Computationally Efficient Type-Reduction Strategies for a Type-2 Fuzzy Logic Controller

Dongrui Wu
Department of Electrical and
Computer Engineering
National University of Singapore
Singapore 117576
E-mail: g0305786@nus.edu.sg

Woei Wan Tan
Department of Electrical and
Computer Engineering
National University of Singapore
Singapore 117576
E-mail: eletanww@nus.edu.sg

***Abstract*— A type-2 fuzzy set is characterized by a concept called footprint of uncertainty (FOU). It provides the extra mathematical dimension that equips type-2 fuzzy logic systems (FLSs) with the potential to outperform their type-1 counterparts. While a type-2 FLS has the capability to model more complex relationships, the output of a type-2 fuzzy inference engine needs to be type-reduced. As type-reduction is very computationally intensive, type-2 FLSs may not be suitable for certain real-time applications. This paper aims at developing more computationally efficient type-reducers. The proposed type-reducer is based on the concept known as *equivalent type-1 sets* (ET1Ss), a collection of type-1 sets that replicates the input-output map of a type-2 FLS. Simulations are presented to demonstrate that the proposed type-reducing algorithms have lower computational cost and better performances than the Karnik-Mendel type-reducer.**

I. Introduction

Unlike type-1 fuzzy sets whose membership grades are crisp numbers, the membership grades of a type-2 set are fuzzy sets in [0, 1]. It is useful in circumstances where it is difficult to determine the exact shape for a fuzzy set. Thus, type-2 fuzzy logic systems (FLSs), constructed by at least one type-2 set, have the potential to outperform their type-1 counterparts. Type-2 FLSs have been widely used so far [1]–[8].

The structure of a typical type-2 FLS is shown in Fig. 1. Compared with type-1 FLSs, the major difference is that an extra type-reducer is needed to convert the output of the fuzzy inference engine (type-2 sets) into a type-1 set so that it can be processed by the defuzzifier to give a crisp output. Unfortunately, existing type-reducers are very computationally intensive, rendering type-2 FLSs unsuitable for certain real-time applications.

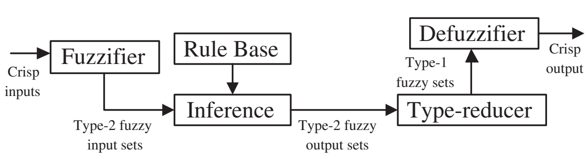


Fig. 1. A type-2 fuzzy logic system

In this paper, more computationally efficient methodologies for performing type-reduction are proposed. The algorithms utilize the concept of equivalent type-1 sets (ET1Ss) [9], [10]. Research has shown that a type-2 set may be replaced by a collection of ET1Ss without affecting the input-output relationship. The role of a type-reducer is to reduce a type-2 set to a type-1 set. By viewing a type-2 fuzzy set as a collection of ET1Ss, the type-reduction process then simplifies to deciding which ET1S to employ in a particular situation. Thus, the computational requirement can be reduced and the resulting type-2 FLSs would be more amenable to real-time embedded applications. Simulation results are presented to demonstrate the feasibility of the proposed idea.

The rest of the paper is organized as follows: Section II introduces the principle of the proposed type-reducer. Section III describes a simple new type-reducer constructed by experiences. Next, in Section IV the idea of evolving better type-reducer by genetic algorithm (GA) is introduced. The two new type-reducers are used to control a first order time delay system in Section V. Their performances are compared with a type-1 FLS and two type-2 FLSs with the Karnik-Mendel type-reducer and the uncertainty bound method. Finally, conclusions are drawn in Section VI.

II. Background and Motivations

A. Equivalent type-1 sets

The proposed type-reducing algorithm is based on the following concept [9], [10] :

> By definition, *equivalent type-1 sets* is the collection of type-1 sets that can be used in place of the FOUs in a type-2 FLS.

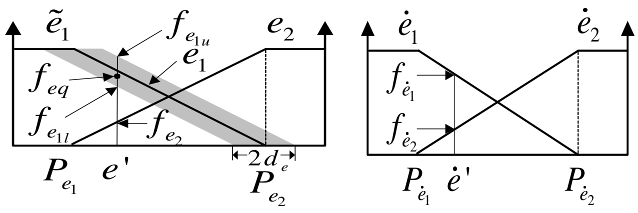
An example will be used to illustrate the ET1S concept.

Consider a two inputs, single output type-2 fuzzy logic controller (FLC) and an accompanying baseline type-1 FLC. Both FLCs have two inputs (e and $\dot{e}$) and one output ($\dot{u}$). Each input is characterized by two membership functions (MFs) in its domain. The MFs are shown in Fig. 2. The type-2 fuzzy set, $\tilde{e}_1$, is obtained by introducing FOU to a type-1 FLS, shown as the dark thick lines in Fig. 2. The type-2 fuzzy set used here is an interval one, where each point of the FOU has a unity secondary membership grade. Table I is the rule base of the

FLCs. The entries in Table I are defined as :

$$\dot{u}_{ij} = K_I \cdot P_{e_i} + K_P \cdot P_{\dot{e}_j} \qquad i, j = 1, 2 \qquad (1)$$

where P_{e_i} is the apex of MF e_i, $P_{\dot{e}_j}$ is the apex of MF $\dot{e}_j$, as labelled in Fig. 2. When the "Product-Sum-Gravity" inference is employed, the resulting type-1 FLC is equivalent to a PI controller with a proportional gain of K_P and an integral gain K_I [11].



(a) MFs of e (b) MFs of $\dot{e}$

Fig. 2. MFs of the FLSs

$e \setminus \dot{e}$	$\dot{e}_1$	$\dot{e}_2$
e_1	$\dot{u}_{11}$	$\dot{u}_{12}$
e_2	$\dot{u}_{21}$	$\dot{u}_{22}$

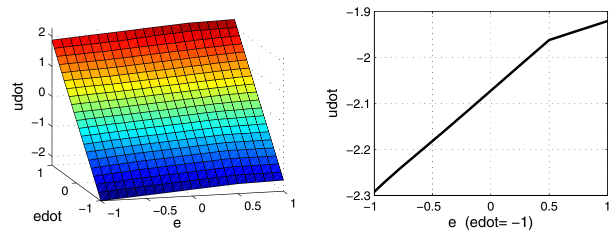
TABLE I
RULE BASE OF THE FLSs

Since the baseline type-1 FLC is actually a PI controller, its control surface is linear. The control surface of the type-2 FLC is more complex and nonlinear. For example, the control surface of the type-2 FLC with the following parameters is shown in Fig. 3(a) :

$$P_{e_1} = P_{\dot{e}_1} = -1, \quad P_{e_2} = P_{\dot{e}_2} = 1, \quad d_e = 0.5$$
$$K_I = 1, \quad K_P = 1$$

As it is more complex, the control surface in Fig. 3(a) may not be implemented by a type-1 FLC. However, the control surface can be cutted into numerous slices according to the input $\dot{e}$. That is, for a particular input $\dot{e} = \dot{e}'$, a curve representing the relationship between the output $\dot{u}$ and the input e can be obtained. Fig. 3(b) shows the slice corresponding to $\dot{e}' = -1$. As presented in [9], [10], each such slice can be replicated by constructing an ET1S to replace the type-2 set $\tilde{e}_1$.

Assume an input vector is $(e', \dot{e}')$. The MFs $\tilde{e}_1$, e_2, $\dot{e}_1$ and $\dot{e}_2$ are fired and the firing strengths are $f_{\tilde{e}_1} = [f_{e_{1l}}, f_{e_{1u}}]$, f_{e_2}, $f_{\dot{e}_1}$ and $f_{\dot{e}_2}$, respectively, where $f_{e_{1l}}$ and $f_{e_{1u}}$ are the firing strengths on the lower and upper MF of $\tilde{e}_1$. Suppose the interval firing strength $f_{\tilde{e}_1} = [f_{e_{1l}}, f_{e_{1u}}]$ is replaced by its *equivalent type-1 membership grade* [9], [10]. By fixing $\dot{e}'$ and varying e' in discrete steps from $P_{e_1} - d_e$ to $P_{e_2} + d_e$ (the FOU of $\tilde{e}_1$), all the *equivalent type-1 membership grades* will form a type-1 set. This type-1 set is the ET1S of $\tilde{e}_1$ corresponding to $\dot{e} = \dot{e}'$. The remaining slices of the control surface can be reconstructed by finding other ET1Ss. The ET1Ss corresponding



(a) Control surface of a type-2 FLS (b) A slice when $\dot{e} = -1$

Fig. 3. Illustration of the control surface and a slice of it

to $\dot{e}' = \{-1, 0, 1\}$ are plotted in Fig. 4. Note here the ET1S corresponding to $\dot{e}' = -1$ coincides with the one to $\dot{e}' = 1$.

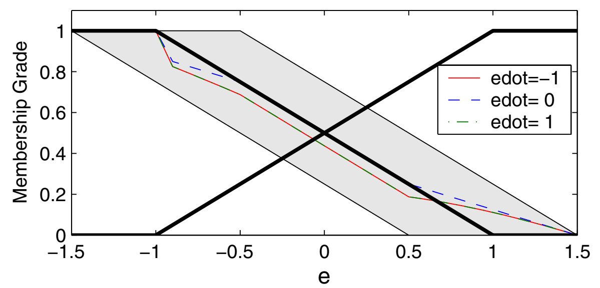


Fig. 4. ET1Ss obtained by the Karnik-Mendel type-reducer

B. Key Ideas of the Proposed Type-Reducers

Fig. 1 shows that a traditional type-reducer is placed after the inference engine. Consequently, both the inference engine and the type-reducer have to process interval firing strengths. This results in a heavy computational burden and may prevent type-2 FLSs from certain real-time applications.

The key idea behind the proposed type-reducer is to view a type-2 set as being equivalent to a collection of ET1Ss. Type reduction is then simplified to finding the ET1S corresponding to a particular input. More specifically, the type-reducer needs to identify the *equivalent type-1 membership grade* (f_{eq}) for each interval firing strength. Once the *equivalent type-1 membership grade* has been deduced, the firing set of a type-2 fuzzy set reduces to a crisp value and a traditional fuzzy inference engine and defuzzifier can be employed to find the output of the type-2 FLS. In summary, the proposed type-reduction procedure is applied before the inference engine, as illustrated in Fig. 5. The goal is to find the appropriate *equivalent type-1 membership grades* according to the inputs.

The new approach retains the characteristics of a type-2 FLS, while offering several advantages over existing techniques. First, the proposed algorithm can be much faster than the Karnik-Mendel iterative method. Second, the firing strengths of rules that the inference engine has to process are all crisp numbers now instead of interval sets. Thus, computational load is reduced because the inference engine behaves like the one in a type-1 FLS. Finally, the most significant advantage is that computational intelligence methods (i.e., Genetic Algorithms, Neural Networks, etc) can be employed to construct the type-reduction algorithm and/or optimize its parameters. This opens up a whole class of tools for developing type-reducers that satisfy specific requirements so that better performances can be achieved. In the following sections, two type-reduction procedures based on the ET1S concept will be described.

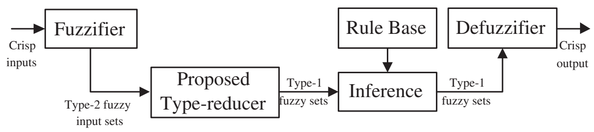


Fig. 5. Structure of a type-2 FLS with the proposed type-reducer

III. A Simple Computationally Efficient Type-Reducer (NewTR1)

From the observations made in [7]–[10], [12], a type-reducer in a type-2 FLC may satisfy the following requirements :

1) Sine a type-2 FLS reduces to its type-1 counterpart when the FOU is zero, a type-reducer must produce ET1Ss that coincide with the baseline type-1 sets in this case.
2) The ET1Ss change with the input. Hence, a type-reducer may be a function of all the input variables.
3) [7], [8], [12] show the control surface of a type-2 FLC is generally smoother than that of a type-1 FLC, especially around the origin ($e = 0$, $\dot{e} = 0$). The smoother control surface is one factor that makes a type-2 FLC more robust than its type-1 counterpart. The proposed type-reducer should, therefore, also give rise to control surfaces that are smoother.

By taking into account the above requirements and trial-and-error, a new type-reducer NewTR1 is one that defines the *equivalent type-1 set* as :

$$f_{eq} = f_u - \frac{1}{N}\sum_{i=1}^{N} rate_i \times (f_u - f_l) \qquad (2)$$

where f_{eq} is the *equivalent type-1 membership grade* of the interval firing strength $[f_l,\ f_u]$ (refer to Fig. 6) [9], [10], $rate_i$ is a function of the ith input and N is the total number of inputs.

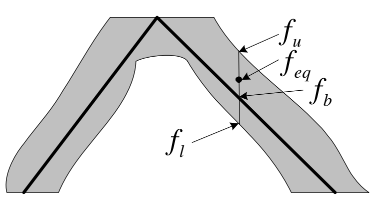


Fig. 6. Illustration of the new type-reducer

Analysis verifying that the algorithm defined in Equation (2) satisfies the three requirements of a type-reducer will now be presented.

1) Requirement (1) essentially states that f_{eq} must equal f_b in Fig. 6 when $f_u = f_l = f_b$. By setting $f_u = f_l = f_b$, it is obvious that Equation (2) satisfies this requirement.
2) The parameter $rate_i$ is used to satisfy the second requirement. For the FLCs given in Fig. 2, there are two inputs e and $\dot{e}$. Since the type-reducer should be a function of all input variables, two functions are defined :

$$rate_1 = \frac{2\,|e|}{P_{e_2} - P_{e_1}} \qquad (3)$$

$$rate_2 = \frac{2\,|\dot{e}|}{P_{\dot{e}_2} - P_{\dot{e}_1}} \qquad (4)$$

For different values of e, $rate_1$ is different. This is also true for $\dot{e}$. Since NewTR1 in Equation (2) is a function of $rate_1$ and $rate_2$, it is hence a function of both inputs e and $\dot{e}$ and the second requirement is fulfilled.

3) To demonstrate that the control surface obtained by using Equation (2) is smoother than the baseline type-1 FLC, the relationship between the slope of the control surface and the value of f_{eq} is examined. Using the structure in Fig. 2 and replacing the interval firing strength $f_{\tilde{e}_1}$ by its *equivalent type-1 membership grade* f_{eq}, the output is :

$$\dot{u} = \frac{f_{eq} f_{\dot{e}_1}\dot{u}_{11} + f_{eq} f_{\dot{e}_2}\dot{u}_{12} + f_{e_2} f_{\dot{e}_1}\dot{u}_{21} + f_{e_2} f_{\dot{e}_2}\dot{u}_{22}}{f_{eq} f_{\dot{e}_1} + f_{eq} f_{\dot{e}_2} + f_{e_2} f_{\dot{e}_1} + f_{e_2} f_{\dot{e}_2}} \qquad (5)$$

The first derivative of $\dot{u}$ with respect to f_{eq} is :

$$\ddot{u} = \frac{f_{e_2}[f_{\dot{e}_1}(\dot{u}_{11} - \dot{u}_{21}) + f_{\dot{e}_2}(\dot{u}_{12} - \dot{u}_{22})]}{(f_{eq} + f_{e_2})^2 (f_{\dot{e}_1} + f_{\dot{e}_2})} \qquad (6)$$

Substitute Equation (1) into Equation (6) :

$$\ddot{u} = \frac{K_I f_{e_2}(P_{e_1} - P_{e_2})}{(f_{eq} + f_{e_2})^2} \qquad (7)$$

Equation (7) shows that the slope of $|\dot{u}|$ will decrease as f_{eq} increases. To achieve fast and robust control, the control surface should have a small slope near the origin (steady state) and a big slope far from the origin. Consequently, the *equivalent type-1 membership grades* (f_{eq}) should be big when e and/or $\dot{e}$ are far from zero, and small when e and/or $\dot{e}$ are around zero. The ET1Ss corresponding to $\dot{e} = \{-1,\ 0,\ 1\}$ are plotted in Fig. 7. Note here the ET1S corresponding to $\dot{e} = -1$ coincides with the one to $\dot{e} = 1$. The plots in Fig. 7 show that the ET1Ss have the desired characteristics as f_{eq} is large when e and $\dot{e}$ is approximately zero and small when the inputs are far away from the origin. Hence, it may be concluded that the control surface would meet our requirements.

IV. A Type-Reducer Evolved by GA (NewTR2)

In this section, genetic algorithms (GAs) will be used to evolve an expression for performing type-reduction. Since a plant model must be used in the GA tuning process of NewTR2, it is introduced first.

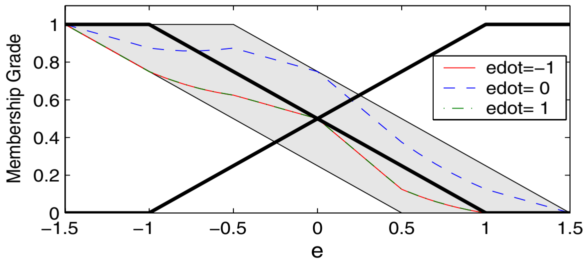


Fig. 7. ET1Ss obtained by the proposed type-reducer

A. Simulation Plant

Consider a first order plus dead-time plant :

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1} e^{-Ls} \quad (8)$$

where K, τ and L are the static gain, time constant and transportation delay respectively. It is assumed that the nominal plant is $\frac{1}{10s+1} e^{-2.5s}$. To ensure good control performance is obtained for the nominal plant, the PI parameters used to design the consequent sets of both FLCs are selected by the ITAE setpoint tracking tuning rule [13] :

$$K_P = \frac{0.586}{K} \left(\frac{L}{\tau} \right)^{-0.916} = 2.086 \quad (9)$$

$$K_I = \frac{1.03 - 0.165\frac{L}{\tau}}{\tau K_P} = 0.206 \quad (10)$$

The MFs of the baseline type-1 FLC used in the study are shown in Fig. 8 as the dark thick lines. The FOU of the type-2 MFs used to construct the type-2 FLCs are the shaded regions in Fig. 8. Substituting the PI parameters shown above into Equation (1), the consequent sets are found and are shown in Table II.

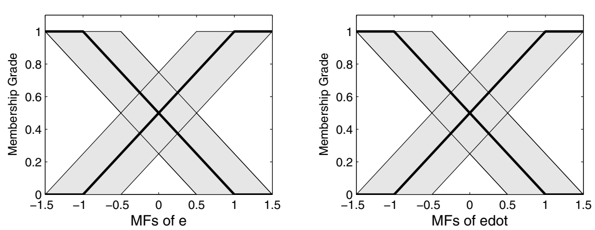


Fig. 8. MFs of the FLCs

$e \setminus \dot{e}$	$\dot{e}_1$	$\dot{e}_2$
e_1	−2.2923	1.8797
e_2	−1.8797	2.2923

TABLE II
RULE BASE OF THE FLSS

B. The Type-Reducer Evolved by GA (NewTR2)

For NewTR1 in Equation (2), the parameters $rate_1$ and $rate_2$ contribute equally to the value of f_{eq}. This is for simplicity sake. However, better performance may be obtained by weighting the contribution of $rate_1$ and $rate_2$. GA can be used for this purpose. Two weights are needed by each type-2 set and each type-2 MF may have its own weights. For the type-2 FLC used herein, there are 4 type-2 MFs. Thus, a total of 8 weights need to be tuned by GA.

In order to evolve a type-reducer that can cope well with modeling uncertainties, the 5 plants with parameters shown in Table III are used to tune the type-reducer. The fitness of a chromosome is evaluated based on the sum of the integral of time-weighted absolute error (ITAE) of the 5 plants. The best type-reducers evolved are :

$$\begin{aligned}
f_{e_1} &= f_{e_{1u}} - (1.4347 rate_1 - 3.8964 rate_2) \cdot (f_{e_{1u}} - f_{e_{1l}}) \\
f_{e_2} &= f_{e_{2u}} - (1.7605 rate_1 - 2.6043 rate_2) \cdot (f_{e_{2u}} - f_{e_{2l}}) \\
f_{\dot{e}_1} &= f_{\dot{e}_{1u}} - (0.9601 rate_1 + 0.2290 rate_2) \cdot (f_{\dot{e}_{1u}} - f_{\dot{e}_{1l}}) \\
f_{\dot{e}_2} &= f_{\dot{e}_{2u}} - (0.9041 rate_1 + 0.1169 rate_2) \cdot (f_{\dot{e}_{2u}} - f_{\dot{e}_{2l}})
\end{aligned}$$

where the definitions of $rate_1$ and $rate_2$ are the same as those in Equations (3) and (4).

Parameter \ Plant	I	II	III	IV	V
K	1	1	1	0.5	2
τ	10	5	20	10	10
L	2.5	2.5	2.5	2.5	2.5

TABLE III
PARAMETERS OF THE FIVE PLANTS

V. COMPARATIVE RESULTS

Consider the following FLCs :

- Type-1: A type-1 FLC realizing a PI controller with $K_P = 2.086$ and $K_I = 0.206$;
- K-M TR: A type-2 FLC using the Karnik-Mendel type-reducer;
- UnctnBound: A type-2 FLC using the uncertainty bound type-reducer [14];
- NewTR1: A type-2 FLC using the type-reducer developed in Section III;
- NewTR2: A type-2 FLC using the type-reducer evolved by GA in Section IV.

Their input MFs are shown in Fig. 8 and rule base in Table II. The performances of the 5 FLCs to handle modelling uncertainties are compared. The 5 plants given in Table III are used as testbeds. The step responses are shown in Fig. 9–13.

Three performance indices are employed as quantitative measures for comparing the 5 FLCs:

- Integral of the absolute error (IAE): IAE=$\int_0^{100} |e(t)| dt$.
- Integral of the squared error (ISE): ISE=$\int_0^{100} e^2(t) dt$.
- Integral of the time-weighted absolute error (ITAE): ITAE=$\int_0^{100} t|e(t)| dt$.

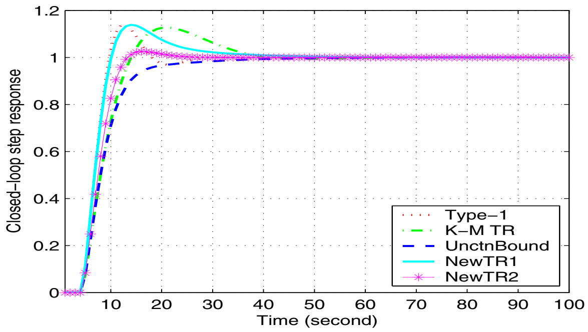


Fig. 9. Step response when $K = 1$, $\tau = 10$

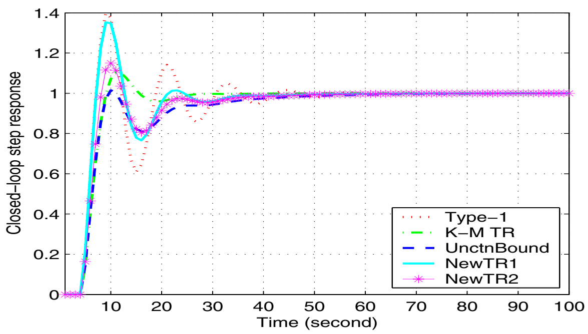


Fig. 10. Step response when $K = 1$, $\tau = 5$

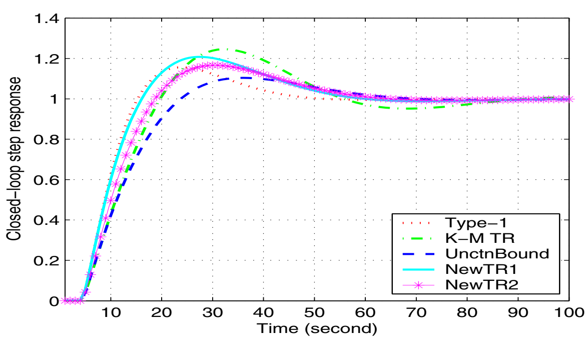


Fig. 11. Step response when $K = 1$, $\tau = 20$

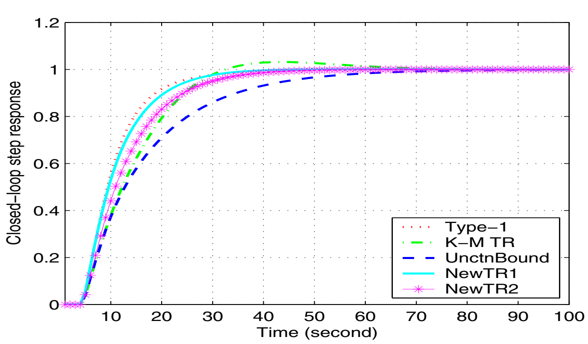


Fig. 12. Step response when $K = 0.5$, $\tau = 10$

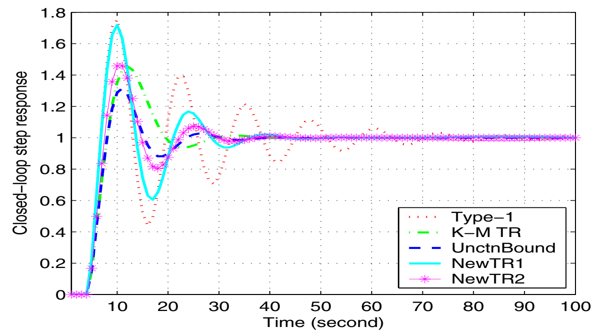


Fig. 13. Step response when $K = 2$, $\tau = 10$

The results are listed in Table IV. When their overall performances are compared, $NewTR2$ is the best of the five, $NewTR1$ is the second. This means the two type-2 FLCs with the proposed type-reducers can handle modelling uncertainties better than the type-1 FLC and the type-2 FLC with Karnik-Mendel type-reducer or uncertainty bound method. Besides, more interesting patterns may be found: when the response is fast, i.e., K is big (Plant V) or τ is small (Plant II), all the 4 type-2 FLCs outperform the type-1 FLC. On the other hand, when the response is slow, i.e., K is small (Plant IV) or τ is big (Plant III), the type-1 FLC outperforms all the 4 type-2 FLCs. The reason will be explored in a forthcoming paper.

(a) IAE of the 5 FLCs

FLC\Plant	I	II	III	IV	V	Sum
Type-1	6.2690	8.7621	10.5164	9.6941	15.3482	50.5898
K-M TR	8.7302	5.6858	16.2223	13.0920	8.2086	51.9389
UnctnBound	7.7965	7.6447	13.2338	15.7551	6.8963	51.3263
NewTR1	7.1429	7.2118	12.9132	10.0955	10.2925	47.6560
NewTR2	6.4481	7.0047	13.1466	11.8570	8.0851	46.5415

(b) ISE of the 5 FLCs

FLC\Plant	I	II	III	IV	V	Sum
Type-1	4.5960	4.8172	6.2603	6.4649	7.6374	29.7758
K-M TR	5.5494	4.2626	8.4429	8.4130	5.1966	31.8645
UnctnBound	5.4474	4.4632	7.8100	9.1500	4.4992	31.3697
NewTR1	4.6890	4.3552	6.7941	6.6841	6.1704	28.6928
NewTR2	5.0406	4.2950	7.3074	7.6748	4.9216	29.2395

(c) ITAE of the 5 FLCs

FLC\Plant	I	II	III	IV	V	Sum
Type-1	38.0564	104.2077	129.1421	85.3527	307.8952	664.6542
K-M TR	84.9387	32.8934	335.7619	157.1687	70.3821	681.1449
UnctnBound	58.2806	91.3164	200.4673	237.7789	52.7201	640.5633
NewTR1	60.4384	67.6799	222.7414	89.2998	112.6438	552.8033
NewTR2	34.8456	70.4150	215.6216	121.7163	72.5361	515.1346

TABLE IV
PERFORMANCES OF THE FIVE FLCS

An advantage of the proposed type-reducers is their low computational cost. Compared with the Karnik-Mendel type-reducer which requires several iterations and the number of iterations may be different from run to run, the new type-reducers are straight forward. The computational burden is fixed and is much less. Without loss of generality, assume that N equally spaced MFs are used to partition each of the two $[-1, 1]$ input domains. The FOU of every type-2 MF is defined as $d_e = d_{\dot{e}} = \frac{1}{N-1}$, i.e. half of the distance between the two adjacent apexes. This study is conducted by first generating 101 points, $e_i = 2(i-1)/100-1 (i = 1, \ldots, 101)$, that divide e domain into 100 equally-spaced intervals. Another 101 points in $\dot{e}$ domain are generated in a similar manner. By combining these points in all possible ways, 10201 input vectors are generated. Computational cost is evaluated by comparing the time needed to calculate the outputs corresponding to these 10201 input vectors. The platform is an Intel Pentium III 996MHz computer with 256M RAM and Windows XP running MATLAB 6.5. The computation time for the 5 FLCs are shown in Table V. It shows that a type-2 FLS with the proposed type-reducer has similar computational cost as a type-1 FLS. Compared to a type-2 FLS with Karnik-Mendel type-reducer, the computational burden is greatly reduced. Though the uncertainty bound method is specially designed for reducing computational cost, it is still about 2 times higher than that of the proposed type-reducers. Thus, the proposed type-reducers may be more suitable for certain types of real-time applications.

N \ FLC	Type-1	K-M TR	UnctnBound	NewTR1 (NewTR2)
2	1.0 sec	11.9 sec	2.5 sec	1.4 sec
3	1.2 sec	12.8 sec	2.8 sec	1.5 sec
5	1.6 sec	13.3 sec	3.4 sec	2.0 sec
7	2.3 sec	15.8 sec	4.9 sec	2.7 sec
9	3.2 sec	19.6 sec	7.0 sec	3.8 sec

TABLE V

COMPARISON OF COMPUTATIONAL COST

In [7] a simplified type-2 FLS structure is proposed to reduce the computational cost. However, that computational cost is still higher than the results here with the same number of input MFs. Besides, the simplified structure is a subset of the type-2 FLS with Karnik-Mendel type-reducer. Thus, its best performance is bounded by the type-2 FLS based on the Karnik-Mendel iterative method. On the other hand, the ideas in this paper enable one to design different type-reducers, and the performances of the resulting type-2 FLS may be better than a traditional type-2 FLS with Karnik-Mendel type-reducer.

There are, however, some limitations to the proposed type-reducers. In this paper the type-reducer transforms a type-2 FLC into a type-1 one before the inference engine. This approach gives rise to minimum computational cost. However, it does not allow the uncertainties to flow to the inference engine which presents a measure of uncertainty. Besides, the definitions of $rate_1$ and $rate_2$ are obtained from experience and only their coefficients are tuned by GA. It may be too constrained.

To overcome the limitations, one may find a function to replace the Karnik-Mendel type-reducer. This function also uses the type-2 output fuzzy sets from the inference engine as its input and outputs a type-1 set which will be used by the defuzzifier. However, it will calculate the type-1 set directly, without the iterations in the Karnik-Mendel type-reducer. This kind of type-reducers may have heavier computational cost than the two proposed in this paper since the inference engine also has to process type-2 sets. However, the computational cost is still much less than that of a Karnik-Mendel type-reducer. Besides, they seem more reasonable since the flow of uncertainties is the same as that in a traditional type-2 FLS. The uncertainty bound method can be considered as an example of this idea [14]. Better type-reducers may be found by genetic programming.

Finally, it should be note that there is no guarantee the two type-reducers proposed herein can be applied to all kinds of type-2 FLSs. However, their success in this paper suggests the feasibility of constructing faster and better type-reducers according to our specific requirements.

VI. CONCLUSIONS

In this paper, computationally efficient type-reducers are proposed. Simulation results show that they are much simpler to implement than the widely used Karnik-Mendel iterative method, while at the same time providing better performances. The results are promising and indicate that GA can be use to evolve faster and better type-reducers according to the specific requirements of a problem.

REFERENCES

[1] R. R. Yager, “Fuzzy subsets of type-2 in decisions,” *J. Cybern.*, vol. 10, pp. 137–159, 1980.

[2] H. Hagras, “A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots,” *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 4, pp. 524– 539, Aug. 2004.

[3] R. I. John, P. R. Innocent, and M. R. Barnes, “Neuro-fuzzy clustering of radiographic tibia images using type 2 fuzzy sets,” *Inform. Sci.*, vol. 125, pp. 65–82, 2000.

[4] J. M. Mendel, *Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. NJ: Prentice-Hall, 2001.

[5] S. Auephanwiriyakul, A. Adrian, and J. M. Keller, “Type-2 fuzzy set analysis in management surveys,” in *Proc. FUZZ-IEEE Conf.*, May 2002, pp. 1321–1325.

[6] C. H. Wang, C. S. Cheng, and T. T. Lee, “Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN),” *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 3, pp. 1462– 1477, June 2004.

[7] D. Wu and W. W. Tan, “A simplified architecture for type-2 FLSs and its application to nonlinear control,” in *Proceedings. IEEE Conf. On Cybern. and Intell. Syst.*, Singapore, Dec. 2004, pp. 485–490.

[8] ——, “Genetic learning and performance evaluation of type-2 fuzzy logic controllers,” *IEEE Trans. Syst., Man, Cybern. B*, 2004, submitted for publication.

[9] ——, “Type-2 FLS modeling capability analysis,” in *FUZZ-IEEE*, Reno, USA, May 2005.

[10] W. W. Tan and D. Wu, “Theory of equivalent type-1 fuzzy logic systems (ET1FLSs),” 2005, submitted to IEEE Trans. on Fuzzy Systems.

[11] M. Mizumoto, “Realization of PID controls by fuzzy control methods,” *Fuzzy sets and systems*, vol. 70, pp. 171–182, 1995.

[12] D. Wu and W. W. Tan, “A type-2 fuzzy logic controller for the liquid-level process,” in *Proceedings. FUZZ-IEEE*, vol. 2, Budapest, July 2004, pp. 953–958.

[13] P. Persson, “Towards autonomous PID control,” Ph.D. dissertation, Lund Institute of Technology, 1992.

[14] H. Wu and J. M. Mendel, “Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems,” *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 5, pp. 622–639, 2002.