ k_{Li}(\tilde{x}_{L+1}) = k_{Li} \]

From the above discussion, this property holds. \( \Box \)

Remark. If the slopes of the consequent trapezoidal type-2 fuzzy numbers satisfy the constraints in Theorem 3, then to determine the output word \( \tilde{x}_{L+1} \) of the PR-LDS, we only need to compute its four vertices \( (\tilde{x}_{L+1} \hat{i} L)(0), (\tilde{x}_{L+1} \hat{i} R)(0), (\tilde{x}_{L+1} \hat{i} L)(0), \) and \( (\tilde{x}_{L+1} \hat{i} R)(0) \). There is no need to compute the end points of the \( \alpha \)-cuts of the output word. So, the computation complexity and the computation time can be reduced greatly. From the above analysis, we can adopt the rule base that satisfies the constraints in Theorem 3 to reduce the computation complexity of the PR-LDS.

3 Data driven design of PR-LDS

In some real-world applications, it is quite difficult to obtain their exact mathematical models, especially for the complex systems that can be affected by human activities. As stated previously, LDS provides us a useful tool for these applications. Building the LDS model is one problem that can not be avoided by us. Although we can not construct the LDS model directly sometimes, usually observed data can reflect the characteristics of such systems can be adopted to generate the fuzzy rules of the LDS.

Suppose that the observed state sequence is \( \tilde{x}(1), \tilde{x}(2), \tilde{x}(3), \ldots, \tilde{x}(N) \), and the observed control sequence is \( \tilde{u}(1), \tilde{u}(2), \tilde{u}(3), \ldots, \tilde{u}(N) \). Below, we will provide a method to construct the LDS model based on these data pairs. The procedure of the presented method is shown in Fig. 6. This data-driven method includes four main steps which will be given in detail below.

![Fig. 6 The proposed LDS rule base generation method](image)

**Step 1. Word library construction**

We should first construct the word library for the state and control variables. This can be achieved through manual or trained fuzzy partitioning, fuzzy clustering, the fuzzistics methods\(^{[17, 38-41]}\), and the interval approach\(^{[42-45]}\). Assume that the constructed word library for the state variable \( x \) is \( \{A_1, A_2, \ldots, A_T\} \) and the constructed word library for the control variable \( u \) is \( \{\tilde{U}_1, \tilde{U}_2, \ldots, \tilde{U}_r\} \).

In this step, we should consider the levels of the information granularity, i.e. the width of the T2 FSs, which can represent the knowledge in data at various levels of resolution or scales. By focusing on high levels of granularity, we can obtain high levels of knowledge and easy-understood knowledge structure. But, usually, we will lose the precision to some extent.

**Step 2. Projection from data to word**

For each observed state data \( \tilde{x}(k) \), we find the type-2 fuzzy number that has the greatest membership grade with respect to \( \tilde{x}(k) \), i.e.,

\[ \tau_x: R \rightarrow F(X) \]

\[ \tilde{x}(k) \rightarrow \tau_x(\tilde{x}(k)) = \tilde{A}_{i_k} \]

where

\[ i_k = \arg \max_{i=1, \ldots, T} \frac{1}{2} \left( \mu_{A_i}(\tilde{x}(k)) + \pi_{A_i}(\tilde{x}(k)) \right) \]

Again, for the observed control data \( \tilde{u}(k) \), we find the type-2 fuzzy number that has the greatest membership grade with respect to \( \tilde{u}(k) \), i.e.,

\[ \tau_u: R \rightarrow F(U) \]

\[ \tilde{u}(k) \rightarrow \tau_u(\tilde{u}(k)) = \tilde{U}_{j_k} \]

where

\[ j_k = \arg \max_{j=1, \ldots, r} \frac{1}{2} \left( \mu_{U_j}(\tilde{x}(k)) + \pi_{U_j}(\tilde{x}(k)) \right) \]

From this step, observed data can be mapped onto the words in the word library.
Step 3. Constructing the decision table

According to the observed states sequence \( \tilde{x}(1), \tilde{x}(2), \tilde{x}(3), \ldots \), and the observed control sequence \( \tilde{u}(1), \tilde{u}(2), \tilde{u}(3), \ldots \), we can obtain the following data pairs for the LDS expressed in (8):

\[
\begin{align*}
[\tilde{x}(L), \ldots, \tilde{x}(L-s+1), \tilde{u}(L), \ldots, \tilde{u}(L-t+1); \tilde{x}(L+1)] \\
[\tilde{x}(N-1), \ldots, \tilde{x}(N-s), \tilde{u}(N-1), \ldots, \tilde{u}(N-t); \tilde{x}(N)]
\end{align*}
\]

where \( L = \max\{s, t\} \).

From (48) \( \sim \) (51), these data pairs can be mapped onto the word pairs and can be listed in a decision table shown in Table 1. The decision table is formally denoted as \( S = \langle U, C \cup \{d\} \rangle \), where \( U = \{X_1, \ldots, X_{N-L}\} \) is a finite set of objects, \( C = \{x(k), \ldots, x(k-s+1), u(k), \ldots, u(k-t+1)\} \) is a finite set of condition attributes, and \( d = x(k+1) \) is a decision attribute.

Step 4. Fuzzy rules generation using the rough set method

The rough set theory provides us a powerful tool to induce rules from the decision table. Generally speaking, we use the following two steps to generate rules:

1) Attribute reduction. In the decision table, not all the condition attributes are needed to discern the objects. The rough set theory has the ability to identify the minimal subset of attributes required to fully describe the information in the table. In the rule generation application, the minimal subsets of attributes are especially useful as fewer attributes may be more helpful for inducing more general and interpretable rules. Today, many algorithms can be adopted to realize the attribute reduction, e.g., the positive approximation method, the minimum cost method, etc. Surely, this step can be omitted if the attributes are determined or chosen previously by the user or engineer.

2) Rule generation. A number of various algorithms, e.g., the LERS method and the incremental object based alternative rule induction method, have been already proposed for the induction of decision rules from decision tables. In our study, the classical LERS algorithm introduced by Grzymala is adopted. This rule generation algorithm obtains a first rule from the decision table. Then, the learning examples that match this rule are removed from consideration. The process is repeated iteratively till some significant examples remain still uncovered. Some rules may conflict with each other, i.e., they have the same antecedent part but different consequent parts. In our algorithm, we are interested in discovering the strongest fuzzy rules. So, once the conflicting rules appear, we delete those rules with smaller strengths.

4 Simulation

In this section, we will verify the data driven design method and the performance of the PR-LDS. To do this, we use the Mackey-Glass time-series prediction problem as an example.

4.1 Problem description

The Mackey-Glass time series is a typical dynamic system and can be generated by the following differential equation:

\[
\dot{x}(t) = \frac{0.2x(t-\tau)}{1 + x^10(t-\tau)} - 0.1x(t)
\]

When \( \tau > 17 \), this dynamic system can appear the chaotic characteristic.

In our simulation, we use four input variables \( x(k-4), x(k-3), x(k-2), x(k-1) \) to forecast the value \( x(k) \), i.e., we construct the PR-LDS for this system to model the function \( f(\cdot) \), where

\[
x(k) = \tilde{f}(x(k-4), x(k-3), x(k-2), x(k-1))
\]

And, as shown in Fig. 7, we extract 800 input-output data points of the following format:

\[
[x(k-4), x(k-3), x(k-2), x(k-1); x(k)]
\]

The first 500 data points are used to construct the PR-LDS by the proposed data driven method, while the left 300 data points are used for evaluation.

![Fig. 7 The training and testing data](image-url)
Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the

Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the

Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the

Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the

Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the

Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the

Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the

Using the data-driven method proposed in the previous section, we get the type-2 fuzzy rule base for the PR-LDS. And, 60 fuzzy dynamic rules are obtained and shown in Fig. 9. From the word library, we observe that the slopes of the triangular type-2 fuzzy numbers are the same. Hence, we can use the fast calculation property to accelerate the dynamic inference speed of the PR-LDS. For the testing data, the performance of the PR-LDS is shown in Figs. 10 and 11. Fig. 10 demonstrates the bounded trajectory of the
LDS, we presented a data-driven strategy which includes trapezoidal words. To realize the application of the PR-LDS, we gave closed-form equations to compute the output transition complexity of the PR-LDS. Especially, for the type-1 hand, this paper also told us how to reduce the computation of the rule base are all type-2 fuzzy numbers. On the other hand, this paper also showed that the consequent T2 FSs in the rule base is still a type-2 fuzzy number if the consequent T2 FSs in the word library are optimized. Better performance can be expected if the type-2 fuzzy numbers in the word library are optimized using optimization methods.

2) Different levels of information granularity may lead to different performances. The PR-LDS with low level of information granularity (more T2 FSs in the word library) performs more precisely than the PR-LDS with high level of information granularity (less T2 FSs in the word library). But, the PR-LDS with high level of information granularity has much less fuzzy rules, so it is more transparent and can be interpreted more easily.

3) Compared with the numeric dynamic systems, the LDS adopts fuzzy rules to depict the dynamic characteristic, which makes the modeling more designer-oriented. As we discussed, the LDS utilizes the information granules (T2 FSs in this study) to depict the studied system. This is important and beneficial, because the users can always be confused by too many details, especially in this big data era.

5 Conclusion

In this study, we presented the perceptual reasoning method based linguistic dynamic systems (PR-LDS). And, the explored properties demonstrated that the output word of PR-LDS is intuitive, i.e. the output word of the PR-LDS is still a type-2 fuzzy number if the consequent T2 FSs in the rule base are all type-2 fuzzy numbers. On the other hand, this paper also told us how to reduce the computation complexity of the PR-LDS. Especially, for the type-1 case, we gave closed-form equations to compute the output trapezoidal words. To realize the application of the PR-LDS, we presented a data-driven strategy which includes four steps, from data collection, construction of word library and decision table to rule base generation. Simulation results verified the reasonableness and effectiveness of the proposed PR-LDS and the data-driven method. In the future, we will apply the proposed method to real-world applications.

References


27. Mendel J M, Wu D R. Challenges for perceptual computer applications and how they were overcome. IEEE Computational Intelligence Magazine, 2012, 7(3): 36–47


LI Cheng-Dong  Associate professor at the School of Information and Electrical Engineering, Shandong Jianzhu University. He received his Ph. D. degree from the Institute of Automation, Chinese Academy of Sciences in 2010. His research interest covers intelligent systems and intelligent control with applications to intelligent buildings. Corresponding author of this paper. E-mail: lichengdong@sdjzu.edu.cn

ZHANG Gui-Qing  Professor at the School of Information and Electrical Engineering, Shandong Jianzhu University. He received his Ph. D. degree from the Xi’an Jiaotong University in 2002. His research interest covers intelligent control methods, intelligent buildings, smart home and Internet of things. E-mail: qqzhang@sdjzu.edu.cn

WANG Hui-Dong  Lecturer at the School of Management Science and Engineering, Shandong University of Finance and Economics. He received his Ph.D. degree from the Institute of Automation, Chinese Academy of Science in 2010. His research interest covers intelligent systems and type-2 fuzzy logic. E-mail: huidong.wang@ia.ac.cn

REN Wei-Na  Master student at the School of Information and Electrical Engineering, Shandong Jianzhu University. Her research interest covers intelligent systems and intelligent buildings. E-mail: weina.ren@foxmail.com