Changes under the Hood - a New Type of Non-Singleton Fuzzy Logic System

Amir Pourabdollah, *Member, IEEE,* Christian Wagner, *Senior Member, IEEE,*
Jabran Aladi, *Student Member, IEEE,*

Abstract—A major asset of fuzzy logic systems is dealing with uncertainties arising in their various applications, thus it is important to make them achieve this task as effectively and comprehensively as possible. While singleton fuzzy logic systems provide some capacity to deal with such uncertainty aspects, non-singleton fuzzy logic systems (NSFLSs) have further enhanced this capacity, particularly in handling input uncertainties. This paper proposes a novel approach to NSFLSs, which further develops this potential by changing the method of handling input fuzzy sets within the inference engine. While the standard approach is getting the maximum of the intersection between input's and antecedent's fuzzy sets (in the "pre-filtering" stage), it is proposed to employ the centroid of the intersection as the basis of each rule's firing degree. The motivation is to capture the interaction of input and antecedent fuzzy sets with high fidelity, thus making NSFLSs more sensitive to the input's uncertainty information. The testbed is the common problem of Mackey-Glass time series prediction in the presence of input noise. Analyses of the results show that the new method outperforms the standard approach (by reducing the prediction error) and has potential for a more efficient uncertainty handling in NSFLS applications.

I. INTRODUCTION

MOST of the applications of fuzzy logic systems (FLSs) include some forms of uncertainty handling. Examples include dealing with sensor noise in engineering applications, measurement approximation in the natural science or prediction of chaotic time series [1]–[3]. Non-Singleton FLSs (NSFLSs) [4], are specifically designed for handling the uncertainties associated with the FLSs' inputs. The input uncertainty in NSFLSs is captured by non-singleton fuzzification, thus transforming the crisp inputs to fuzzy input sets [5], as opposed to singleton fuzzification in the case of singleton FLSs (SFLSs).

Mendel and Mouzouris comprehensively established the theoretical basis of NSFLSs in [5] and [4]. When Mamdani's inference method [6] is utilized, they have shown that input and antecedent fuzzy sets are intersected, and the maximum value of their intersection is employed as the degree of firing for the given antecedent. It is also shown that this process can be conceptualised as a *pre-filtering stage*, i.e. the NSFLS intersects each fuzzy input set with each antecedent set, then extracts a crisp, representative value for each intersection. The inference engine then processes the produced crisp values as

Amir Pourabdollah, Christian Wagner and Jabran Aladi are with the Intelligent Modelling and Analysis Group and the Horizon Digital Economy Research Institute, School of Computer Science, University of Nottingham, Nottingham, UK. (email: amir.pourabdollah@nottingham.ac.uk, christian.wagner@nottingham.ac.uk, itxjha@nottingham.ac.uk)

it would have processed them in a SFLS. As an example of NSFLS applications, they applied the theory in predicting Mackey-Glass time series.

The described pre-filtering is a critical point in NSFLS's functionality since it "concentrates" each input's captured uncertainty into a single crisp value. In this paper we propose a novel pre-filtering method, motivated by the desire to capture the interaction of input and antecedent FSs with high fidelity, thus translating the information in these models consistently and comprehensively to FLS outputs. In turn, this should enable the separation of numerical and linguistic uncertainty modelling, both of which are commonly "mixed" in the input and antecedent FSs. We feel this is valuable in particular, as it enables the capture and modelling of uncertainties *where they arise*, i.e. whether in the inputs (e.g., a sensor) or in the antecedents (e.g., unsure expert views about the linguistic labels - what is "low").

Similar to Mendel and Mouzouris's approach, we apply the proposed deviation to the well-known problem of Mackey-Glass time series prediction which provides verifiable ground truth, enables easy reproduction of our results by other researchers, and enables us to easily modulate the different characteristics such as the level of uncertainty/noise in the experiments.

There are some other research works on using NSFLSs for time-series prediction under uncertainty, showing that NSFLSs provide better prediction in comparison to the SFLSs. For example in [7] NSFLSs are utilized to forecast time series from the domain of financial markets, or in [8], the Box-Jenkin's gas furnace time series data is used in conjunction with NSFLS-based prediction.

This paper presents essential details (with a focus on the practical application) of the proposed type of NSFLSs. A more detailed version, including a discussion of the underlying changes to the *sup-star composition of fuzzy relations* has been submitted as a journal publication [9].

Following this introduction, the standard NSFLS architecture will be reviewed in Section 2. We proceed in Section 3 by the proposed alternative type of NSFLSs. We then experimentally compare the new and standard methods in predicting Mackey-Glass time-series in Section 4. Finally in Section 5, we conclude and provide the direction of future work.
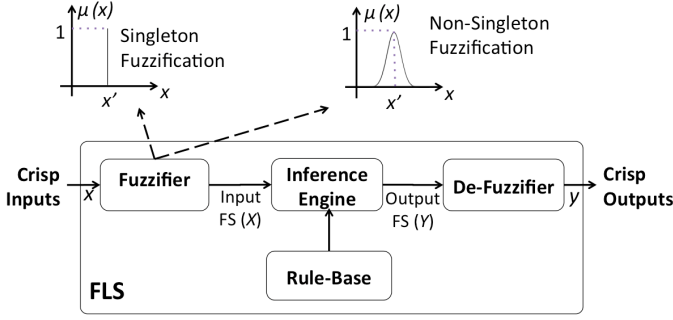
Fig. 1. FLS components and the illustration of different fuzzification methods. Singleton and non-singleton fuzzification of a sample input x' are also shown.

II. BACKGROUND

In this section we briefly introduce background material related to the functionality of NSFLSs in handling uncertain inputs.

A. Overview

In order to address uncertainty in the actual inputs to a FLS, the fuzzifier block in NSFLS, maps a given crisp input to a fuzzy input set, rather than to a fuzzy singleton as is the case in SFLSs. Thus, as shown in Fig. 1, the overall components of a SFLS and a NSFLS are identical, the only difference is the handling of the crisp inputs in the fuzzifier.

In NSFLSs, the actual type of membership function to be employed for the input FSs is application dependent, with the most common being a type of fuzzy number, i.e. a convex, normal FS. In Fig. 1, a Gaussian distribution is shown as an example. While NSFLSs have been defined for type-1 and type-2 FLSs, in this paper we limit ourselves to type-1 non-singleton type-1 FLS (i.e. type-1 FLSs that use type-1 input MFs), and will leave type-2 systems to future work, even though the proposed modifications still apply.

The general mapping between NSFLSs' inputs and outputs, i.e. between input set X and output set Y in Fig. 1, is comprehensively detailed in [5]. We do not repeat the details of how the comprehensive formula for the mapping is derived, instead we consider a simplified NSFLS, and focus on the details of "pre-filtering".

B. Pre-filtering in Standard NSFLSs

To illustrate the role of pre-filtering in NSFLSs, we consider a single-input, single-rule and single-output discrete FLS where Mamdani implication is used. Let x and y be members of input and output FSs (X and Y in Fig. 1) and let A and C be two FSs representing an antecedent and a consequent. The only defined rule is *If x is A then y is C*. We also define $\mu_X(x)$, $\mu_A(x)$, $\mu_C(y)$ and $\mu_Y(y)$ as the membership functions of X, A, C and Y respectively. In this case, the input-output mapping derived in [5] is reduced to:

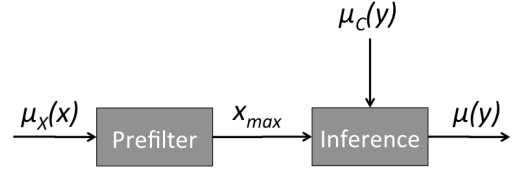$$\mu_Y(y) = \mu_C(y) \star max_{x \in X}[\mu_X(x) \star \mu_A(x)] \quad (1)$$
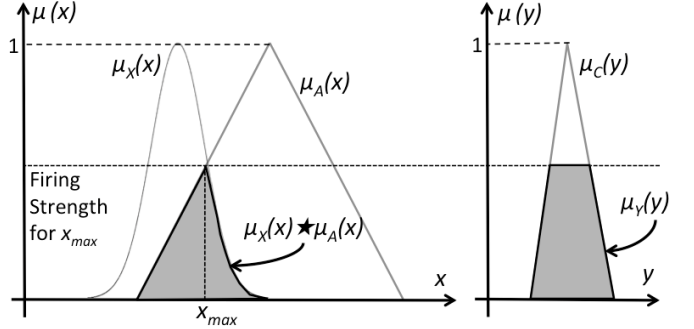

Fig. 2. Pre-filtering of input FS to a NSFLS

Fig. 3. The illustration of how a simple standard NSFLS calculates its output (Y) according to its input (X), antecedent (A) and consequent (C) FSs.

or equally;

$$\mu_Y(y) = \mu_C(y) \star \mu_X(x_{max}) \star \mu_A(x_{max}) \quad (2)$$

where $\star$ is any t-norm operator and x_{max} is the value of x at which $\mu_X(x) \star \mu_A(x)$ takes its maximum. The inference engine in a NSFLS can be imagined as a pre-filter unit added to a SFLS inference engine. The pre-filter unit transforms the uncertain input set to a representative numerical value x_{max} (Fig. 2) [5]. Handling the input uncertainty in NSFLSs is concentrated in this pre-filtering unit, i.e. the rest of the FLS acts identically to a singleton FLS.

If the minimum-operator is used as the t-norm, $\mu_X(x) \star \mu_A(x)$ is the intersection of X and A. Based on the definition of x_{max}, $\mu_X(x_{max})$ and $\mu_A(x_{max})$ are equal. As such, (1) can be written as:

$$\mu_Y(y) = min[\mu_A(x_{max}), \mu_C(y)] \quad (3)$$

Equation (3) expresses the input-output mapping in our simplified system. Briefly, this formula tells us that *the firing level of an antecedent is the maximum of its intersection with the input set* (Fig. 3). We refer to standard NSFLSs based on employing the described pre-filtering method as *standard NSFLS* hereafter.

Equation (3) is a simple version of input-output mapping used in the defined simple NSFLS. For the general NSFLS case, the same formulation is iterated for an arbitrary number of inputs and rules (detailed in [5]).

In this section, we have briefly reviewed how the uncertain inputs are handled in the standard NSFLSs. However, while very useful, the pre-filtering method in the standard NSFLSs is not the only possible method. The next section introduces the possible benefits of using an alternative method.

III. AN ALTERNATIVE TYPE OF NSFLS

In this section we first discuss the motivation for proposing an alternative type of NSFLS, then we proceed by formally introducing the alternative.

A. Motivation

The standard approach to NSFLSs has proved highly useful and effective in comparison to SFLS (e.g., [8], [10], [11]). However, while standard NSFLSs provide the capability for improved uncertainty capture in inputs (over that of SFLSs), the nature of the employed pre-filtering mechanism does not provide a fine-grained tracking of uncertainty information - as it is largely insensitive to the shape of the input FSs. In other words, the "pre-filter" in standard NSFLSs results in a significant loss of information regarding the intersection of input and antecedent models.

To illustrate, in Fig. 4, two different input FSs in a typical NSFLS are intersected with a single antecedent. Even though the actual input FSs are different, the firing levels calculated by the standard method are the same in both cases. Intuitively, we would expect a different output of the FLS when the uncertainty model of our inputs are different. Specifically in the example of Fig. 4, one would expect that the antecedent A is fired differently (at a lower level in this example) for input X_2 than for X_1, since $\mu_A(x)$ decreases for those higher values of x within X_2 that are not within X_1.

It is arguable that x_{max} may not be the best choice in terms of capturing the interaction (i.e. intersection) between the input and antecedent FSs with high fidelity. The details of the intersection are partially omitted in standard NSFLSs. An approach with a more detailed capture of input uncertainty and its interaction with the prospective antecedent FS is desirable. Specially, an alternative should have a higher sensitivity to the "shape" of the intersection. This is particularly relevant in the case when input fuzzy sets are complex, e.g., are designed based on for example prior modelling [12].

Based on the motivation above, we proceed to introduce an alternative type of NSFLS in the next sub-section.

B. The Alternative to Standard NSFLS

We propose an alternative type of NSFLSs with improved sensitivity for capturing the detailed interaction of input and antecedent FSs. Since we are interested in taking as much of the information about the *shape* of interacting FSs into account as possible, our choice is to compute the *centroid* of the resulting intersection of input and antecedents FSs as a pre-filter, rather than relying on the maximum. Clearly, the centroid is still a drastic simplification of the intersection, and richer models (e.g., an interval of FS) could be used. However, the latter would significantly increase the overall complexity of the NSFLS, while the centroid-based approach preserves the same complexity as standard NSFLS, after the firing strength has been determined.

In general, for a discrete fuzzy set X with a membership function $\mu_X(x_i)$, the centroid is defined as:

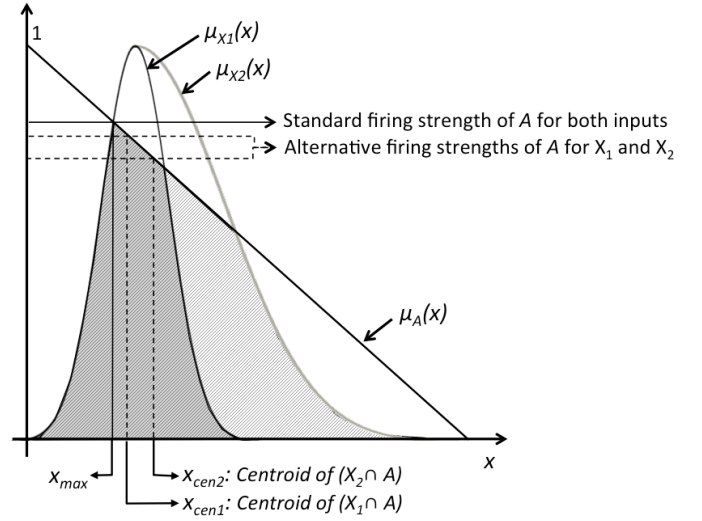$$x_{cen}(X) = \frac{\sum_{i=1}^{n} x_i \mu_X(x_i)}{\sum_{i=1}^{n} \mu_X(x_i)}, \qquad (4)$$

Fig. 4. The intersection of two different input fuzzy sets X_1 and X_2 with a single antecedent fuzzy set A, where x_{max} (and consequently the firing strength) is equal for both cases. Alternative firing strengths of A are shown when the centroid of each intersection is used instead of their maximum.

where n is the number of discretization levels used in a discrete system. In our case, we are interested in the centroid of the intersection of an input X and an antecedent A, i.e. the centroid of $X \cap A$. In this case we define the new input/output mapping as:

$$\mu_Y(y) = \mu_{X \cap A}\big(x_{cen}(X \cap A)\big) \star \mu_C(y), \qquad (5)$$

or for minimum t-norm:

$$\mu_Y(y) = min\big[\mu_{X \cap A}\big(x_{cen}(X \cap A)\big), \mu_C(y)\big] \qquad (6)$$

Briefly, this formula tells us that *the firing level of an antecedent is the membership degree of its intersection with the input set at the intersection's centroid point* (Fig. 4).

We refer to this type of NSFLS as *Cen-NSFLS* hereafter. Note that while in (3), $\mu_A(x_{max}) = \mu_C(x_{max})$, this is not necessarily the case in Cen-NSFLSs, thus resulting in the requirement for specifying the membership degree in the actual intersection, i.e. $\mu_{X \cap A}\big(x_{cen}(X \cap A)\big)$.

In the following section we focus on the exploration of the proposed approach through a series of experiments and analysis.

IV. EXPERIMENTS AND RESULTS

Following the proposition of the Cen-NSFLS in the previous section, its practical applicability is examined in this section. Specifically, we aim to shed light on the hypothesis that the novel method provides a superior integration and modelling of the uncertainty in system inputs and its interaction with antecedents, in turn enabling in superior performance when the input uncertainty is accurately captured within the input FSs. In this context, both the standard NSFLS and the new proposed Cen-NSFLS are employed within the prediction of Mackey-Glass chaotic time series under different Gaussian noise conditions (reflected in the different input FSs). In addition we compare both NSFLSs to a standard SFLS, aligning to [4] and [5].
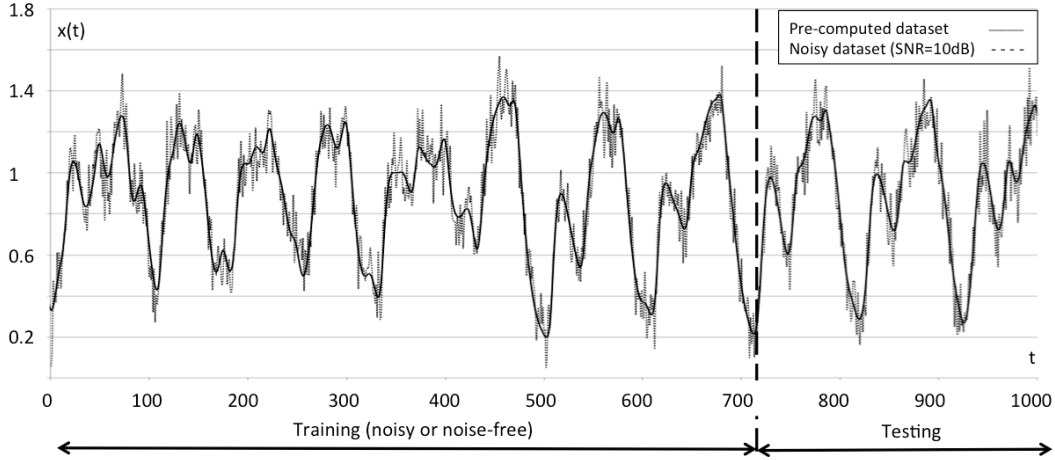
Fig. 5. The illustration of the pre-computed (noise-free) and noisy (SNR=10dB) time series. The training stage is from $t = 1$ to $t = 700$ (in both noise-free and noisy conditions) and the testing stage is from $t = 701$ to $t = 1000$. The period $t = -999$ to $t = 0$ (not shown here) is for the initial transients to die out.

A. Methodology

The method described in [4], uses standard NSFLSs for the prediction of Mackey-Glass chaotic time series in noisy conditions and compares the results to SFLSs. We follow a similar approach but conduct the same experiment on both the standard and the Cen-NSFLS. Each type of FLS is evaluated based on the MSE (Mean Squared Error) indicating the deviation of the actual FLS output from the expected value. We then compare the calculated MSEs for the individual experiments. Regarding the rule-base generation, we follow the method given in [13], an established approach to learning rules from existing input-output pairs.

The differential equation of the modeled Mackey-Glass time series [14] is defined as:

$$\frac{dx(t)}{dt} = \frac{0.2x(t - \tau)}{1 + x^{10}(t - \tau)} - 0.1x(t) \qquad (7)$$

This time series is a benchmark problem which has been considered by a number of researchers (e.g., in [15]–[19]). For $\tau > 17$, (7) demonstrates a *chaotic* behaviour. We have selected $\tau = 30$. Using (7), $x(t)$ is calculated for 2000 consecutive time points, i.e.: $t = -999$ to $t = 1000$. The first 1000 points are for the initial transients to die out, then using points $t = 1$ to $t = 700$ the system is trained to develop its rule-base. The last 300 points from $t = 701$ to $t = 1000$ are used for testing the system. Fig. 5 shows the pre-computed and an example of a noisy time series for SNR (Signal to Noise Ratio) of 10dB in training and testing intervals.

Nine past points in the time series are employed as inputs to generate a predicted value. Seven equally-distributed triangular membership functions are also used to model the input domains. Rules are trained according to the one-pass method described in [13]. Three FLSs are designed: a singleton and two non-singleton systems (standard and Cen-NSFLSs). For all the systems, Mamdani inference is used with *min* and *max* operators for the t-norm and t-conorm respectively. All the output FSs are deffuzified using centroid defuzzification. The same discretization level (=100) is used for all FLSs.

The input models used for NSFLSs are Gaussian membership functions centred on the crisp (noise-free) inputs, with a standard-deviation equal to that of the noise added in the given experiment. We note that the Gaussian fuzzifier employed for the NSFLSs is not necessarily the best approach, but we feel it provides a reasonable choice which should allow the NSFLS to "track" the increasing uncertainty in the inputs. For the SFLS, a singleton fuzzification is used.

The work in [4] used noisy data for both training and testing. While this is the case in most real world scenarios, it is also practically possible that a system is trained in noise-free (lab-) conditions and then used in a noisy real-world setting. To take this into account, we repeat our experiments for the two cases, i.e., where the system is trained in either noise-free or noisy conditions. In noise-free training, the same rule-base generated for the singleton FLS is used whereas in noisy training the rule-base is trained with noisy inputs for each noise level. For each NSFLS, two experiments are conducted for the different signal-to-noise ratios (SNRs) of 10dB and 5dB. Fig. 6 illustrates the noise-free and noisy input types for singleton and non-singleton FLSs. The described combinations of the system configuration lead to 8 individual systems/experiments in addition to the SFLS experiment.

Each experiment generates 300 outputs which are compared to the pre-computed outputs calculated by (7). The MSE over the 300 points is used as a measure of the overall error of a given FLS. Note that the aim of the experiments is the comparison of the individual FLS types, i.e. the relative best performance. We are not seeking to build a "best" time series prediction FLS for which one could employ more FSs, a different strategy for rule creation, etc.

In order to mitigate the effect of randomness, each experiment is repeated 30 times and the generated MSEs are averaged. This means that there are 270 individual experiments in total. The experiment design is illustrated in Fig. 7.

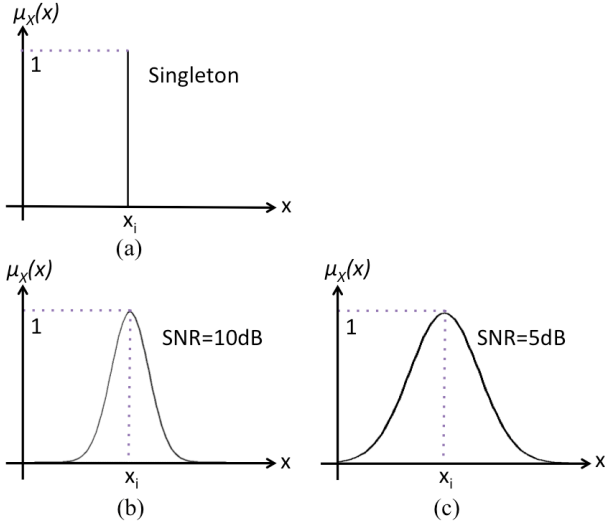The next sub-section provides the experiments' results.

Fig. 6. Three different uncertainty models that are used in the experiments. x_i represents the ith crisp input, that has been fuzzified differently. (a) singleton fuzzification; (b) non-singleton Gaussian fuzzification for SNR=10dB; (c) non-singleton Gaussian fuzzification for SNR=5dB.

Fig. 7. Illustration of the experiment design

B. Results

In order to illustrate the results, we provide both visual representations of individual outputs and tables capturing the MSE results averaged over the 30 repeated experiments.

First, we focus on the prediction of the SFLS. The SFLS is trained using noise-free data for $0 < t \leq 700$ in which 184 rules are generated. Then the SFLS is executed to compute its outputs over $700 < t \leq 1000$. The prediction results are shown in Fig. 8, which demonstrates that the system is relatively well trained and closely follows the pre-computed ground truth. This was anticipated based on the results shown in [4]. The averaged MSE over 30 FLS runs is 0.0014. Note that no noise is added at this stage, and the SFLS experiment is merely conducted as a benchmark.
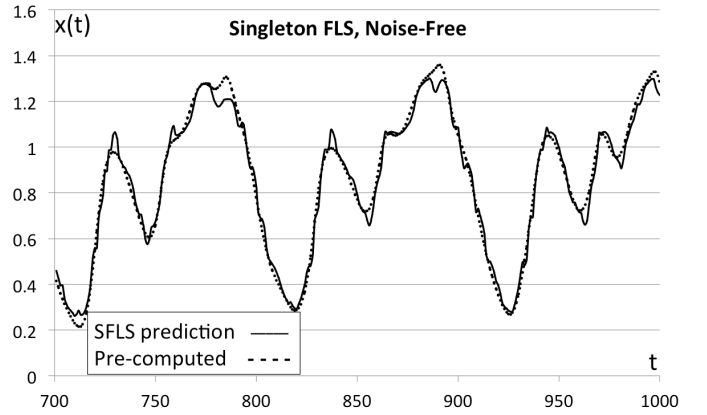

Fig. 8. SFLS output when trained with noise-free data compared to the pre-computed dataset.

Secondly, we focus on the NSFLSs and show the time series prediction under different noise conditions. We do not compare the results of the NSFLS with those of the SFLS (under noise) in this paper, since the same comparison has already been done in [4] where the authors showed that NSFLSs are more capable in handling noisy inputs than SFLS in different noise conditions. Instead, we focus on the comparison of the prediction results, when the two methods (standard and Cen-NSFLS) are utilized.

As far as the rule-training process concerns, while the noise-free training produced 184 rules, the training process under noise produced 557 rules for SNR=10dB and 664 rules for SNR=5dB. In all the cases, noisy data is used for testing as is intuitive in real-world applications.

Fig. 10 shows the NSFLS predictions when trained with noise-free data and tested with noisy inputs with SNR=10dB and SNR=5dB. This figure and specifically Table I highlight that utilizing Cen-NSFLS showed a reduced deviation from the pre-computed time series and thus better performance in comparison to when the standard NSFLS is used. Specifically, with SNR=10dB, the averaged MSE dropped from 0.0067 to 0.0058 (reduced by 13.19%). Also in SNR=5dB, averaged MSE changed from 0.0155 to 0.0135 (reduced by 12.61%). The same experiments are conducted for a NSFLS trained with noisy data. The average MSE in this case changed from 0.0124 to 0.0114 (7.71% reduction) for SNR=10dB, and from 0.0406 to 0.0332 (17.31% reduction) for SNR=5dB.

Table I and Fig. 11 show the summarized results for both NSFLS types. Note that all results are based on the average of 30 repeated experiments (see section IV-A), while figures 8 and 10 show a "single run" for visualization purposes.

C. Discussion

The summary of calculated MSEs (Table I and Fig. 11) show that for Cen-NSFLS, the MSE has improved in all four experiments. The highest improvement (17.31%) was found for noisy training with higher levels of noise (SNR=5dB) and the lowest improvement (7.71%) was achieved in noisy training but under lower noise levels (SNR=10dB). It is also shown that when the system is trained with noise-free data,
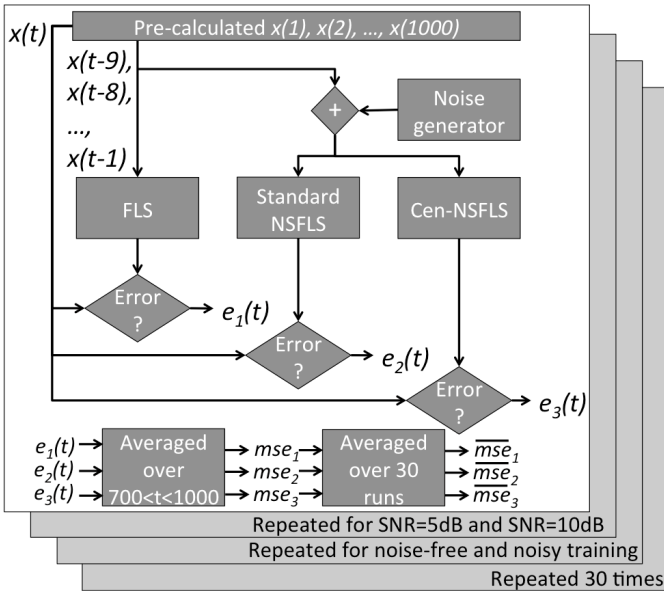
Non-singleton FLS: Noise-Free Training, Noisy Input (SNR=10dB)

(a)

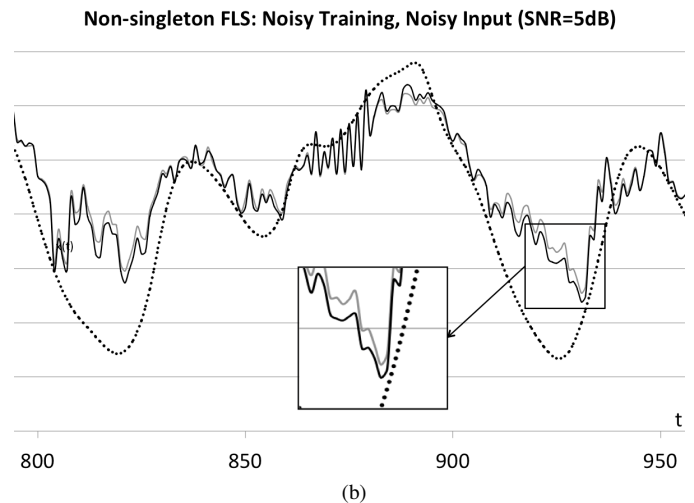**Non-singleton FLS: Noise-Free Training, Noisy Input (SNR=5dB)**
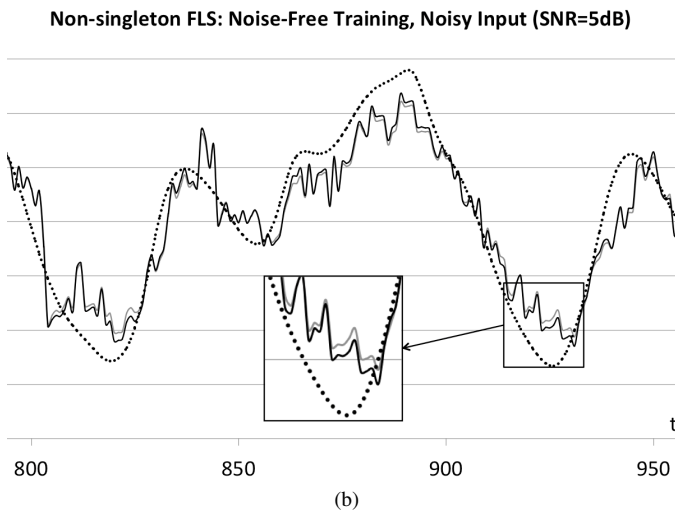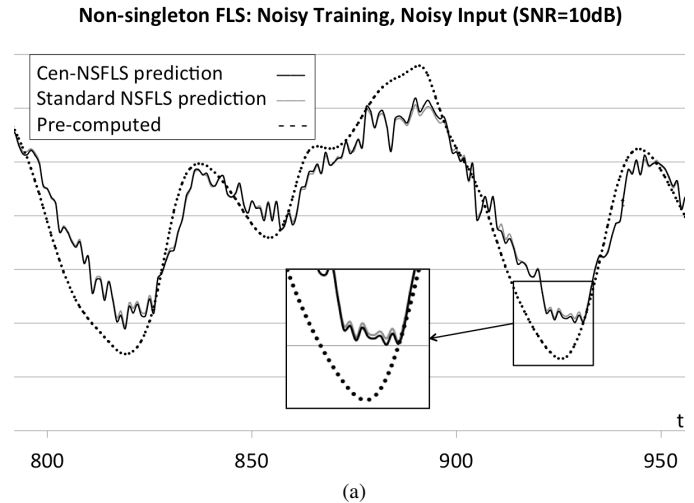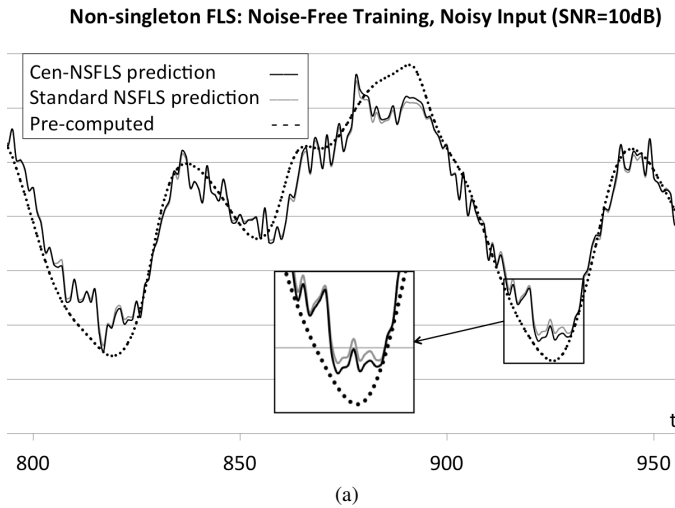

(b)

Fig. 9. NSFLS outputs, trained with noise-free data and tested with noisy data using Cen-NSFLS compared to the standard NSFLS. (a) SNR=10dB; (b) SNR=5dB. Note that only a subset of the samples are shown to enable better (larger) visualization.

Non-singleton FLS: Noisy Training, Noisy Input (SNR=10dB)

(a)

Non-singleton FLS: Noisy Training, Noisy Input (SNR=5dB)

(b)

Fig. 10. NSFLS outputs, trained and tested with noisy data using Cen-NSFLS compared to the standard NSFLS. (a) SNR=10dB; (b) SNR=5dB. Note that only a subset of the samples are shown to enable better (larger) visualization.

the MSE improvement is similar for different noise levels compared to when the system is trained with noisy data.

In order to examine the source of the observed MSE improvements in the described time-series predictions, it is helpful to investigate an individual output calculation and compare the results between both NSFLSs (standard and Cen-NSFLS). We focus on one of the designed NSFLSs where SNR=5dB and where noise-free training data are used. Here, we focus on one of the data samples, namely at t=1000. In this case, the outputs of the two systems (standard and Cen-NSFLS) are 1.150 and 1.180 respectively, whereas the expected value derived by the differential equation (7) is 1.280. Each of the two first values are results of a centroid defuzzification, predicted using 9 previous samples and a set of 184 fuzzy rules.

For this particular NSFLS at t=1000, the output FSs and the applied centroid defuzzification in the standard NSFLS and Cen-NSFLS are illustrated in Fig. 12. Intuitively, as shown in Fig. 12, we expect that the membership function of the Cen-NSFLS's output at each point is less than its counterpart for
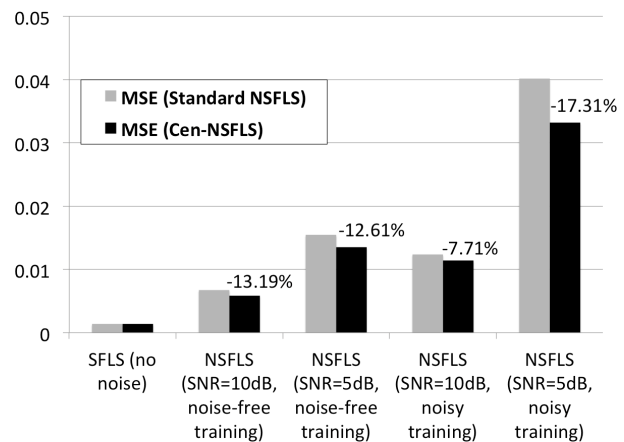

Fig. 11. Comparing the time series prediction errors (MSEs) when both methods are used in different experiments. MSE improvement is shown as a percentage.

TABLE I

THE RESULTS OF THE DIFFERENT FLSs BASED ON AVERAGE MSEs
(MEAN SQUARED ERRORS) AND THE CHANGE IN ERROR FROM THE
STANDARD NSFLS TO THE CEN-NSFLS.

System	MSE (Cen-NSFLS)	MSE (standard NSFLS)	Change in Error
Singleton FLS (no noise)	0.00138	0.00138	0.00%
NSFLS (SNR=10dB, noise-free training)	0.00673	0.00584	-13.19%
NSFLS (SNR=5dB, noise-free training)	0.01546	0.01351	-12.61%
NSFLS (SNR=10dB, noisy training)	0.01236	0.01141	-7.71%
NSFLS (SNR=5dB, noisy training)	0.04016	0.03320	-17.31%

TABLE II

THE NUMBER OF SAMPLES (OUT OF 300) WHERE EACH METHOD IS
OUTPERFORMED IN DIFFERENT NSFLS CONFIGURATIONS

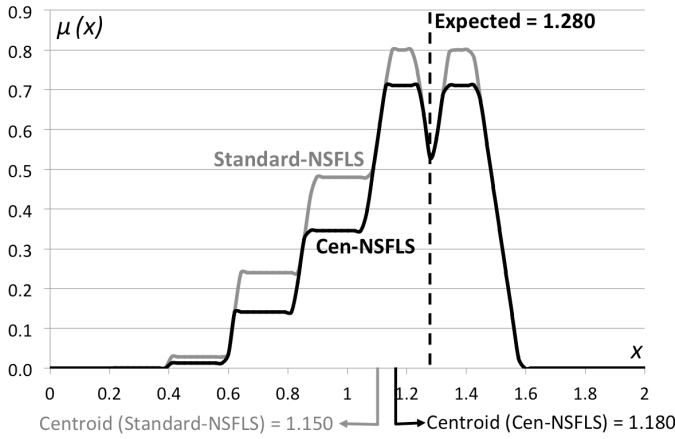| NSFLS Configuration	Outperformed Cen-NSFLS's prediction	Outperformed standard NSFLS's prediction
SNR=10dB noise-free training	216	84
SNR=5dB noise-free training	214	86
SNR=10dB noisy training	204	96
SNR=5dB noisy training	216	84

Fig. 12. Output FSs for t=1000 in a sample NSFLS with SNR=5dB and noise-free training. The centroid defuzzification results for both cases, together with the expected ground-truth value are shown.

the standard NSFLS, because for each intersection between an input and an antecedent FS, the firing strength of x_{cen} is equal or less than the firing strength of x_{max} (see also Fig. 4). Beyond this, the centroids of the output FSs are calculated based on the overall shape of the output membership function, not only on the individual membership grades. In the example shown in Fig. 12, the change in the shape of the membership function has shifted the graph's centroid to the right and has resulted in it being closer to the expected value.

This particular set of results shows a better estimation of the actual value by the Cen-NSFLS compared to the standard NSFLS. However, for other samples, this may not be always the case. On a sample-by-sample analysis of the particular NSFLS, the Cen-NSFLS provided a better estimate (lower MSE) than the standard NSFLS in 216 out of 300 samples (from t=701 to t=1000). This shows that in most of the cases (in this arbitrarily chosen set of samples), the change in the shape of the output membership function made the centroid point closer to the expected value. In the case of SNR=10dB and when noisy training data are used, the result of the similar sample-by-sample analysis is almost the same.

Table II shows the number of times that each of the two methods outperformed the other over the 300 samples for different experiment configurations. The figures in Table II

show a similar pattern to the MSE improvement percentages in Table I. For example, the smallest outperformance and the minimum MSE improvement are both in the case of the higher-SNR noisy-trained system.

The described increase in performance of the Cen-NSFLSs means that in the studied time-series prediction systems, the new method provides more accurate results especially when the data samples (and thus the inputs) are more noisy. This indicates that the original motivation - to better track uncertainty captured in the input fuzzy sets - was achieved by the Cen-NSFLS. Clearly, this conclusion is so far based on observing a particular set of NSFLSs in a specific application. It also relates to the particular shape of the fuzzy sets used to capture the noise levels. With this in mind, the generalizability of the results to a wide range of NSFLS applications (particularly in time-series prediction) is a direction for future work.

D. Computational Complexity

It is intuitive to expect higher computational complexity for calculating the centroid in comparison to finding the maximum of a fuzzy set. While this may arguably be a drawback for the new method, the discrete representation of FSs in most real-world applications means that any increase in computation time may not be as large as one could expect.

A timing test is conducted for the NSFLSs where the input is noisy (SNR=10dB) and training data are noise-free. On average over 300 samples (from $t=701$ to $t=1000$), the calculation of a single output took 37.19ms for the standard NSFLS and 36.84ms for Cen-NSFLS (using a regular PC). The test shows no significant difference in the necessary time for a control iteration between the two methods.

To explain this, we notice that both methods rely on traversing the discrete MFs, while performing basic computation as illustrated by the following sample Java code listings for both approaches. If $disc$ is the discretization level and $u(i)$ is the discrete degree of membership of the intersection between the input and antecedent set, the following listing shows the computation of a FS's maximum:

```
peak=0; xmax=1;
for (i=1; i<=disc; i++) {
  if (u(i)>peak) {
    peak=u(i);
    xmax=x(i);
  }
} return xmax;
```

and the following listing computes the FS's centroid:

```
numerator=0; denominator=0;
for (i=1; i>=disc; i++) {
  numerator+=x(i)*u(i);
  denominator+=u(i);
} return numerator/denominator;
```

Both listings include a similar loop over $disc$ sampled points. While the first listing employs a Boolean check following two assignments, the second listing employs a multiplications and an incrementation. The similarity between the two listing's complexity explains the observed timing similarity between the two methods. Moreover, both listings are computationally so simple that on modern computer architectures, the timing difference is negligible. Finally, it is noticeable that both listings purposefully do not make any assumptions about the FS shapes (e.g. convexity).

V. Conclusions and Future Work

This paper aims to develop the capacity of NSFLSs to better exploit models of input uncertainty in order to provide better overall uncertainty capture and thus, better performance. We proposed a novel type of NSFLS in order to enable this higher fidelity capture of the input uncertainty embodied by input fuzzy sets in comparison to traditional NSFLSs. In the novel approach, the firing level of the rules are calculated based on the centroid of the intersection of input and antecedent FSs, instead of being based on the intersection's maximum. This operation visually better captures the detail of the input FS's membership function during inference and thus should intuitively provide the potential for more fine-grained modelling of the input uncertainty.

In order to support this intuition, we applied the new approach for NSFLSs to the common Mackey-Glass chaotic time series prediction problem and compared the results of both the novel and traditional NSFLS formulations for (a) different noise levels on SNR=10dB and SNR=5dB, and (b) different training methods with noise-free and noisy data. The results show that the mean-squared error (MSE) between the NSFLS predictions and the actual time series values is reduced for the new approach by about 7 to 17 percent - for all different noise/training conditions. These results where achieved using a very basic input FS construction method, where the standard deviation of the input FSs was adjusted in line with the applied noise levels.

While the scope of this conference paper focuses on the presentation of the novel approach to NSFLSs, their practical exploration and evaluation; a journal article [9] elaborating on the theoretical aspects of the work has been submitted.

Beyond the work presented here, there are a number of interesting avenues for future work. They include (a) trying other use cases in time-series or other applications when input data is noisy; (b) studying the comparison for other types of membership functions including non-Gaussian, non-convex and dynamic/adaptive types of input FSs (as in [11], [20], [21]); and (c) expanding the new method to non-singleton type-2 systems and type-2 non-singleton type-2 systems (i.e. where the input FS is also of type-2).

References

[1] G. Bojadziev and M. Bojadziev, *Fuzzy logic for business, finance, and management*. World Scientific Publishing Co., Inc., 2007.

[2] L. Duckstein *et al.*, *Fuzzy rule-based modeling with applications to geophysical, biological, and engineering systems*. CRC press, 1995, vol. 8.

[3] J. Yen, R. Langari, and L. A. Zadeh, *Industrial applications of fuzzy logic and intelligent systems*. IEEE Press, 1995.

[4] G. C. Mouzouris and J. M. Mendel, "Nonsingleton fuzzy logic systems: theory and application," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 1, pp. 56–71, 1997.

[5] J. M. Mendel, "Uncertain rule-based fuzzy logic system: introduction and new directions," 2001.

[6] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International journal of man-machine studies*, vol. 7, no. 1, pp. 1–13, 1975.

[7] G. C. Mouzouris and J. M. Mendel, "Nonlinear time-series analysis with non-singleton fuzzy logic systems," in *Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering*. IEEE, 1995, pp. 47–56.

[8] D. Kim, S.-H. Huh, and G.-T. Park, "Modeling corrupted time series data via nonsingleton fuzzy logic system," in *Neural Information Processing*. Springer, 2004, pp. 1298–1303.

[9] A. Pourabdollah, C. Wagner, and J. Aladi, "Improved uncertainty capture for non-singleton fuzzy systems," *(submitted to) IEEE Transaction on Fuzzy Systems*, 2014.

[10] A. B. Cara, I. Rojas, H. Pomares, C. Wagner, and H. Hagras, "On comparing non-singleton type-1 and singleton type-2 fuzzy controllers for a nonlinear servo system," in *IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*. IEEE, 2011, pp. 126–133.

[11] G. C. Mouzouris and J. M. Mendel, "Dynamic non-singleton fuzzy logic systems for nonlinear modeling," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 199–208, 1997.

[12] C. Wagner and H. Hagras, "Novel methods for the design of general type-2 fuzzy sets based on device characteristics and linguistic labels surveys," in *Proceedings of the Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference*. European Society for Fuzzy Logic and Technology, 2009, pp. 537–543.

[13] L. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, 1992.

[14] M. C. Mackey, L. Glass *et al.*, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[15] A. Lapedes and R. Farber, "Nonlinear Signal Processing Using Neural Networks," Los Alamos National Laboratory, Los Alamos, NM, Tech. Rep. LA-UR-87-2662, 1987.

[16] R. S. Crowder, "Predicting the mackey-glass time series with cascade-correlation learning," in *Proceedings of Connectionist Models Summer School*. Carnegie Mellon Univ, 1990, pp. 117–123.

[17] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, 1993.

[18] R. D. Jones, Y. Lee, C. Barnes, G. Flake, K. Lee, P. Lewis, and S. Qian, "Function approximation and time series prediction with neural networks," in *IJCNN International Joint Conference on Neural Networks*. IEEE, 1990, pp. 649–665.

[19] J. H. Aladi, C. Wagner, and J. M. Garibaldi, "Type-1 or interval type-2 fuzzy logic systemson the relationship of the amount of uncertainty and fou size," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2360–2367.

[20] N. Sahab and H. Hagras, "Adaptive non-singleton type-2 fuzzy logic systems: A way forward for handling numerical uncertainties in real world applications." *International Journal of Computers, Communications & Control*, vol. 6, no. 3, 2011.

[21] N. Sahab and Hagras, "Towards comparing adaptive type-2 input based non-singleton type-2 fls and non-singleton flss employing gaussian inputs," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2012, pp. 1–8.