International Journal of Approximate Reasoning 66 (2015) 39–52

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Approximation of centroid end-points and switch points for replacing type reduction algorithms

Syed Moshfeq Salaken [a,*], Abbas Khosravi [a], Saeid Nahavandi [a], Dongrui Wu [b]

[a] *Center for Intelligent Systems Research, Deakin University, Australia*
[b] *Machine Learning Lab, GE Global Research, United States*

ARTICLE INFO

Article history:
Received 27 February 2015
Received in revised form 28 July 2015
Accepted 28 July 2015
Available online 31 July 2015

Keywords:
Type-2 fuzzy logic system
Type reduction

ABSTRACT

Despite several years of research, type reduction (TR) operation in interval type-2 fuzzy logic system (IT2FLS) cannot perform as fast as a type-1 defuzzifier. In particular, widely used Karnik–Mendel (KM) TR algorithm is computationally much more demanding than alternative TR approaches. In this work, a data driven framework is proposed to quickly, yet accurately, estimate the output of the KM TR algorithm using simple regression models. Comprehensive simulation performed in this study shows that the centroid end-points of KM algorithm can be approximated with a mean absolute percentage error as low as 0.4%. Also, switch point prediction accuracy can be as high as 100%. In conjunction with the fact that simple regression model can be trained with data generated using exhaustive defuzzification method, this work shows the potential of proposed method to provide highly accurate, yet extremely fast, TR approximation method. Speed of the proposed method should theoretically outperform all available TR methods while keeping the uncertainty information intact in the process.

1. Introduction

Soon after the introduction of type-2 fuzzy set by Zadeh in 1974 [1], its applicability in different real world application become apparent. Presently, type-2 fuzzy logic system is used with significant success in the field of decision making [2–6], mobile robotics [7,8], control [9,10], prior processing of data [11], forecasting accuracy [12], noise reduction [13], survey processing [14,15], prediction interval construction [16,17], clustering [18], intelligent environment realization [19] and time series forecasting [20,14,21–26], to name a few. This vast field of application generally utilizes high research concentration on computationally simpler version of general type-2 fuzzy sets, interval type-2 fuzzy set (IT2FS) and corresponding interval type-2 fuzzy logic system (IT2FLS) [27–29]. One integral part of IT2FLS is the type reduction (TR) block (see Fig. 1), which generally poses a bottleneck in computation process [30,29,31]. Existing TR algorithms for non-simplified fuzzy sets, which intend to tackle the computational bottleneck using soft algorithmic approach, can be divided into two main classes depending on their approach to preserve the associated uncertainty information during TR operation. Wu termed them as “Enhancements to KM TR algorithm” [30] and “Alternative TR algorithms” [30]. Recently, Greenfield argued that most widely used TR algorithm, Karnik–Mendel algorithm [32], is not the most accurate TR algorithm available [33]. This poses an interesting question. If the exhaustive defuzzification is the most accurate method of TR operation, how should this be

* Corresponding author at: Deakin University, 75 Pigdons Road, Waurn Ponds, Victoria 3216, Australia. Tel.: +61 424262359.
E-mail addresses: syed.salaken@gmail.com, syed.salaken@research.deakin.edu.au (S.M. Salaken), abbas.khosravi@deakin.edu.au (A. Khosravi), saeid.nahavandi@deakin.edu.au (S. Nahavandi), drwu09@gmail.com (D. Wu).

http://dx.doi.org/10.1016/j.ijar.2015.07.010
0888-613X/

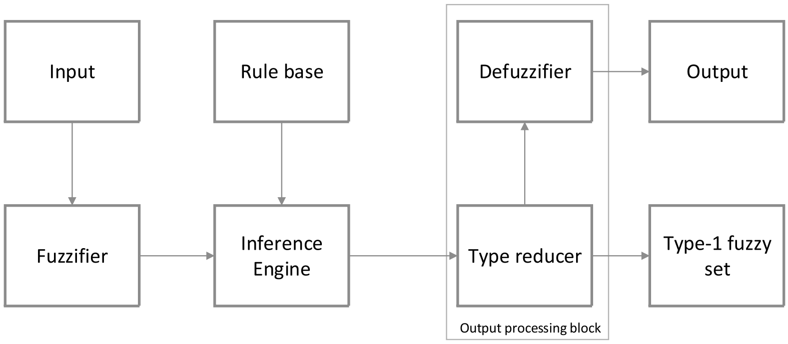


Fig. 1. IT2 fuzzy logic system.

approached while ensuring minimal run-time computational overhead with an acceptable error margin? In this research work, we propose a method to answer this problem. Formally stating, our motivation in this work is to find a systematic approach which will minimize the computational overhead associated with type reduction operation in IT2FLS and T2FLS while ensuring the accuracy of exhaustive defuzzification is preserved.

It is known for a long time that regression can be used for function approximation [34,35]. Here, we show that simple polynomial regression can be used for TR operation with very low error to address the issue mentioned in the above paragraph. Using regression based approach to approximate the underlying function mapping among membership functions (MFs), input and rulebase has a number of advantages including simpler and faster training, high robustness to capture the increased complexities with increasing number of inputs & MFs and the potential of utilizing long established tuning techniques for performance enhancement.

Since the KM type TR algorithms preserve the uncertainty by finding centroid end-points, which is closely intertwined with switch points, this work addresses both of them. Even though either centroid end-point approximation or switch point prediction is sufficient to determine the uncertainty information, we propose regression model for both of them to ensure completeness of the work.

It should be noted that regression model can be trained by either exhaustive defuzzification method, which guarantees most accurate output, or any existing TR algorithm. The essence of this study remains same regardless of the defuzzification method used in data generation stage. For simplicity, Karnik–Mendel TR algorithm is used in this work.

Rest of this paper is structured as follows: Section 2 states KM algorithm in detail and mentions all available TR algorithms for completeness, Section 3 gives an overview of experimental methods, Section 4 describes the data generation methods in detail, Section 5 describes regression modelling technique for both centroid end-points approximation and switch point prediction task, Section 6 discusses the findings and finally, Section 7 concludes the paper.

2. Preliminaries

2.1. Karnik–Mendel algorithm

Karnik and Mendel proposed an algorithm, referred as KM algorithm hereafter, for type reduction in 2001 [32]. Since then KM algorithm has been considered as a benchmark for TR. KM algorithm provides the left and right end-points of centroid which represent the uncertainties associated with a type-2 FLS. We describe the KM algorithm below for the completeness of this work. Mathematical proof of this algorithm can be found in [32].

For the left end of centroid, y_l:

1. Sort rule consequent x_i $(i = 1, 2, 3, \dots, N)$ in increasing order while keeping the variable name same, but now each element of x_i is positioned in ascending order such that $x_1 \le x_2 \le \dots \le x_N$.
2. Match the corresponding firing intervals or weights in the order that matches with related original value of rule consequent. In other words, renumber the index of lower and firing intervals, $\underline{w}_i$ and $\overline{w}_i$ respectively, to maintain the original one to one correspondence with rule consequent.
3. Initialize the weight, w_i as follows:

$$w_i = \frac{\underline{w}_i + \overline{w}_i}{2} \qquad \text{where, } i = 1, 2, 3, \dots, N \tag{1}$$

4. Compute the initial centroid value by

$$y = \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{2}$$

5. Find switch point $k \in [1, N-1]$ such that

$$x_k \leq y \leq x_{k+1} \tag{3}$$

6. Modify the weight as

$$w_i = \begin{cases} \overline{w}_i, & i \leq k \\ \underline{w}_i, & i > k \end{cases} \tag{4}$$

7. Compute the modified centroid as

$$y' = \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{5}$$

8. Check if $y' = y$. If yes, stop and set $y_l = y$ and call k as L. If no, go to step-9.
9. Set $y = y'$ and go to step-5.

For the right end of centroid, y_r:

1. Sort rule consequent x_i $(i = 1, 2, 3, \ldots, N)$ in increasing order while keeping the variable name same, but now each element of x_i is positioned in ascending order such that $x_1 \leq x_2 \leq \ldots \leq x_N$.
2. Match the corresponding firing intervals or weights in the order that matches with related original value of rule consequent. In other words, renumber the index of lower and firing intervals, $\underline{w}_i$ and $\overline{w}_i$ respectively, to maintain the original one to one correspondence with rule consequent.
3. Initialize the weight, w_i as follows:

$$w_i = \frac{\underline{w}_i + \overline{w}_i}{2} \qquad \text{where, } i = 1, 2, 3, \ldots, N \tag{6}$$

4. Compute the initial centroid value by

$$y = \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{7}$$

5. Find switch point $k \in [1, N-1]$ such that

$$x_k \leq y \leq x_{k+1} \tag{8}$$

6. Modify the weight as

$$w_i = \begin{cases} \underline{w}_i, & i \leq k \\ \overline{w}_i, & i > k \end{cases} \tag{9}$$

7. Compute the modified centroid as

$$y' = \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i} \tag{10}$$

8. Check if $y' = y$. If yes, stop and set $y_r = y$ and call k as R. If no, go to step-9.
9. Set $y = y'$ and go to step-5.

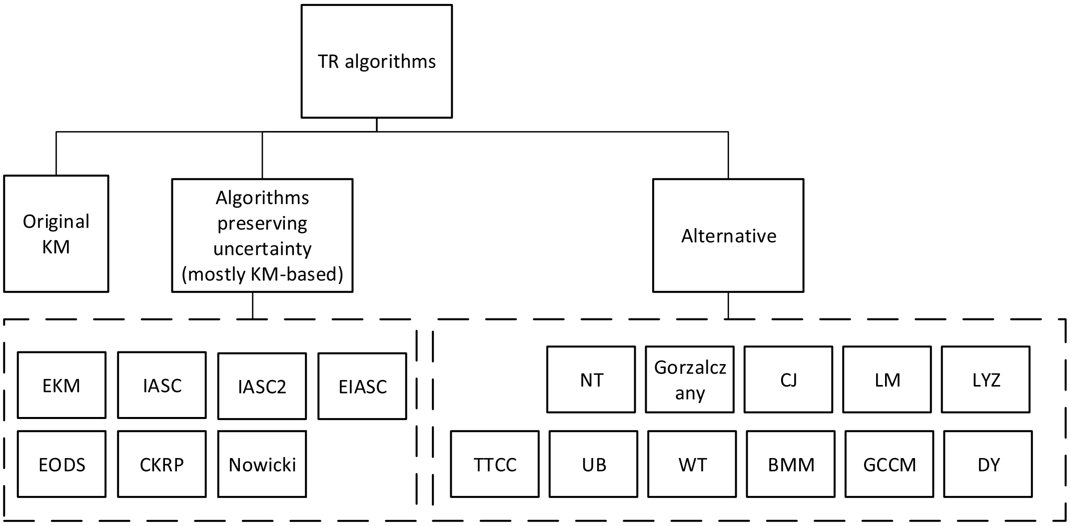
In summary, end-points of centroid, y_l and y_r, can be expressed as

$$y_l = \frac{\sum_{i=1}^{L} x_i \overline{w}_i + \sum_{i=L+1}^{N} x_i \underline{w}_i}{\sum_{i=1}^{L} \overline{w}_i + \sum_{i=L+1}^{N} \underline{w}_i} \tag{11}$$

$$y_r = \frac{\sum_{i=1}^{R} x_i \underline{w}_i + \sum_{i=R+1}^{N} x_i \overline{w}_i}{\sum_{i=1}^{R} \underline{w}_i + \sum_{i=R+1}^{N} \overline{w}_i} \tag{12}$$

where switch points, L and R are determined from KM algorithm. Defuzzified output is calculated as the average of y_l and y_r.

Please note that we did not consider different values for upper and lower end of rule consequent since they are same in most systems [36].



KM: Karnik-Mendel algorithm[32], EKM: Enhanced Karnik-Mendel algorithm[37], EKMANI: Enhanced Karnik-Mendel with New Initialization[38], IASC: Iterative Algorithm with Stop Condition[39, 40], EIASC: Enhanced Iterative Algorithm with Stop Condition[41], EODS: Enhanced Opposite Direction Searching Algorithm[42], IASC2: New Iterative Algorithm with Stop Condition[43], NT: Nie-Tan Algorithm[31], CJ: Coupland-John Geometric Method[44], LM: Liang-Mendel Unnormalized Method[45], UB: Wu-Mendel Uncertainty Bound method[46], BMM: Begian-Melek-Mendel method[47], GCCM: Greenfield-Chiclana Collapsing method[48], LYZ: Li-Yi-Zhao method[49], DY: Du-Ying method[50], TTCC: Tao-Taur-Chang-Chang method[51],CKRP:Chakraborty-Konar-Ralescu-Pal[52], Nowicki: Nowicki TR method [53, 54]

Fig. 2. IT2 fuzzy logic system.

2.2. Available TR algorithms

For completeness, a list of currently available TR algorithms is provided in Fig. 2.

3. Experimental set-up

Our objective in this research work is to compare and verify the approximation techniques for TR purpose. Primary focus of approximation technique is on largely used machine learning algorithms and methods. In this work, we investigate regression for approximation. The method of experiment is described in this section.

First, sample data is generated, then a regression model is chosen and trained. Afterwards, the regression model is tuned to enhance its performance and finally the model is tested. The data generation step consists of decision regarding the nature of FLS, number of MFs, number of input and distribution of rule consequent. The regression modelling step consists of choosing a hypothesis (model structure) and feature mapping. Training step consists of finding regression model parameters that minimizes cost function over training set. Tuning step finds the optimum regularization based on lowest cost from validation set. In this step, regression model is retrained using the optimum regularization. Finally, learned hypothesis is applied on test set to measure the performance. Please note, data in training set, validation set and test sets are completely different.

Both centroid end-points approximation and switch point prediction is covered in this work. These two are conducted on separate experiment. Therefore, data generation and regression modelling for these two tasks are independent and different. In Sections 4 and 5, both are described in detail. In this section, a pictorial view of overall experiment is presented, which is available in Fig. 3.

For the centroid end-points approximation task, a total of 100 FLSs are simulated and analysed. Performance is measured with mean absolute percentage error and root mean squared error. For the switch point prediction task, 12 FLSs are simulated with different number of inputs and MFs. Afterwards, impact of different hypothesis is analysed to measure the performance where the indicators are accuracy and error histogram.

4. Data generation

4.1. Approximation of centroid end-points

In order to show the effectiveness of KM output approximation, a fairly complex FLS is considered. If our approximation method can follow the actual output with small error, it would be logical to assume that proposed approximation tech-

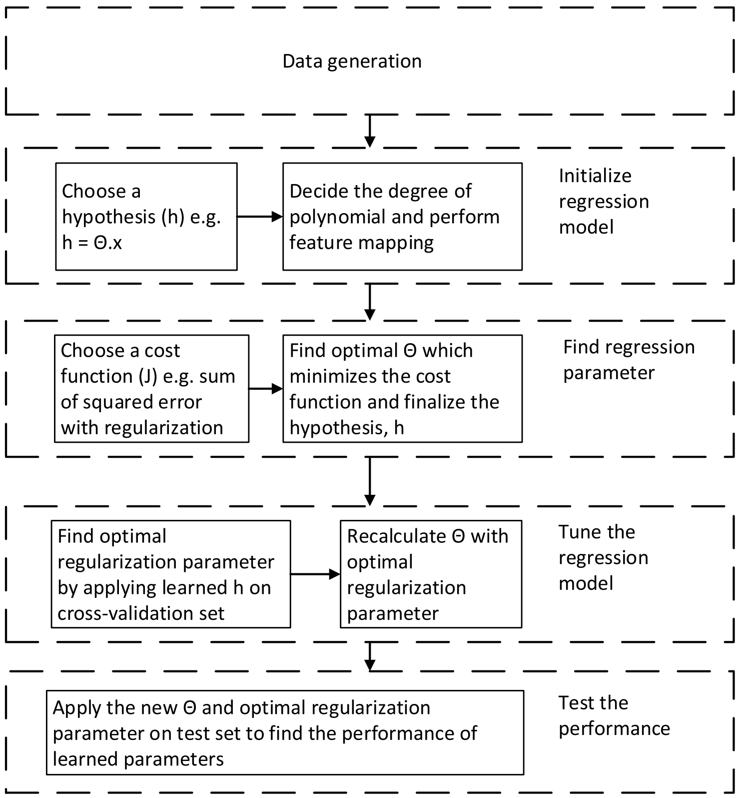


Fig. 3. Experimental procedure.

niques will be able to generalize over other FLS and is not limited to simple scenarios only. Therefore, an imaginary 4-input, 3 output fuzzy logic system with 3 membership functions for each input is simulated. Two random numbers between 0 and 1 are drawn from a uniform distribution and the smaller one is used as lower firing interval. This process is continued until the required length of firing interval is achieved. Rule consequents are generated randomly between 0 and +10 following the formula $0 + (10 - 0) * rand$, which also follows a uniform distribution. At this point, KM algorithm is called and centroid end-points are calculated. This is repeated 1000 times for a single FLS. Furthermore, the complete process is repeated 100 times which indicates a total of 100 different FLS has been simulated with same architecture. These data are stored and used throughout regression phases.

For a 4-input, 3 MF on each input FLS, length of vector containing each of the firing intervals and rule consequents are first calculated and then generated accordingly. For regression purpose, they are saved in the below format afterwards:

$$[\underline{f} \quad \overline{f} \quad y_c]$$

where $\underline{f}$, $\overline{f}$ and y_c indicate lower firing intervals, upper firing intervals and rule consequents, respectively. This is called input matrix. Centroid values are saved as per the below format:

$$[y_l \quad y_r \quad y]$$

where y_l, y_r indicates the left and right end-points of centroid and y indicates the defuzzified value found by KM algorithm. This is called the target matrix.

Please note, other methods of data generation i.e. keeping lower firing intervals in the range $[0, 0.5]$ and upper firing intervals in the range of $[0.51, 1.0]$ is tested as well. They produce similar observations and are not reported in this work.

4.2. Prediction of switch points

For this portion of study, a number of different FLSs with different number of inputs and MFs are simulated. This is done to demonstrate the relationship between polynomial degree in hypothesis and complexity in FLS architecture e.g. number of input and number of MFs on each input. For each FLS, 15 000 sample data are generated which indicates our data matrix will have 15 000 rows. Firing intervals are generated in the same fashion described in Section 4.1. For each FLS, rule consequents are generated between −10 to 10 following a uniform random number distribution.

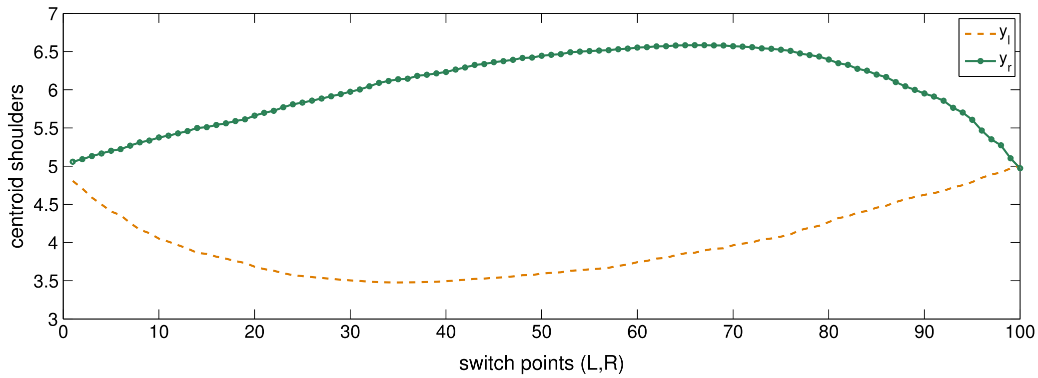


Fig. 4. Shape of y_l and y_r.

The input matrix for switch point prediction contains lower and upper firing interval, rule consequents, number of input in the system and number of membership function on each input. The i-th row of input and target matrix can be represented as follows:

$$input(i,:) = [\underline{f} \quad \overline{f} \quad y_c \quad n_{input} \quad n_{MF}];$$
$$target(i,:) = [L \quad R];$$

where,

n_{input} = Number of inputs in FLS;
n_{MF} = Number of MFs on each input;
L = Left switch point from KM algorithm;
R = Right switch point from KM algorithm;

In this study, 3 values for number of inputs and 4 values for number of membership functions on each input is considered. This can be expressed as follows:

$$n_{input} = \{2, 3, 4\};$$
$$n_{MF} = \{2, 3, 4, 6\};$$

To illustrate this experiment at a more detailed level, lets consider a FLS with 2 inputs and 4 MFs on each input. A FLS model is then built. A total of 15 000 sample data are generated for this FLS and are divided for training the regression model, tuning the model (validation) and testing the model in the ratio of 60%, 20% and 20%, respectively. The regression model is then trained with the chosen hypothesis. Afterwards, the optimum regularization parameter is chosen based on the lowest validation error. Using this optimum regularization parameter, the regression model is retrained and finally tested on the data uniquely reserved for testing purpose. The switch point is predicted as the nearest integer of regression output in this case as switch points can never be a fraction. This is done using the MATLAB command *round*. It is possible to define a threshold value instead of 0.5, coming from the MATLAB function *round*, to determine the flooring and ceiling operation. This threshold will then be considered as a design parameter. However, this option is not explored in this paper. Please note, there is no overlap among training, validation and testing data which is essential to ensure the result is not biased. Also note that feature mapping is performed on firing intervals and rule consequents before training is done. However, no such preprocessing is done on n_{input} and n_{MF}.

5. Regression modelling

5.1. Hypothesis

In the process of verifying the effectiveness of linear regression to approximate the output, as well as switch points, for KM algorithm, a polynomial hypothesis is chosen. A quick check (see Fig. 4) clearly indicates that any hypothesis with first degree polynomial is unlikely to sufficiently approximate the type reduction algorithm. After few trial and error, a third degree polynomial form is selected for further investigation to demonstrate the impact of polynomial degree on the success of regression modelling. However, impacts of polynomial degree 2 and 4 are also shown for switch point approximation/prediction.

Accordingly, our hypothesis can be represented in the following form:

$$h_x(\theta) = \sum_{i=0}^{3} \theta_i x^i = \theta_0 + \theta_1 x + \theta_2^2 x^2 + \theta_3^3 x^3; \tag{13}$$

Please note that using a polynomial hypothesis does not imply a non-linear regression.

5.2. Cost function

The aim of this regression model is to determine suitable value of θ parameter, which is usually a vector, that minimizes some form of error or cost function over the training data in conjunction with further tuning with validation data. In this work, a mean squared error form is chosen for the cost function with regularization.

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_x(\theta)^{(i)} - T^{(i)})^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \tag{14}$$

Here, λ is the regularization parameter and the second term in equation (14) is the regularization term. T is the target of the approximation function and can take the form of y_l, y_r, L or R depending on the problem at hand. For example, when approximating the right end-point of centroid, T will be replaced by y_r and in case of left switch point approximation, L will replace T. In addition, please note that regularization term does not include bias terms in the cost function.

5.3. Choosing optimal regularization

Further tuning of regression model depends on the test conducted on validation set as it increases generalization capacity of hypothesis and ensures better performance on test set. As part of further tuning, the value of regularization parameter λ is optimized based on the validation error. Finally, the value of λ that produces lowest validation error is reintegrated in into the hypothesis even if it is producing higher training error. It ensures that regression performance is not biased towards training set. The value of λ associated with lowest validation cost function value is called optimum λ and denoted as λ_{opt} through out this paper.

In this work, we have considered following values of λ:

$$\lambda = \{0, 0.5, 1, 1.5, 2, 3, 4, 5, 7, 9, 11\};$$

5.4. Feature mapping

As indicated in Sections 4.1 and 4.2, firing intervals and rule consequents are placed columnwise in a row in the input matrix. However, as a polynomial regression is done here, each column is considered as a feature and mapped accordingly. More specifically, firing intervals and rule consequents are mapped upto their third degree polynomial in the case of both end-points approximation and switch point prediction. However, number of input and MFs are not included in feature mapping operation. Therefore, i-th row of final input matrix takes the below form for centroid end-points approximation problem:

$$input(i,:) = [\underline{f} \quad \underline{f}^2 \quad \underline{f}^3 \quad \overline{f} \quad \overline{f}^2 \quad \overline{f}^3 \quad y_c \quad y_c^2 \quad y_c^3];$$

And for the switch point prediction problem, feature mapped input matrix takes the below form:

$$input(i,:) = [\underline{f} \quad \underline{f}^2 \quad \underline{f}^3 \quad \overline{f} \quad \overline{f}^2 \quad \overline{f}^3 \quad y_c \quad y_c^2 \quad y_c^3 \quad n_{input} \quad n_{MF}];$$

Due to this unconventional way of feature mapping in switch point prediction, relevant hypothesis becomes slightly complex and takes the below form:

$$h_x(\theta) = \sum_{i=0}^{3} \theta_i x^i + \beta_i z_i; \tag{15}$$

where x now contains firing intervals and rule consequents and z contains the number of inputs to the FLS and number of MFs on each input. Please note, x now has $3N$ number of columns where $N+2$ is the number of columns in x in equation (13). Also, z has only two columns. Bias is not considered when counting the number of columns.

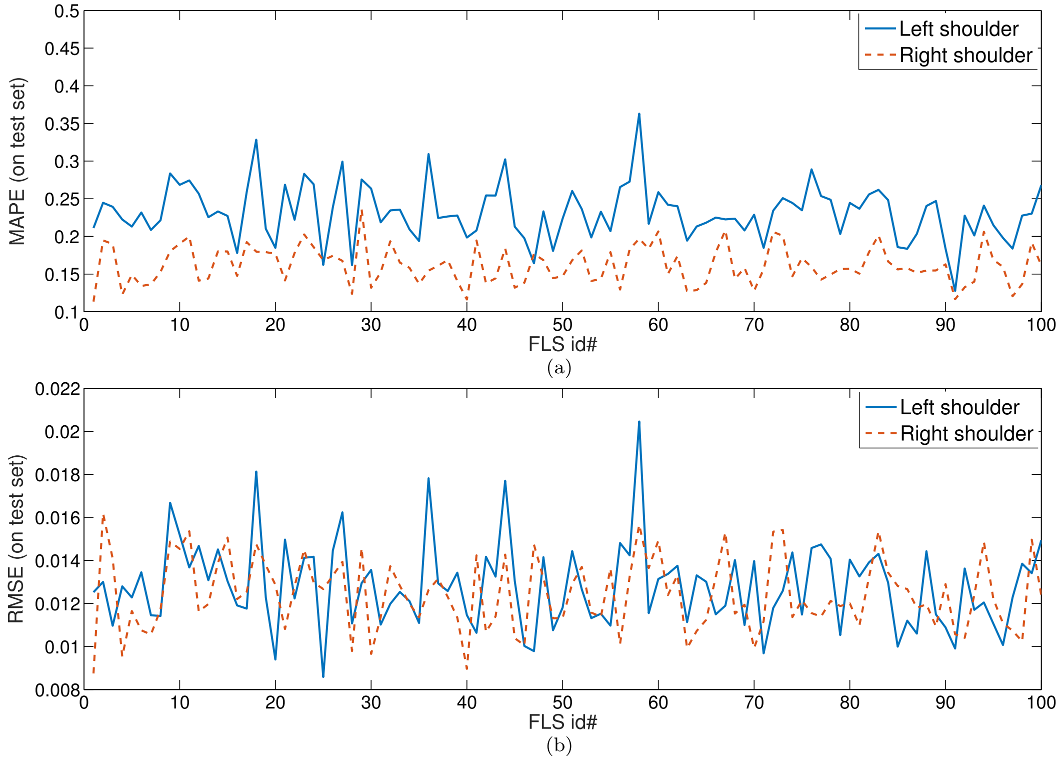


Fig. 5. Performance measurement of regression model (rules: [0, 10]).

6. Result

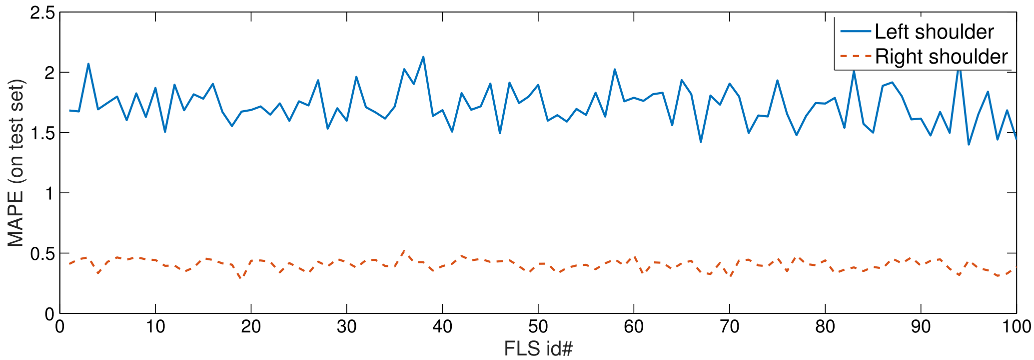
6.1. Centroid end-points approximation

Result for centroid end-points approximation is shown in Fig. 5 where Fig. 5a shows the mean absolute percentage error (MAPE) and Fig. 5b shows the root mean square error (RMSE) for 100 different FLS with positive rule consequents only. As described in Section 4.1, these results are the outcome of 1000 runs per FLS. Observe from 5a that MAPE values are very small and are less than 0.4% for both left and right end-points. Also note that, the absolute errors are similar. The MAPE for the left end-point is higher because the left end-point has smaller values. Also, here we are talking about centroid end-points instead of switch points. These small MAPE values strongly indicates that the error in regression approximation is very small and the process should be considered fairly accurate. This conclusion is reinforced by Fig. 5b where RMSE values lie in the range of $[0.007, 0.021]$, which indicates very small deviation from the output of KM algorithm.

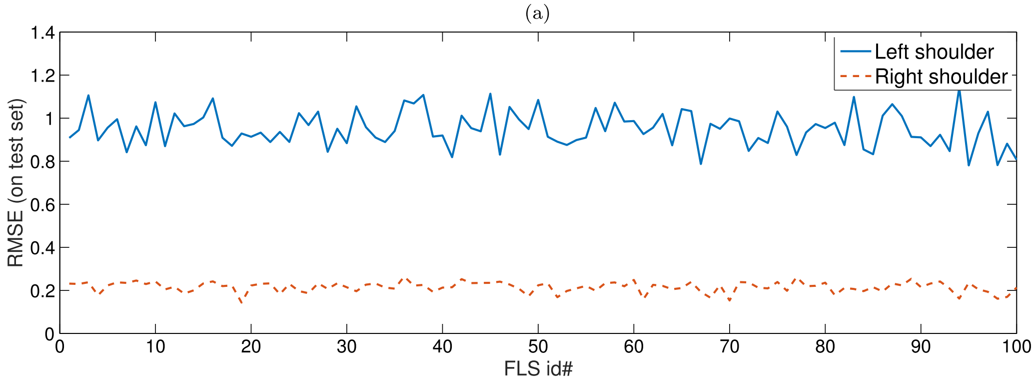
Fig. 6 shows the approximation error when rule consequents are negative i.e. in the range of $[-60, -30]$ in this case. Please note that, both MAPE and RMSE values have increased compared to the case where rule consequents are positive and does not include any negative values. However, the errors are still small and below 2.5% and 1.4% for MAPE and RMSE, respectively.

6.2. Switch point regression

The success of regression modelling for switch point prediction is depicted in Fig. 7–9 and Table 1. Recalling the fact that 20% sample is used for test set from a total of 15 000 samples, it is clear that prediction is highly accurate for FLS with low number of inputs and MFs. For example, top left image in Fig. 7a shows 100% accuracy as all prediction error lies in the zero bin, where 3000 is the total number of elements in test set (equivalent to 20% of 15 000). Also note from this figure, when complexity in FLS architecture increases with increased number of inputs and MFs, a 2nd degree polynomial can no longer predict the switch point with considerable accuracy. The worst case scenario can be seen from the bottom right image of Figs. 7a and 7b where number of inputs and number of MFs on each input is 4 and 6, respectively. However, when the degree of polynomial is increased, general performance for complex FLS is also improved. This can be realized by examining the bottom right images of Figs. 8a and 8b. Note that number of zero errors increases for both left and right switch prediction in comparison with 2nd degree polynomial for the same FLS. However, when a 4th degree polynomial is used for regression hypothesis for that same FLS, performance does not improve with respect to a 3rd degree polynomial



(a)



(b)

Fig. 6. Performance measurement of regression model (rules: [−60, −30]).

hypothesis. This indicates that regression is over-optimized for training data and other steps e.g. different features may be needed to improve this prediction.

Observe that prediction accuracy is very good for less complex system i.e. FLS which concurrently has small number of inputs (N_{input}) and small number of MFs (N_{MF}) on each input. For example, when $N_{input} = 2$ and $N_{MF} = 3$, prediction accuracy for both left and right switch point is 100% using a 2nd degree polynomial in hypothesis. With the increased complexity of FLS, prediction accuracy with this hypothesis starts to fall. This can be realized by looking at the case where $N_{input} = 3$ and $N_{MF} = 6$. Prediction accuracy for left and right switch points are 84.7% and 53.57%, respectively, in this case. However, when the polynomial degree increases in hypothesis, prediction accuracy may increase as well. This is evident from the fact that a 2nd degree polynomial yields 95.6% and 98.93% accuracy for left and right switch point prediction with a FLS configuration of $N_{input} = 2$ and $N_{MF} = 4$, whereas a 3rd degree polynomial provides 96.63% and 99.53% accuracy, respectively. It is important to note that different polynomial degree can be optimum for left and right switch point prediction for same FLS. For example, increase of polynomial degree (from 2 to 3) increases prediction accuracy for right switch point from 82.56% to 96.03%, but produces a significant reduction in left switch prediction for the same FLS (e.g. $N_{input} = 3$ and $N_{MF} = 4$). This highlights the fact that left and right switch point prediction are two separate problems. However, once modelled, they can be run in parallel.

The main point of these results is prediction accuracy and error histogram results are clearly showing that a regression model can successfully estimate the switch point for a complex FLS provided that proper tuning is done for that particular regression model.

At this point, we want to emphasize that both the KM approximation and switch point prediction are data-driven models. Accordingly, the regression model parameters will be different for every system. Every application, not every prototype of one specific application, needs to be designed separately and then parameters of regression models need to be tuned to achieve desired accuracy. Typical regression techniques can be used for tuning the model parameters. Also note that, this data-driven TR approach is very simple and moderately accurate. Once designed and tuned with the help of KM algorithm or exhaustive defuzzification, it will significantly reduce the run time computational cost because of the simplicity of learned hypothesis. TR approximation function will consist of few multiplication and addition to find the value of left and right end-points of centroid.

It should be further noted that the results in this research work only shows the accuracy for left and right end-points approximation. Since the end-point values carry the information required to calculate the defuzzified value and uncertainty estimation, we believe these results sufficiently describe the success of regression techniques in TR approximation.

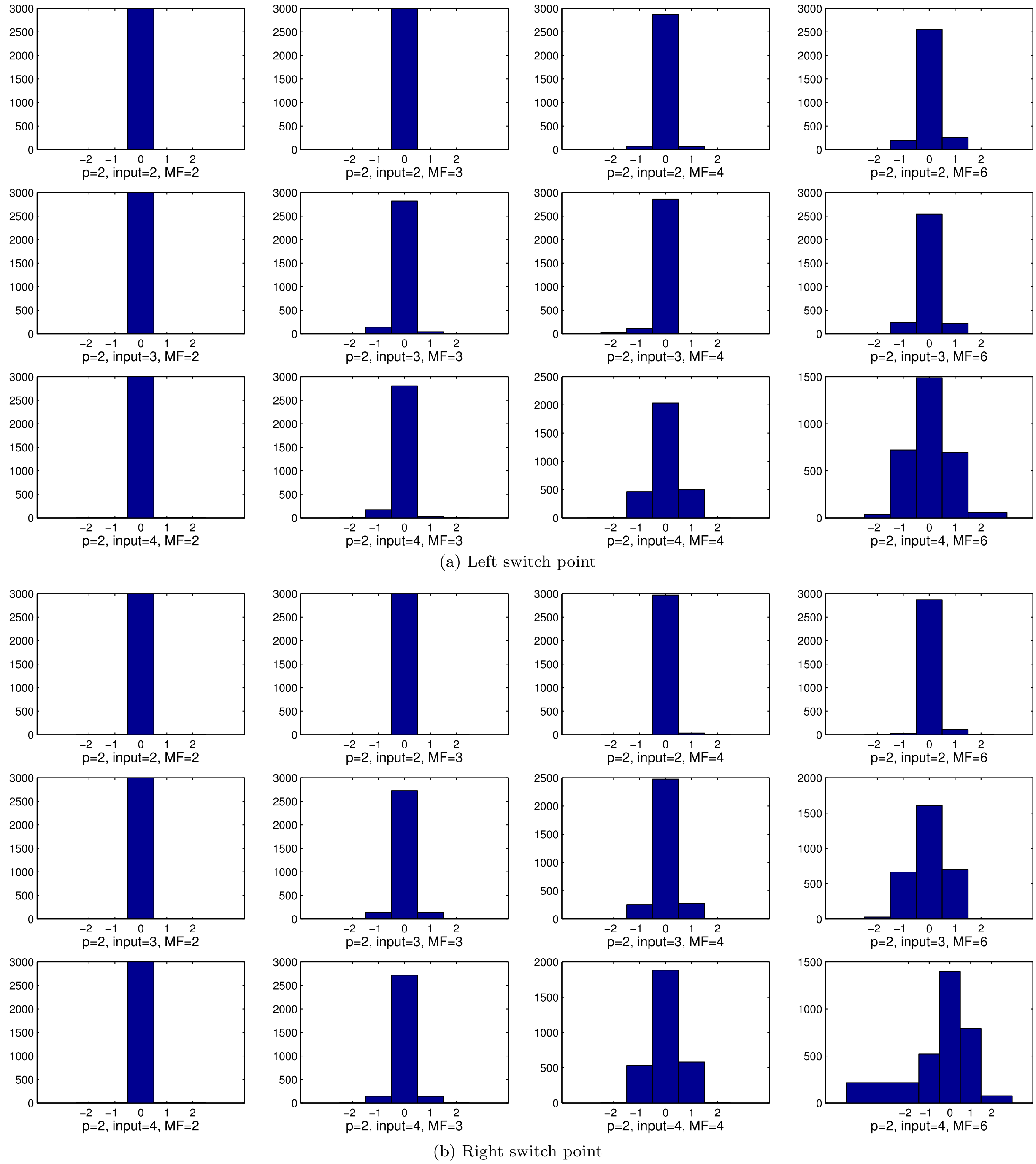
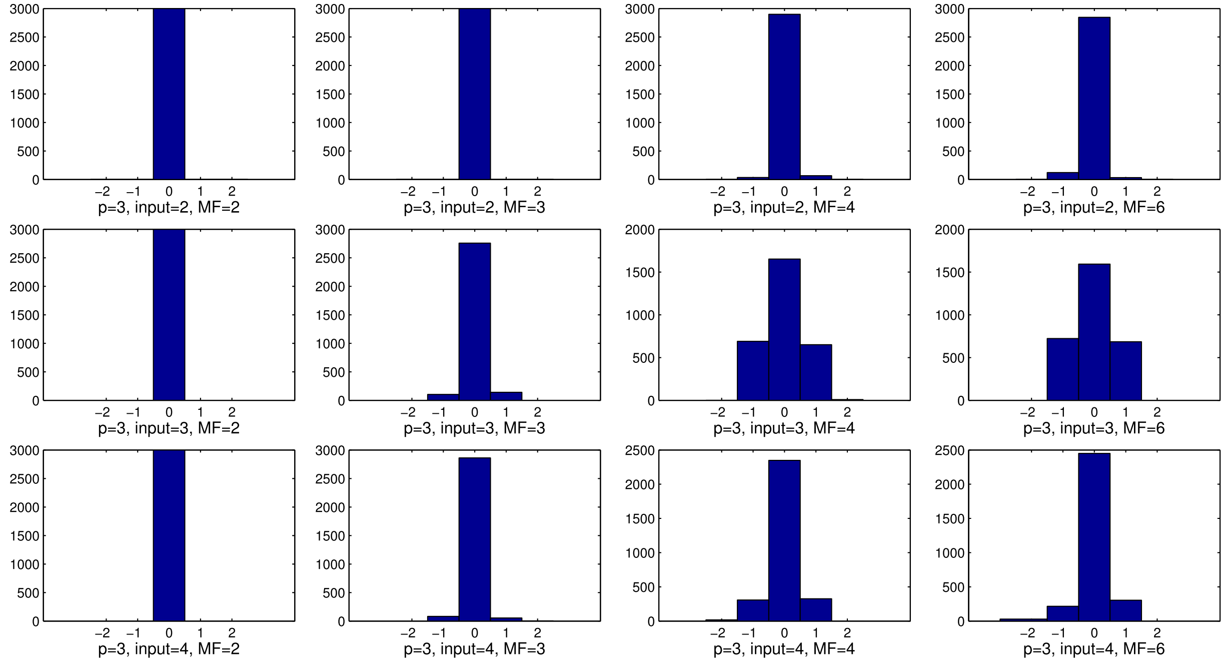


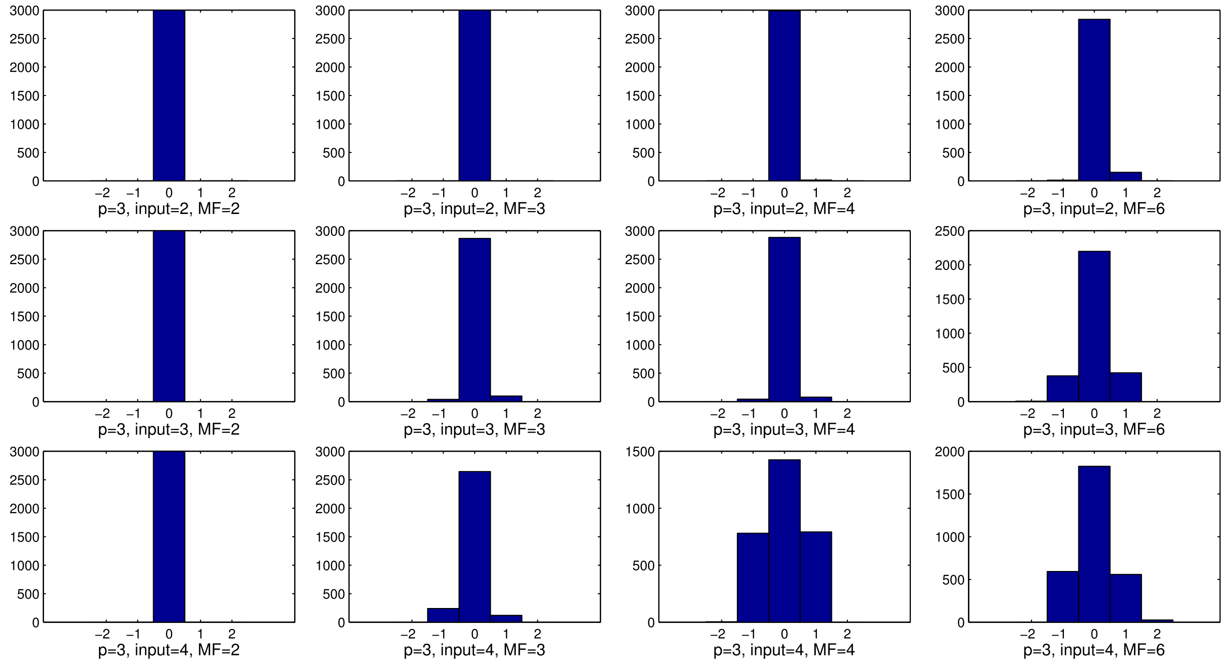
Fig. 7. Prediction accuracy for 2nd degree polynomial.

7. Conclusion

This research work demonstrates the applicability of regression modelling techniques for approximation of Karnik–Mendel type reduction algorithm. It addresses both end-point approximation and switch point finding tasks. Since every FLS is designed for a specific application and therefore has predicted input range, it is possible to use approximation models to replace type reduction block. In this paper, we show that regression technique can be used in design phase for such approximation purpose. This work, by its nature, is applicable to exhaustive defuzzification as well. Hence the question raised in the beginning of this paper about accuracy of TR operation can be answered by using proposed method in conjunction



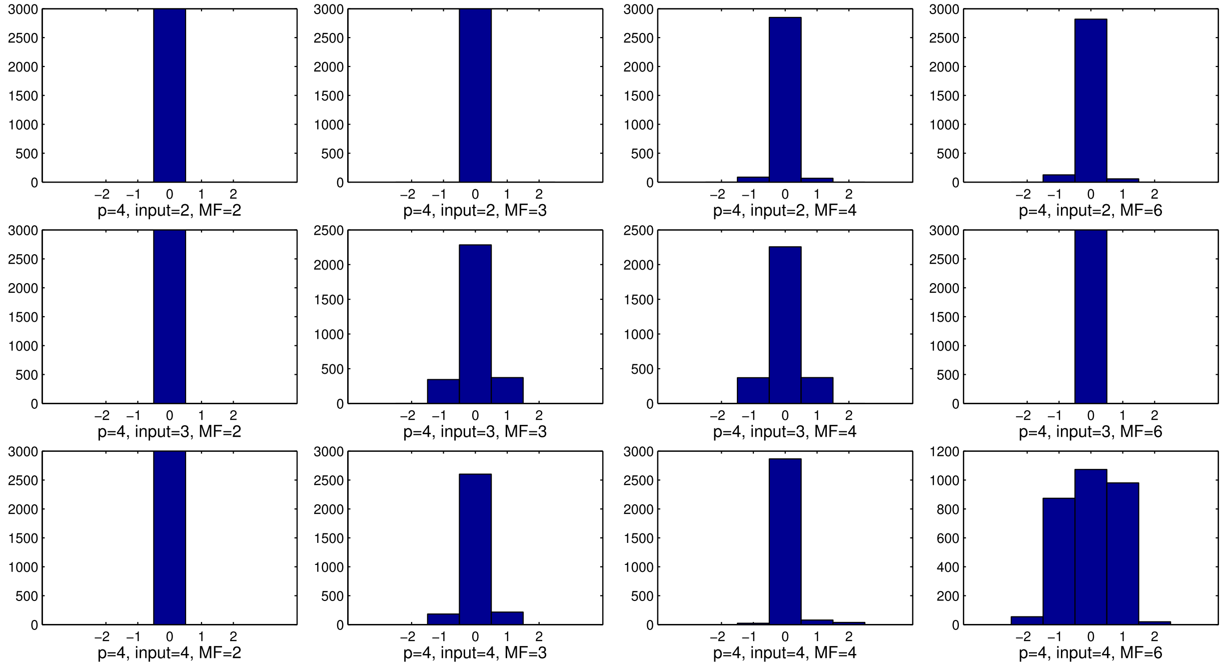
(a) Left switch point



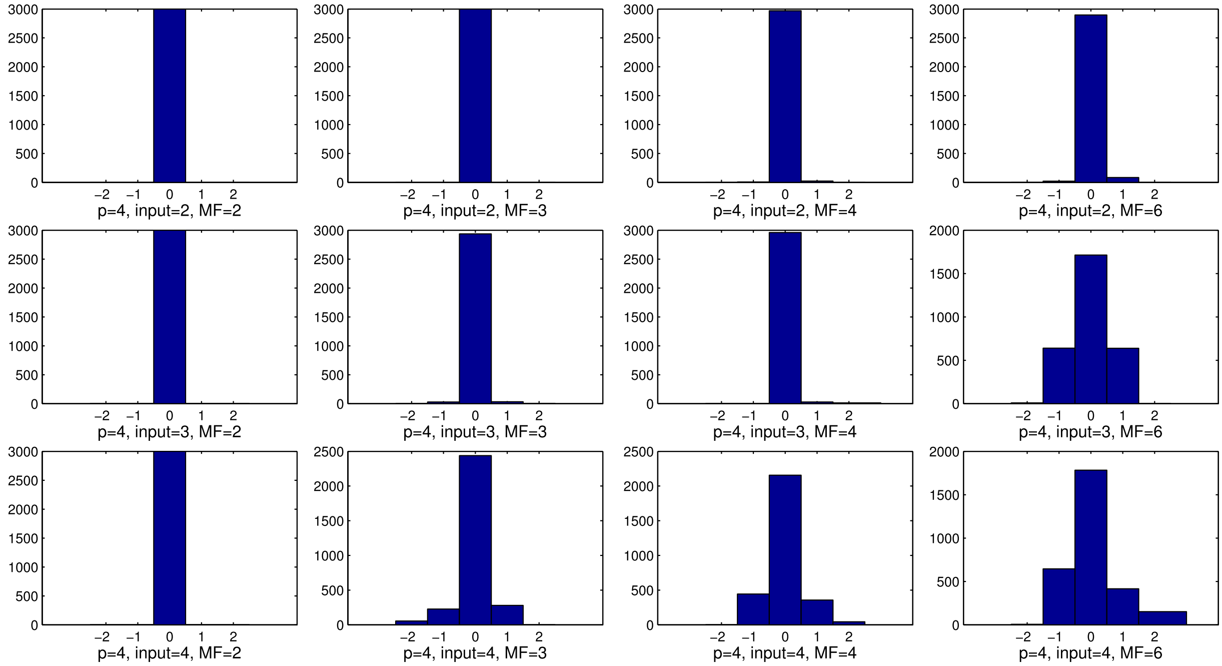
(b) Right switch point

Fig. 8. Prediction accuracy for 3rd degree polynomial.

with prior training by exhaustive defuzzification. Our simulation shows that proposed regression model can achieve very small MAPE values e.g. less than 1% for positive rule consequents and less than 2.5% for negative rule consequents. It is also demonstrated that regression model keeps the RMSE values very low, which is an additional indicator of good accuracy in centroid end-points approximation. It is also demonstrated that switch point prediction accuracy can be as high as 100%. Since regression modelling can be completed in design stage and it is possible to achieve high precision with very little effort, this study implies that type reduction block can be replaced by a properly trained and tuned approximation block in the implementation phase of interval type-2 fuzzy logic system. Therefore, this work demonstrates that the application



(a) Left switch point



(b) Right switch point

Fig. 9. Prediction accuracy for 4th degree polynomial.

specific tuning can remove the run time computational burden from standalone systems, without sacrificing substantial amount of accuracy, by adopting the proposed methodology in design phase.

The most significant contribution of this paper is that it demonstrates a simple data-driven TR approach is moderately accurate and very simple in terms of implementation. Theoretically, it has the potential of performing as fast as a type-1 defuzzifier because only a few multiplication and addition is necessary for defuzzification purpose in run-time. Therefore, proposed method addresses the question posed at the beginning of this paper about both minimal run-time complexity and acceptable error margin.

Table 1
Accuracy and error in switch point prediction.

P_{degree}	N_{input}	N_{MF}	$Accuracy_L$	$Accuracy_R$	$MAPE_L$	$MAPE_R$	λ_L	λ_R
2	2	2	100	100	0	0	9	9
		3	100	100	0	0	0	0
		4	95.6	98.93333333	3.25	1.066666667	0	0
		6	85.23333333	95.73333333	10.00555556	2.533333333	0	0
	3	2	100	100	0	0	1.5	1.5
		3	93.96666667	90.86666667	3.633333333	6.055555556	0	0
		4	95.4	82.56666667	1.265	3.836587302	0	0
		6	84.7	53.56666667	3.026984127	5.077585378	0	0
	4	2	100	100	0	0	9	9
		3	93.5	90.56666667	2.405555556	2.873333333	0	0
		4	67.73333333	62.8	4.296013616	2.510572869	0	0
		6	49.63333333	46.6	1.870346844	1.677369741	0	0
3	2	2	100	100	0	0	5	5
		3	100	100	0	0	9	9
		4	96.63333333	99.53333333	2.783333333	0.466666667	0	0
		6	94.83333333	94.56666667	3.133333333	5.211111111	0	0
	3	2	100	100	0	0	5	5
		3	91.86666667	95.46666667	5.822222222	2.094444444	0	0
		4	55.06666667	96.03333333	12.74388889	0.748571429	0	0
		6	53.06666667	73.26666667	8.373690476	3.033029101	0	0
	4	2	100	100	0	0	9	9
		3	95.4	88.06666667	1.569444444	2.402222222	0	0
		4	78.26666667	47.5	2.143381248	3.348384687	0	0
		6	81.66666667	60.8	0.739517867	1.077100407	0	0
4	2	2	100	100	0	0	9	9
		3	100	100	0	0	9	9
		4	94.96666667	99	3.616666667	0.883333333	0	0
		6	93.96666667	96.53333333	2.338888889	1.616666667	0	0
	3	2	100	100	0	0	5	5
		3	76.13333333	97.9	16.06666667	1.394444444	0	0
		4	75.23333333	98.63333333	7.286666667	0.41	0	0
		6	100	57.1	0	6.10030303	7	0
	4	2	100	100	0	0	5	5
		3	86.73333333	81.26666667	4.869444444	4.442777778	0	0
		4	95.5	71.9	0.589415954	2.098437118	0	0
		6	35.76666667	59.43333333	2.76102718	1.273215167	0	0

p_{degree}: polynomial degree in hypothesis, N_{input}: number of input in FLS, N_{MF}: number of MF on each input, $Accuracy_{L/R}$: prediction accuracy for left/right switch point, $MAPE_{L/R}$: mean absolute percentage error for left/right switch point, $\lambda_{L/R}$: optimum regulation parameter for left/right switch point prediction.

Acknowledgement

This work is supported by the Centre for Intelligent System Research at Deakin University.

References

[1] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, Inf. Sci. 8 (3) (1975) 199–249.
[2] S.-M. Chen, L.-W. Lee, Fuzzy multiple attributes group decision-making based on the interval type-2 topsis method, Expert Syst. Appl. 37 (4) (2010) 2790–2798.
[3] H.-J. Zimmermann, Fuzzy Sets, Decision Making, and Expert Systems, vol. 10, Springer Science & Business Media, 1987.
[4] S. Coupland, R. John, New geometric inference techniques for type-2 fuzzy sets, Int. J. Approx. Reason. 49 (1) (2008) 198–211, http://dx.doi.org/10.1016/j.ijar.2008.03.001, special Section on Logical Approaches to Imprecise Probabilities and Special Section on Imprecise Probabilities in Finance and Economics.
[5] F. Sabahi, M.-R. Akbarzadeh-T, Introducing validity in fuzzy probability for judicial decision-making, Int. J. Approx. Reason. 55 (6) (2014) 1383–1403, http://dx.doi.org/10.1016/j.ijar.2013.12.003.
[6] W. Wang, X. Liu, Y. Qin, Multi-attribute group decision making models under interval type-2 fuzzy environment, Knowl.-Based Syst. 30 (2012) 121–128.
[7] H.A. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Trans. Fuzzy Syst. 12 (4) (2004) 524–539.
[8] R. Martínez, O. Castillo, L.T. Aguilar, Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, Inf. Sci. 179 (13) (2009) 2158–2174.
[9] D. Wu, W. Tan, A type-2 fuzzy logic controller for the liquid-level process, in: 2004 IEEE International Conference on Fuzzy Systems, 2004. Proceedings, vol. 2, 2004, pp. 953–958.
[10] R. Sepúlveda, O. Castillo, P. Melin, A. Rodríguez-Díaz, O. Montiel, Experimental study of intelligent controllers under uncertainty using type-1 and type-2 fuzzy logic, Inf. Sci. 177 (10) (2007) 2023–2048.
[11] R.I. John, P.R. Innocent, M. Barnes, Neuro-fuzzy clustering of radiographic tibia image data using type 2 fuzzy sets, Inf. Sci. 125 (1) (2000) 65–82.
[12] A. Khosravi, S. Nahavandi, Effects of type reduction algorithms on forecasting accuracy of IT2FLS models, Appl. Soft Comput. 17 (2014) 32–38.
[13] O. Castillo, P. Melin, 17 adaptive noise cancellation using type-2 fuzzy logic and neural networks, in: Type-2 Fuzzy Logic: Theory and Applications, Springer, 2008, pp. 213–223.
[14] J.M. Mendel, Uncertain rule-based fuzzy logic system: introduction and new directions.

[15] S. Auephanwiriyakul, A. Adrian, J.M. Keller, Type 2 fuzzy set analysis in management surveys, in: Proceedings of the 2002 IEEE International Conference on Fuzzy Systems, 2002, FUZZ-IEEE'02, vol. 2, IEEE, 2002, pp. 1321–1325.
[16] A. Khosravi, S. Nahavandi, An interval type-2 fuzzy logic system-based method for prediction interval construction, Appl. Soft Comput. 24 (2014) 222–231.
[17] A. Khosravi, S. Nahavandi, D. Creighton, R. Naghavizadeh, Prediction interval construction using interval type-2 fuzzy logic systems, in: 2012 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, 2012, pp. 1–7.
[18] C. Hwang, F.-H. Rhee, Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means, IEEE Trans. Fuzzy Syst. 15 (1) (2007) 107–120.
[19] H. Hagras, D. Alghazzawi, G. Aldabbagh, Employing type-2 fuzzy logic systems in the efforts to realize ambient intelligent environments [application notes], IEEE Comput. Intell. Mag. 10 (1) (2015) 44–51.
[20] A. Khosravi, S. Nahavandi, D. Creighton, D. Srinivasan, Interval type-2 fuzzy logic systems for load forecasting: a comparative study, IEEE Trans. Power Syst. 27 (3) (2012) 1274–1282, http://dx.doi.org/10.1109/TPWRS.2011.2181981.
[21] N.N. Karnik, J.M. Mendel, Applications of type-2 fuzzy logic systems to forecasting of time-series, Inf. Sci. 120 (1) (1999) 89–111.
[22] K. Huarng, H.-K. Yu, A type 2 fuzzy time series model for stock index forecasting, Physica A 353 (2005) 445–462.
[23] A. Khosravi, S. Nahavandi, Load forecasting using interval type-2 fuzzy logic systems: optimal type reduction, IEEE Trans. Ind. Inform. 10 (2) (2014) 1055–1063.
[24] R. Dash, P. Dash, R. Bisoi, A differential harmony search based hybrid interval type2 fuzzy {EGARCH} model for stock market volatility prediction, Int. J. Approx. Reason. 59 (0) (2015) 81–104, http://dx.doi.org/10.1016/j.ijar.2015.02.001.
[25] P. Singh, B. Borah, Forecasting stock index price based on m-factors fuzzy time series and particle swarm optimization, Int. J. Approx. Reason. 55 (3) (2014) 812–833, http://dx.doi.org/10.1016/j.ijar.2013.09.014.
[26] W. Lu, X. Chen, W. Pedrycz, X. Liu, J. Yang, Using interval information granules to improve forecasting in fuzzy time series, Int. J. Approx. Reason. 57 (0) (2015) 1–18, http://dx.doi.org/10.1016/j.ijar.2014.11.002.
[27] J.M. Mendel, Type-2 fuzzy sets and systems: an overview, IEEE Comput. Intell. Mag. 2 (1) (2007) 20–29.
[28] Q. Liang, J.M. Mendel, Interval type-2 fuzzy logic systems: theory and design, IEEE Trans. Fuzzy Syst. 8 (5) (2000) 535–550.
[29] J. Mendel, R. John, F. Liu, Interval type-2 fuzzy logic systems made simple, IEEE Trans. Fuzzy Syst. 14 (6) (2006) 808–821, http://dx.doi.org/10.1109/TFUZZ.2006.879986.
[30] D. Wu, An overview of alternative type-reduction approaches for reducing the computational cost of interval type-2 fuzzy logic controllers, in: 2012 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE, IEEE, 2012, pp. 1–8.
[31] M. Nie, W.W. Tan, Towards an efficient type-reduction method for interval type-2 fuzzy logic systems, in: IEEE International Conference on Fuzzy Systems, 2008. FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), 2008, pp. 1425–1432.
[32] N.N. Karnik, J.M. Mendel, Centroid of a type-2 fuzzy set, Inf. Sci. 132 (1) (2001) 195–220.
[33] S. Greenfield, F. Chiclana, Accuracy and complexity evaluation of defuzzification strategies for the discretised interval type-2 fuzzy set, Int. J. Approx. Reason. 54 (8) (2013) 1013–1033.
[34] D.M. Bates, D.G. Watts, Nonlinear Regression: Iterative Estimation and Linear Approximations, Wiley Online Library, 1988.
[35] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.
[36] J.M. Mendel, Uncertain rule-based fuzzy logic system: introduction and new directions.
[37] D. Wu, J. Mendel, Enhanced Karnik–Mendel algorithms, IEEE Trans. Fuzzy Syst. 17 (4) (2009) 923–934, http://dx.doi.org/10.1109/TFUZZ.2008.924329.
[38] C.-Y. Yeh, W.-H. Jeng, S.-J. Lee, An enhanced type-reduction algorithm for type-2 fuzzy sets, IEEE Trans. Fuzzy Syst. 19 (2) (2011) 227–240.
[39] K. Duran, H. Bernal, M. Melgarejo, Improved iterative algorithm for computing the generalized centroid of an interval type-2 fuzzy set, in: Annual Meeting of the North American Fuzzy Information Processing Society, 2008. NAFIPS 2008, 2008, pp. 1–5.
[40] M. Melgarejo, A fast recursive method to compute the generalized centroid of an interval type-2 fuzzy set, in: Annual Meeting of the North American Fuzzy Information Processing Society, 2007. NAFIPS'07, IEEE, 2007, pp. 190–194.
[41] D. Wu, M. Nie, Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems, in: 2011 IEEE International Conference on Fuzzy Systems (FUZZ), IEEE, 2011, pp. 2131–2138.
[42] H. Hu, Y. Wang, Y. Cai, Advantages of the enhanced opposite direction searching algorithm for computing the centroid of an interval type-2 fuzzy set, Asian J. Control 14 (5) (2012) 1422–1430.
[43] C. Celemin, M. Melgarejo, A faster iterative computation of the centroid of an interval type-2 fuzzy set, in: 2012 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), 2012, pp. 1–6.
[44] S. Coupland, R. John, An investigation into alternative methods for the defuzzification of an interval type-2 fuzzy set, in: 2006 IEEE International Conference on Fuzzy Systems, IEEE, 2006, pp. 1425–1432.
[45] Q. Liang, J.M. Mendel, Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters, IEEE Trans. Fuzzy Syst. 8 (5) (2000) 551–563.
[46] H. Wu, J.M. Mendel, Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst. 10 (5) (2002) 622–639.
[47] M.B. Begian, W.W. Melek, J.M. Mendel, Stability analysis of type-2 fuzzy systems, in: IEEE International Conference on Fuzzy Systems, 2008. FUZZ-IEEE 2008 (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 947–953.
[48] S. Greenfield, F. Chiclana, S. Coupland, R. John, The collapsing method of defuzzification for discretised interval type-2 fuzzy sets, Inf. Sci. 179 (13) (2009) 2055–2069.
[49] C. Li, J. Yi, D. Zhao, A novel type-reduction method for interval type-2 fuzzy logic systems, in: Fifth International Conference on Fuzzy Systems and Knowledge Discovery, 2008, FSKD'08, vol. 1, IEEE, 2008, pp. 157–161.
[50] X. Du, H. Ying, Derivation and analysis of the analytical structures of the interval type-2 fuzzy-pi and pd controllers, IEEE Trans. Fuzzy Syst. 18 (4) (2010) 802–814.
[51] C.-W. Tao, J.-S. Taur, C.-W. Chang, Y.-H. Chang, Simplified type-2 fuzzy sliding controller for wing rock system, Fuzzy Sets Syst. 207 (2012) 111–129.
[52] S. Chakraborty, A. Konar, A. Ralescu, N.R. Pal, A fast algorithm to compute precise type-2 centroids for real-time control applications.
[53] R. Nowicki, On combining neuro-fuzzy architectures with the rough set theory to solve classification problems with incomplete data, IEEE Trans. Knowl. Data Eng. 20 (9) (2008) 1239–1253, http://dx.doi.org/10.1109/TKDE.2008.64.
[54] R. Nowicki, Rough neuro-fuzzy structures for classification with missing data, IEEE Trans. Syst. Man Cybern., Part B, Cybern. 39 (6) (2009) 1334–1347, http://dx.doi.org/10.1109/TSMCB.2009.2012504.