2006 IEEE International Conference on Fuzzy Systems
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

Using Uncertainty Bounds in the Design of an Embedded Real-Time Type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines

Christopher Lynch, Hani Hagras, *Senior Member, IEEE*, Victor Callaghan

***Abstract*—Marine diesel engines operate in highly dynamic and uncertain environments, hence they require robust and accurate speed controllers that can handle the encountered uncertainties. Type-2 Fuzzy Logic Controllers (FLCs) can handle such uncertainties; however they have a computational overhead associated with the iterative type-reduction process which can diminish the FLC real-time performance. Furthermore, manually designing a type-2 FLC is a difficult task particularly as the number of membership function parameters and rules increase. In this paper, we will introduce an embedded Real-Time Type-2 Neuro-Fuzzy Controller (RT2NFC) which overcomes the iterative type-reduction overhead and learns the parameters of interval type-2 FLC for marine engines. We have performed numerous experiments on a real diesel engine testing platform in which we compared our RT2NFC to a T2NFC based on the iterative type reduction procedure. Both T2NFCs were embedded on an industrial microcontroller platform where they handled the uncertainties to produce accurate and robust speed controllers that outperformed the currently used commercial engine controller. The RT2NFC gave approximately the same control response as the T2NFC, whilst the RT2NFC avoided the type-reduction overhead thus giving a faster real-time response.**

I. Introduction

Marine diesel engines are utilised not only for ship propulsion but also for the on board power generation [1]. Fig. 1 illustrates an example of a marine load sharing arrangement for one of the world's largest suction hopper dredgers which employs three marine diesel engines to drive not only propellers but also generator sets [1]. An arrangement such as this can only fulfil its task properly if the speed control/governing system operates accurately and robustly.

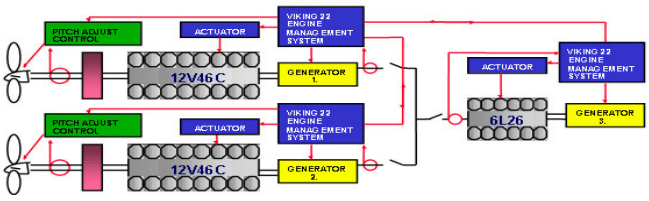


Fig. 1. Block diagram of load sharing arrangement

The engine speed is associated with fuel delivery which in turn is a function of a hydraulic servo actuator which is controlled electronically by the speed controller.

Accurate speed control of marine diesel engines is of critical importance as significant deviations from the speed set point could be detrimental and damaging to the engine and the respective loads [2]. Moreover, for applications such as power generation sets, the engine speed in rpm must be stable multiples of the generated base frequency i.e. a frequency of 50Hz would require the engine to operate at 1000 rpm, 1500 rpm etc. Hence, significant speed deviation can cause the generation of incorrect frequencies resulting in loss of synchronisation between the generator and associated power grid, which is obviously very problematic for any power generation system and coupled loads [2].

Robustness in speed control is required to overcome and recover quickly from the inherent instabilities and disturbances associated with the fast and dynamic changes of the environment, load and operation conditions [2]. For example, in case of sudden decrease in load (which is quite common in marine vessels due to propeller displacement as a result of rough sea conditions or in power generation when loads are commonly added/removed at will) the speed controller has to react very quickly by decreasing the fuel delivery otherwise the engine will continue to accelerate causing excessive overshoots and possible damage in a few seconds [2]. The ability to provide improved speed control response for marine diesel engines is not just desirable but a requirement of the British Standard BS5514 "Reciprocating internal combustion engines: Speed Governing", which details regulations concerning the speed controllers ability to recover from load changes and disturbances in terms of settling time and overshoot/undershoot [3].

Marine diesel engines operate in highly dynamic and uncertain environments experiencing vast changes in ambient temperature, fuel, humidity and load. This will lead to high uncertainty levels to be associated with the inputs and outputs of the speed controller as the sensors will be affected by high noise levels and their characteristics also change due to the diverse environmental conditions facing the engines, in addition the actuators characteristics change due to wear and tear or environmental changes. Moreover, there are also linguistic uncertainties as words mean different things to different people [4] and experts do not always agree on the design of the controller [5]. However, the uncertainties associated with the change in engine operation and load condition are considered to be the most dynamic and severe uncertainties that can affect both the inputs and outputs of

This work was supported by Regulateurs Europa Ltd.

C. Lynch, H. Hagras and V. Callaghan are with the Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK (phone: +44 1206 872770; fax: +44 1206 872788; e-mail: clynch@essex.ac.uk).

0-7803-9489-5/06/$20.00/©2006 IEEE

the controller to cause serious degradation in the performance of the engine [2]. For example the resistance (the force working against the ship propulsion) as a result of weather and sea variations could in general increase by as much as 50-100% of the total ship resistance in calm weather [2]. The increase in ship resistance can consequently cause drastic reduction of the ship speed [2]. Hence, choosing an appropriate speed control mechanism that is able to model and handle the encountered uncertainties to produce an accurate and robust control is of vital importance for marine engines. Moreover, the chosen control mechanism has to be computationally undemanding to be able to operate on the industrial embedded electronic platforms which have limited computational and memory capabilities that must typically be shared between speed control, alarm and monitoring, measurements, communications and signal conditioning.

Due to their simplicity and their suitability for the industrial embedded platforms, various forms of the PID controller have been used for speed control in marine diesel engines [6]. It has been shown that Fuzzy Logic Controllers (FLCs) and Fuzzy PID controllers can provide improved control over traditional PID controllers [6]. As a result, FLCs have found use in the speed control of various marine diesel engines [6], [7]. All the FLCs previously used in marine diesel engines were based on the traditional type-1 FLCs that use precise type-1 fuzzy sets. Type-1 fuzzy sets handle the uncertainties associated with the FLC inputs and outputs by using precise and crisp Membership Functions (MFs) that might be only valid under specific conditions [8]. However, because of the changes in these conditions and the associated uncertainties the designed crisp MFs might not best represent the associated linguistic label under the new conditions; this can cause degradation in the FLC performance resulting in poor control and inefficiency as well as continuous tuning of type-1 FLC [8]. Hence, type-1 FLCs cannot fully handle or accommodate the high levels of linguistic and numerical uncertainties present in the changing and dynamic marine engine environments.

Interval type-2 FLCs that use interval type-2 fuzzy sets have been proven to be able to model and handle the uncertainties to give a good performance that outperformed their type-1 counterparts [8], [9], [10], [11], [12]. However, the interval type-2 FLC involves a computational overhead associated with the computation of the type-reduced fuzzy sets using the iterative *Karnik-Mendel* procedure [13], [14]. This computational overhead translates to a protracted controller response which can diminish the robustness and the real-time performance of the type-2 FLC especially when operating on industrial embedded platforms (this is more evident for FLCs with large rule bases).

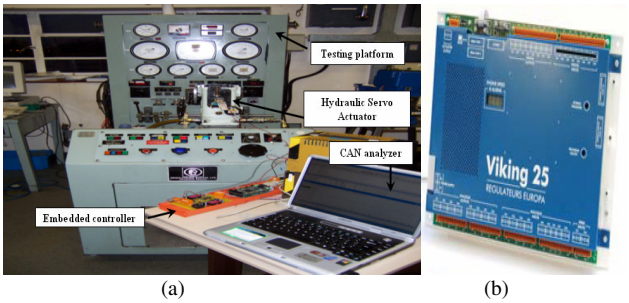
Wu and Mendel [15] introduced a method to approximate the type-reduced set by the inner and outer bound sets, thus avoiding the use of the iterative *Karnik-Mendel* procedure. In [9], we have shown that type-2 FLCs using the uncertainty bounds had the potential to provide a valid approximation from the control point of view to the type-2 FLC using the iterative type-reduction procedure. However, manually designing and tuning a type-2 FLC to give a response that follows the required industrial standards is a difficult task, particularly as the number of MFs parameters and rules increase. There has been other work that utilised neural based systems to learn the parameters of type-2 FLCs that were Mamdani based like [16], [17] or TSK based like [18]. However, to the author's knowledge no work has been done in employing neural based system to learn and tune type-2 FLCs that are embedded on industrial electronic platforms and are suitable for the real-time control of a heavy industrial application like marine diesel engines.

In this paper, we will introduce a neural based method to learn the parameters of a type-2 FLC that uses the uncertainty bounds to find quickly the FLC crisp outputs thus producing a Real-Time Type-2 Neuro Fuzzy Controller (RT2NFC) that was embedded on industrial microcontroller.

Section II, we will introduce the testing platform for marine engines. Section III will briefly review the interval type-2 fuzzy sets. Section IV will present the RT2NFC. Section V will present our experiments and results followed by the conclusions in section VI.

II. The Testing Platform

Due to the size and cost of the engines, it is important to test and verify the engine speed controllers under different operation and load conditions before deployment on the target engine. The speed controllers are tested and verified using the testing platform shown in Fig. 2a which is designed to realistically reflect the characteristics and operating conditions of the marine diesel engines; with the ability to alter speed, load, inertia and torque. The platform uses the same noisy sensors and actuators used on the engine with the ability to introduce the same uncertainty levels encountered on the real engines.



(a) (b)

Fig. 2. (a) The testing platform (b) Viking 25.

We have embedded both the RT2NFC and the T2NFC that uses type-reduction on the 32-bit Texas Instruments TMS320F2812 150MHz industrial DSP which was programmed in ANSI C and Assembly. Both the embedded RT2NFC and T2NFC govern the speed of the engine by controlling a hydraulic servo actuator which manages the rate of fuel delivery to the engine. Communication with the DSP was achieved via a Controller Area Network (CAN) bus

using Vectors automotive "CAN analyzer" software used in industry to observe, analyse and supplement data traffic on up to 32 CAN channels. The real engines use the Viking 25 (shown in Fig. 2b) which is an embedded commercial controller designed and marketed specifically for the control of diesel engines. The Viking 25 is based on a PID algorithm with various non-linear and gain scheduling functions

III. INTERVAL TYPE-2 FUZZY SETS

An interval type-2 fuzzy set $\tilde{F}$ can be written as follows:

$$\tilde{F} = \int_{x\in X}\left[\int_{u\in[\underline{\mu}_{\tilde{F}}(x),\overline{\mu}_{\tilde{F}}(x)]} 1/u\right]\Big/x \tag{1}$$

Where $\overline{\mu}_{\tilde{F}}(x)$, $\underline{\mu}_{\tilde{F}}(x)$ represent the upper and lower MFs respectively. For each input k and rule i, we will use the interval type-2 fuzzy set which is represented by Gaussian primary MF having uncertain mean m_k^i and uncertain standard deviation σ_k^i where $m_k^i \in [m_{k1}^i, m_{k2}^i]$ and $\sigma_k^i \in [\sigma_{k1}^i, \sigma_{k2}^i]$. The upper and lower MFs for this interval type-2 fuzzy set can be written as follows:

$$\overline{\mu}_{\tilde{F}_k^i}(x_k) = \begin{cases} N(m_{k1}^i, \sigma_{k2}^i; x_k) & x_k < m_{k1}^i \\ 1 & m_{k1}^i \le x_k \le m_{k2}^i \\ N(m_{k2}^i, \sigma_{k2}^i; x_k) & x_k > m_{k2}^i \end{cases} \tag{2}$$

$$\underline{\mu}_{\tilde{F}_k^i}(x_k) = \begin{cases} N(m_{k2}^i, \sigma_{k1}^i; x_k) & x_k \le \dfrac{m_{k1}^i + m_{k2}^i}{2} \\ N(m_{k1}^i, \sigma_{k1}^i; x_k) & x_k > \dfrac{m_{k1}^i + m_{k2}^i}{2} \end{cases} \tag{3}$$

Where $N(m_k^i, \sigma_k^i; x_k) = \exp\left[-\frac{1}{2}\left(\frac{x_k - m_k^i}{\sigma_k^i}\right)^2\right]$ (4)

When an input, x_k is located in a specific x-domain segment, we call its corresponding MF an *active branch* [14]. The MF is piece-wise differentiable i.e. each branch is differentiable over its segment domain. For example the piece-wise derivatives of Equations (2) and (3) with respect to m_{k1}^i and σ_{k1}^i are given as follows:

$$\frac{\partial\overline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} = \begin{cases} (x_k - m_{k1}^i)N(m_{k1}^i, \sigma_{k2}^i; x_k)/{\sigma_{k2}^i}^2 & x_k < m_{k1}^i \\ 0 & m_{k1}^i \le x_k \le m_{k2}^i \\ 0 & x_k > m_{k2}^i \end{cases} \tag{5}$$

$$\frac{\partial\underline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} = \begin{cases} 0 & x_k \le \dfrac{m_{k1}^i + m_{k2}^i}{2} \\ (x_k - m_{k1}^i)N(m_{k1}^i, \sigma_{k1}^i; x_k)/{\sigma_{k1}^i}^2 & x_k > \dfrac{m_{k1}^i + m_{k2}^i}{2} \end{cases} \tag{6}$$

$$\frac{\partial\underline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial \sigma_{k1}^i} = \begin{cases} (x_k - m_{k2}^i)^2 N(m_{k2}^i, \sigma_{k1}^i; x_k)/{\sigma_{k1}^i}^3 & x_k \le \dfrac{m_{k1}^i + m_{k2}^i}{2} \\ (x_k - m_{k1}^i)^2 N(m_{k1}^i, \sigma_{k1}^i; x_k)/{\sigma_{k1}^i}^3 & x_k > \dfrac{m_{k1}^i + m_{k2}^i}{2} \end{cases} \tag{7}$$

$$\frac{\partial\overline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial \sigma_{k1}^i} = 0 \tag{8}$$

IV. THE REAL-TIME TYPE-2 NEURO FUZZY CONTROLLER

A. The RT2NFC Structure

Fig. 3 shows the structure of the proposed RT2NFC which mimics the operation of the interval type-2 FLC. However, the RT2NFC employs the Wu-Mendel Uncertainty Bounds method which is used to approximate the type-reduced set and can further be used to directly derive the defuzzified output [15]. Consequently this will enable us to avoid the computational overhead associated with using iterative *Karnik-Mendel* procedure for type-reduction and hence the RT2NFC will be able to provide real-time operation for embedded controllers operating in marine diesel engines.

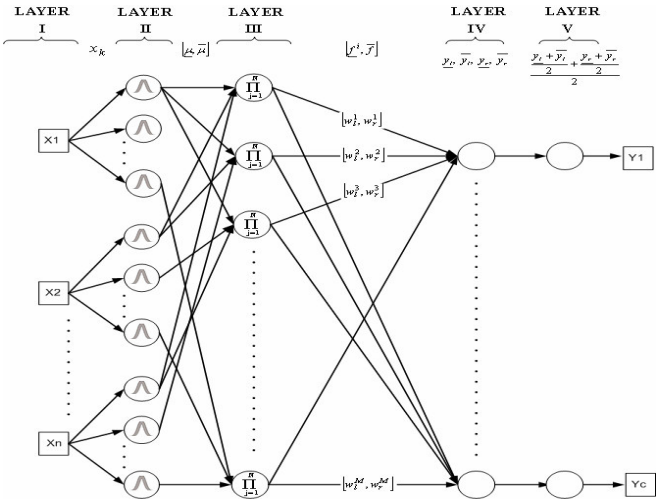


Fig 3. The Real-Time Type-2 Neuro-Fuzzy Controller

Layer I of the RT2NFC is the *Input Layer* which interfaces to the crisp inputs. Layer II is the *Fuzzification Layer* which maps a crisp input to a type-2 fuzzy set. Singleton fuzzification was chosen due to its low computational burden. Using singleton fuzzification, the upper $\overline{\mu}_{\tilde{F}_k^i}(x_k)$ and lower $\underline{\mu}_{\tilde{F}_k^i}(x_k)$ membership values are calculated using Equations (2) and (3) respectively. Layer III is the *Inference and Rule Base Layer,* where each node in layer III represents a given rule connecting antecedents to consequents. The i^{th} rule ($i=1...M$, where M is the number of rules) which has n inputs and c outputs can be written as follows:

$$R^i{}_{MIMO}: IF\ x_1\ is\ \tilde{F}_1^i\ ...and\ x_n\ is\ \tilde{F}_n^i\ THEN\ y_1\ is\ [w_{l1}^i, w_{r1}^i], ..., y_c\ is\ [w_{lc}^i, w_{rc}^i] \tag{9}$$

Where $[w_l^i, w_r^i]$ represent the centroid interval set of the consequent type-2 fuzzy set for a given output of the i^{th} rule. Layer III combines rules and gives a mapping from input type-2 sets to output type-2 sets. The firing strength f^i of the i^{th} rule is an interval type-1 set determined by its left most point $\underline{f}^i$ and its right most point $\overline{f}^i$ which are calculated as follows [14]:

$$\underline{f}^i = \underline{\mu}_{\tilde{F}_1^i}(x_1) * \ldots\ldots * \underline{\mu}_{\tilde{F}_n^i}(x_n) = \prod_{j=1}^{n} \underline{\mu}_{\tilde{F}_j^i}(x_j) \quad (10)$$

$$\overline{f}^i = \overline{\mu}_{\tilde{F}_1^i}(x_1) * \ldots\ldots * \overline{\mu}_{\tilde{F}_n^i}(x_n) = \prod_{j=1}^{n} \overline{\mu}_{\tilde{F}_j^i}(x_j) \quad (11)$$

Where * denotes the product t-norm. Layer IV approximates the type-reduced set using the Wu-Mendel Uncertainty Bounds which provides mathematical formulas for the inner and outer bound sets which can be used to approximate the type-reduced set [15]. Equations (12) and (13) define the inner bound set while (14) and (15) define the outer bound set [15].

$$\underline{y}_l = \min\left\{\frac{\sum_{i=1}^{M}\underline{f}^i w_l^i}{\sum_{i=1}^{M}\underline{f}^i}, \frac{\sum_{i=1}^{M}\overline{f}^i w_l^i}{\sum_{i=1}^{M}\overline{f}^i}\right\} \quad (12)$$

$$\overline{y}_r = \max\left\{\frac{\sum_{i=1}^{M}\underline{f}^i w_r^i}{\sum_{i=1}^{M}\underline{f}^i}, \frac{\sum_{i=1}^{M}\overline{f}^i w_r^i}{\sum_{i=1}^{M}\overline{f}^i}\right\} \quad (13)$$

$$\underline{y}_l = \overline{y}_l - \left[\frac{\sum_{i=1}^{M}(\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{M}\overline{f}^i \sum_{i=1}^{M}\underline{f}^i} * \frac{\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1)\sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)}{\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1) + \sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)}\right] \quad (14)$$

$$\overline{y}_r = \underline{y}_r + \left[\frac{\sum_{i=1}^{M}(\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{M}\overline{f}^i \sum_{i=1}^{M}\underline{f}^i} * \frac{\sum_{i=1}^{M}\overline{f}^i(w_r^i - w_r^1)\sum_{i=1}^{M}\underline{f}^i(w_r^M - w_r^i)}{\sum_{i=1}^{M}\overline{f}^i(w_r^i - w_r^1) + \sum_{i=1}^{M}\underline{f}^i(w_r^M - w_r^i)}\right] \quad (15)$$

Layer V is the *Defuzzification Layer* which calculates the crisp outputs as follows:

$$y(\vec{x}) = \frac{\frac{\underline{y}_l + \overline{y}_l}{2} + \frac{\underline{y}_r + \overline{y}_r}{2}}{2} \quad (16)$$

The RT2NFC operates as an interval type-2 FLC in the forward mode mapping crisp inputs to crisp outputs. In order, to ensure that the defuzzified outputs of the type-2 FLC using the uncertainty bounds method are very similar to the outputs of the type-2 FLC using type-reduction, we have used a neural based system to learn the parameters of both type-2 FLCs such that both FLCs will give very close outputs similar to those of the currently used commercial control.

B. Learning the Type-2 FLC Parameters

To ensure that the type-2 FLC learnt by the RT2NFC is suitable for the speed control of marine diesel engines, we have to make sure that its performance will be at least as good as the currently used Viking 25 commercial controller. Hence, the RT2NFC will be trained with input/output data captured from the Viking 25 so that a satisfactorily trained RT2NFC will produce a type-2 FLC that will have the same performance as the Viking 25 controller under no or limited uncertainty levels associated with the change in engine operation and load conditions. However, as these uncertainty levels increase the type-2 FLC will outperform the Viking 25 controller as will be shown in the experiments section.

The training data (10,000 samples) for the RT2NFC was split into an estimation set (6000 samples) and a validation set (4000 samples). The estimation set was used to train the RT2NFC whilst the validation set was used to avoid over-fitting the data where the training was stopped periodically (every 2 epochs) and the network was tested on the validation set by calculating the Scaled Root Mean Squared Error (SRMSE). The SRMSE decreases monotonically to a minimum point at which it starts to increase as the training continues [19]. Training beyond this minimum point is essentially learning from noise contained within the estimation set [19]; thus stopping the training at this point allows us to avoid overfitting of the estimation set.

For the i^{th} rule, the RT2NFC learns the necessary parameters for each input k antecedent fuzzy sets which are $m_{k1}^i, m_{k2}^i, \sigma_{k1}^i, \sigma_{k2}^i$, as well as the consequent parameters for each output w_l^i, w_r^i. By using the Back Propagation (BP) method, for P input-output training data supplied from the Viking 25 $(\vec{x}^p : d^p)$, $p=1,.P$ we tune the design parameters so the following error function can be minimized:

$$e^p = \frac{1}{2}\left[y(\vec{x}^p) - d^p\right]^2 \quad p = 1,.... P \quad (17)$$

By using α to represent the learning rate for the BP tuning then w_l^i can be tuned as follows:

$$w_l^i(p+1) = w_l^i(p) - \alpha \left.\frac{\partial e^p}{\partial w_l^i}\right|_p$$

$$= w_l^i(p) - \alpha\left[\frac{\partial e^p}{\partial y(\vec{x})}\frac{\partial y(\vec{x})}{\partial \underline{y}_l}\frac{\partial \underline{y}_l}{\partial w_l^i} + \frac{\partial e^p}{\partial y(\vec{x})}\frac{\partial y(\vec{x})}{\partial \overline{y}_l}\frac{\partial \overline{y}_l}{\partial w_l^i}\right]_p \quad (18)$$

Where: $\left.\frac{\partial e^p}{\partial y(\vec{x})}\right|_p = y(\vec{x}^p) - d^p$, $\left.\frac{\partial y(\vec{x})}{\partial \underline{y}_l}\right|_p = \frac{1}{4}$, $\left.\frac{\partial y(\vec{x})}{\partial \overline{y}_l}\right|_p = \frac{1}{4}$

If $\overline{y}_l = \min\left\{\frac{\sum_{i=1}^{M}\underline{f}^i w_l^i}{\sum_{i=1}^{M}\underline{f}^i}, \frac{\sum_{i=1}^{M}\overline{f}^i w_l^i}{\sum_{i=1}^{M}\overline{f}^i}\right\} = \frac{\sum_{i=1}^{M}\underline{f}^i w_l^i}{\sum_{i=1}^{M}\underline{f}^i}$, then $\frac{\partial \overline{y}_l}{\partial w_l^i} = \frac{\underline{f}^i}{\sum_{i=1}^{M}\underline{f}^i}$,

Otherwise $\frac{\partial \overline{y}_l}{\partial w_l^i} = \frac{\overline{f}^i}{\sum_{i=1}^{M}\overline{f}^i}$ (19)

$$\frac{\partial \underline{y}_l}{\partial w_l^i} = \frac{\partial \overline{y}_l}{\partial w_l^i} - \left[\frac{\sum_{i=1}^{M}(\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{M}\overline{f}^i \sum_{i=1}^{M}\underline{f}^i} * \left[\frac{\left(\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1)\right)\left(-\overline{f}^i\right) + \left(\sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)\right)\left(\underline{f}^i\right)}{\left(\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1) + \sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)\right)} - \frac{\left(\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1)\sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)\right)\left(\underline{f}^i - \overline{f}^i\right)}{\left(\sum_{i=1}^{M}\underline{f}^i(w_l^i - w_l^1) + \sum_{i=1}^{M}\overline{f}^i(w_l^M - w_l^i)\right)^2}\right]\right] \quad (20)$$

Similarly we can derive the BP update equation for w_r^i whilst m_{k1}^i can be tuned as follows:

$$m_{k1}^i(p+1) = m_k^i(p) - \alpha \left.\frac{\partial e^p}{\partial m_{k1}^i}\right|_p$$

$$= m_{k1}^i(p) - \alpha \left[\begin{array}{l} \frac{\partial e^p}{\partial y(\vec{x})} \frac{\partial y(\vec{x})}{\partial \underline{y}_l} \frac{\partial \underline{y}_l}{\partial m_{k1}^i} + \frac{\partial e^p}{\partial y(\vec{x})} \frac{\partial y(\vec{x})}{\partial \overline{y}_l} \frac{\partial \overline{y}_l}{\partial m_{k1}^i} + \\ \frac{\partial e^p}{\partial y(\vec{x})} \frac{\partial y(\vec{x})}{\partial \underline{y}_r} \frac{\partial \underline{y}_r}{\partial m_{k1}^i} + \frac{\partial e^p}{\partial y(\vec{x})} \frac{\partial y(\vec{x})}{\partial \overline{y}_r} \frac{\partial \overline{y}_r}{\partial m_{k1}^i} \end{array} \right]_p \quad (21)$$

Where $\left.\frac{\partial e^p}{\partial y(\vec{x})}\right|_p = y(\vec{x}^p) - d^p, \left.\frac{\partial y(\vec{x})}{\partial \underline{y}_l}\right|_p = \frac{1}{4}, \left.\frac{\partial y(\vec{x})}{\partial \overline{y}_l}\right|_p = \frac{1}{4}, \left.\frac{\partial y(\vec{x})}{\partial \underline{y}_r}\right|_p = \frac{1}{4}, \left.\frac{\partial y(\vec{x})}{\partial \overline{y}_r}\right|_p = \frac{1}{4}$

$$\left.\frac{\partial \underline{y}_l}{\partial m_{k1}^i}\right|_p = \left[\frac{\partial \underline{y}_l}{\partial \overline{\mu}_{F_k^i}(x_k)} \frac{\partial \overline{\mu}_{F_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial \underline{y}_l}{\partial \underline{\mu}_{F_k^i}(x_k)} \frac{\partial \underline{\mu}_{F_k^i}(x_k)}{\partial m_{k1}^i} \right] \quad (22)$$

$$\left.\frac{\partial \overline{y}_l}{\partial m_{k1}^i}\right|_p = \left[\frac{\partial \overline{y}_l}{\partial \overline{\mu}_{F_k^i}(x_k)} \frac{\partial \overline{\mu}_{F_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial \overline{y}_l}{\partial \underline{\mu}_{F_k^i}(x_k)} \frac{\partial \underline{\mu}_{F_k^i}(x_k)}{\partial m_{k1}^i} \right] \quad (23)$$

$$\left.\frac{\partial \underline{y}_r}{\partial m_{k1}^i}\right|_p = \left[\frac{\partial \underline{y}_r}{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)} \frac{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial \underline{y}_r}{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)} \frac{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} \right] \quad (24)$$

$$\left.\frac{\partial \overline{y}_r}{\partial m_{k1}^i}\right|_p = \left[\frac{\partial \overline{y}_r}{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)} \frac{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial \overline{y}_r}{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)} \frac{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} \right] \quad (25)$$

If $\underline{y}_l = \dfrac{\sum_{i=1}^{M} \underline{f}^i w_l^i}{\sum_{i=1}^{M} \underline{f}^i}$, then

$$\frac{\partial \overline{y}_l}{\partial \underline{\mu}_{F_k^i}(x_k)} = \left[\frac{\left(\sum_{i=1}^{M} \underline{f}^i\right)\left(\prod_{j=1 j\neq k}^{N} \underline{\mu}_{\tilde{F}_j^i} w_l^i\right) - \left(\sum_{i=1}^{M} \underline{f}^i w_l^i\right)\left(\prod_{j\neq k}^{N} \underline{\mu}_{\tilde{F}_j^i}\right)}{\left(\sum_{i=1}^{M} \underline{f}^i\right)^2} \right]$$

Otherwise $\dfrac{\partial \overline{y}_l}{\partial \underline{\mu}_{F_k^i}(x_k)} = 0$ (26)

If $\underline{y}_l = \dfrac{\sum_{i=1}^{M} \overline{f}^i w_l^i}{\sum_{i=1}^{M} \overline{f}^i}$, then

$$\frac{\partial \overline{y}_l}{\partial \overline{\mu}_{F_k^i}(x_k)} = \left[\frac{\left(\sum_{i=1}^{M} \overline{f}^i\right)\left(\prod_{j=1 j\neq k}^{N} \overline{\mu}_{\tilde{F}_j^i} w_l^i\right) - \left(\sum_{i=1}^{M} \overline{f}^i w_l^i\right)\left(\prod_{j\neq k}^{N} \overline{\mu}_{\tilde{F}_j^i}\right)}{\left(\sum_{i=1}^{M} \overline{f}^i\right)^2} \right]$$

Otherwise $\dfrac{\partial \overline{y}_l}{\partial \overline{\mu}_{F_k^i}(x_k)} = 0$ (27)

$$\frac{\partial \underline{y}_l}{\partial \overline{\mu}_{F_k^i}(x_k)} = \frac{\partial \overline{y}_l}{\partial \overline{\mu}_{F_k^i}(x_k)} -$$

$$\left[\begin{array}{l} \frac{\sum_{i=1}^{M} (\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{M} \overline{f}^i \sum_{i=1}^{M} \underline{f}^i} * \\ \frac{\left[\left(\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) + \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j) \right) \left(\left(\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) \right) \left(\prod_{j\neq k}^{N} \overline{\mu}_{\tilde{F}_j^i} (w_l^M - w_l^j) \right) \right) - \right.}{\left. \left(\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) \right) \left(\sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j) \right) \left(\prod_{j\neq k}^{N} \overline{\mu}_{\tilde{F}_j^i} (w_l^M - w_l^j) \right) \right]} \\ \left(\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) + \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j) \right)^2 \\ + \\ \frac{\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j)}{\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) + \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j)} * \\ \frac{\left(\sum_{i=1}^{M} \overline{f}^i \sum_{i=1}^{M} \underline{f}^i \left(\prod_{j\neq k}^{N} \overline{\mu}_{\tilde{F}_j^i} \right) - \left(\sum_{i=1}^{M} \overline{f}^i - \underline{f}^i \right) \left(\left(\sum_{i=1}^{M} \underline{f}^i \right) \left(\prod_{j\neq k}^{N} \overline{\mu}_{\tilde{F}_j^i} \right) \right) \right)}{\left(\sum_{i=1}^{M} \overline{f}^i \sum_{i=1}^{M} \underline{f}^i \right)^2} \end{array} \right] \quad (28)$$

$$\frac{\partial \underline{y}_l}{\partial \underline{\mu}_{F_k^i}(x_k)} = \frac{\partial \overline{y}_l}{\partial \underline{\mu}_{F_k^i}(x_k)} -$$

$$\left[\begin{array}{l} \frac{\sum_{i=1}^{M} (\overline{f}^i - \underline{f}^i)}{\sum_{i=1}^{M} \overline{f}^i \sum_{i=1}^{M} \underline{f}^i} * \\ \frac{\left[\left(\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) + \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j) \right) \left(\left(\sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j) \right) \left(\prod_{j\neq k}^{N} \underline{\mu}_{\tilde{F}_j^i} (w_l^j - w_l^1) \right) \right) - \right.}{\left. \left(\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) \right) \left(\sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j) \right) \left(\prod_{j\neq k}^{N} \underline{\mu}_{\tilde{F}_j^i} (w_l^j - w_l^1) \right) \right]} \\ \left(\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) + \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j) \right)^2 \\ + \\ \frac{\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j)}{\sum_{i=1}^{M} \underline{f}^i (w_l^j - w_l^1) + \sum_{i=1}^{M} \overline{f}^i (w_l^M - w_l^j)} * \\ \frac{\left(\sum_{i=1}^{M} \overline{f}^i \sum_{i=1}^{M} \underline{f}^i \left(\prod_{j\neq k}^{N} \underline{\mu}_{\tilde{F}_j^i} \right) - \left(\sum_{i=1}^{M} \overline{f}^i - \underline{f}^i \right) \left(\left(\sum_{i=1}^{M} \overline{f}^i \right) \prod_{j\neq k}^{N} \underline{\mu}_{\tilde{F}_j^i} \right) \right)}{\left(\sum_{i=1}^{M} \overline{f}^i \sum_{i=1}^{M} \underline{f}^i \right)^2} \end{array} \right] \quad (29)$$

By following a similar procedure as above we can derive:

$\dfrac{\partial \underline{y}_r}{\partial \underline{\mu}_{F_k^i}(x_k)}, \dfrac{\partial \underline{y}_r}{\partial \overline{\mu}_{F_k^i}(x_k)}, \dfrac{\partial \overline{y}_r}{\partial \underline{\mu}_{F_k^i}(x_k)}$ and $\dfrac{\partial \overline{y}_r}{\partial \overline{\mu}_{F_k^i}(x_k)}$

Similarly we can derive the BP update equation for m_{k2}^i and further $\sigma_{k1}^i, \sigma_{k2}^i$.

Finding $\dfrac{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i}$ and $\dfrac{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i}$ would require finding

the active branches of the MFs and using the piece wise derivatives in (5) and (6). For example if $x_k < m_{k1}^i$ then:

$$\frac{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} = (x_k - m_{k1}^i) N(m_{k1}^i, \sigma_{k2}^i; x_k) / \sigma_{k2}^i \quad (30)$$

$$\frac{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} = 0 \quad (31)$$

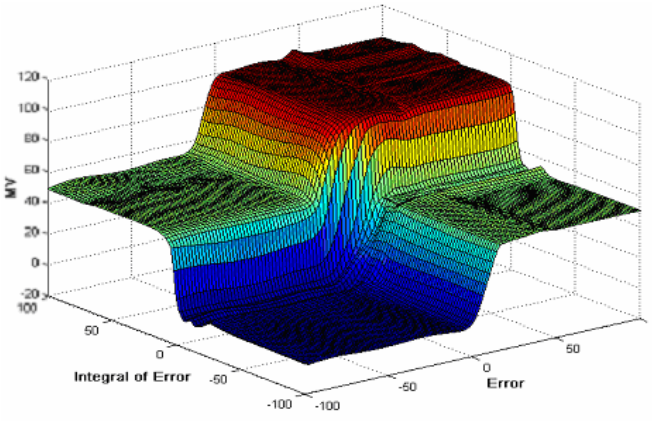
Training RT2NFC is simpler than the T2NFC that uses the iterative type-reduction as T2NFC necessitates keeping track in memory of the parameters L and R for each training pattern that was computed and used in the forward pass [20].

V. Experiments and Results

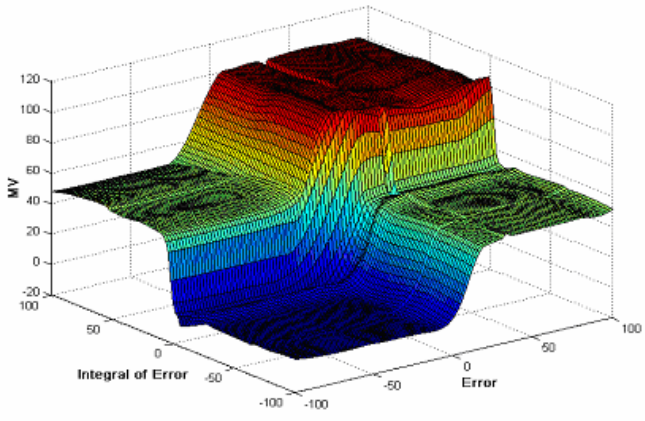
In this section, we will compare the performance of the RT2NFC (using uncertainty bounds) and the T2NFC (using type-reduction) against the Viking 25 commercial controller that is currently used in the speed control of marine diesel engines. We will start first by comparing the control surfaces produced by the RT2NFC and the T2NFC and we will show that both surfaces are approximately the same indicating that they will give very similar responses. We will then compare the control performances of the RT2NFC, T2NFC and the Viking 25 in handling the uncertainties and disturbances that are associated with the change of operation and load conditions. We will then demonstrate that although the control performance of both the RT2NFC and the T2NFC are approximately similar, the RT2NFC will produce a faster response thus enabling the real-time operation of the type-2 FLC. Each of the compared controllers had a rule base of 42 rules with two inputs; Error (Percentage Nominal Error represented by 7 antecedent MF) and the Error Integral (represented by 6 antecedent MF). The outputs of all controllers govern the Manipulating Variable which was used to drive a hydraulic actuator on the engine testing platform.

A. The Control Surface of the T2NFCs

The control surface enables us to visualise the unknown input-output mapping function articulated by the controller which will allow us to analyse the response of each controller. Fig. 4a and Fig. 4b illustrate the control surfaces produced by the T2NFC and the RT2NFC respectively.



(a) T2NFC surface plot



(b) The RT2NFC surface plot

Fig. 4. Surface plots of the type-2 FLCs produced by RT2NFC and T2NFC.

The control surface plots of both the RT2NFC and the T2NFC are noticeably similar and equally quite smooth near the desired point where *error=0*. This means that both type-2 FLCs will react smoothly when the error deviates from zero due to uncertainties associated with change of load and operation conditions. This will be illustrated on the control responses to be shown in the next subsection. From the control surface analysis, it is obvious that the T2NFC and the RT2NFC will generate very similar type-2 FLCs. Note that the distinctive dip in both surface plots where the *Integral of Error =0* is a function of the integrator windup control embedded within the Viking 25 PID controller.

B. The Control Experiments

The experiments shown in this section are a representative subset of an extensive set of experiments we performed all of which repeatedly led to the same results shown below. All the experiments were performed on the engine testing platform presented in Section II. Both the RT2NFC and the T2NFC were coded in ANSI C and embedded in the DSP (discussed in Section II). For the engine testing platform, a set-point of 905 rpm was chosen to correspond with the requirements of medium speed diesel engines. We mimicked the real operation of diesel engines where in each experiment the controllers were allowed to reach the set-point and stabilise with no load, after which we began to add different loads suddenly to mimic the uncertainties associated with change of operation and load conditions. It is necessary for the diesel engine speed controller to be able to deal quickly with the uncertainties associated with a change of load, producing minimum overshoot/undershoot which must be in accordance with the British Standard BS5514 which dictates that the overshoot and undershoot must be within 15% of the nominal set-point and must settle within 1% of the set-point within 4 seconds for up to a 100% load addition [3].

In the representative experiments shown in Fig. 5, the Viking 25 was tuned to handle disturbances that were equivalent to 20% of the full load. The data used to train the RT2NFC and the T2NFC were obtained from the tuned Viking 25.

Fig. 5(a) shows that the performances of both the RT2NFC and the T2NFC are similar to the Viking 25 when introducing the disturbance of 20 % load that they were tuned to handle. However as the uncertainty associated with the change of load increases to 80 % and 100% load as shown in Fig. 5b and Fig. 5c respectively, the performance of the Viking 25 degrades significantly producing large overshoots/undershoots as well as long settling times.

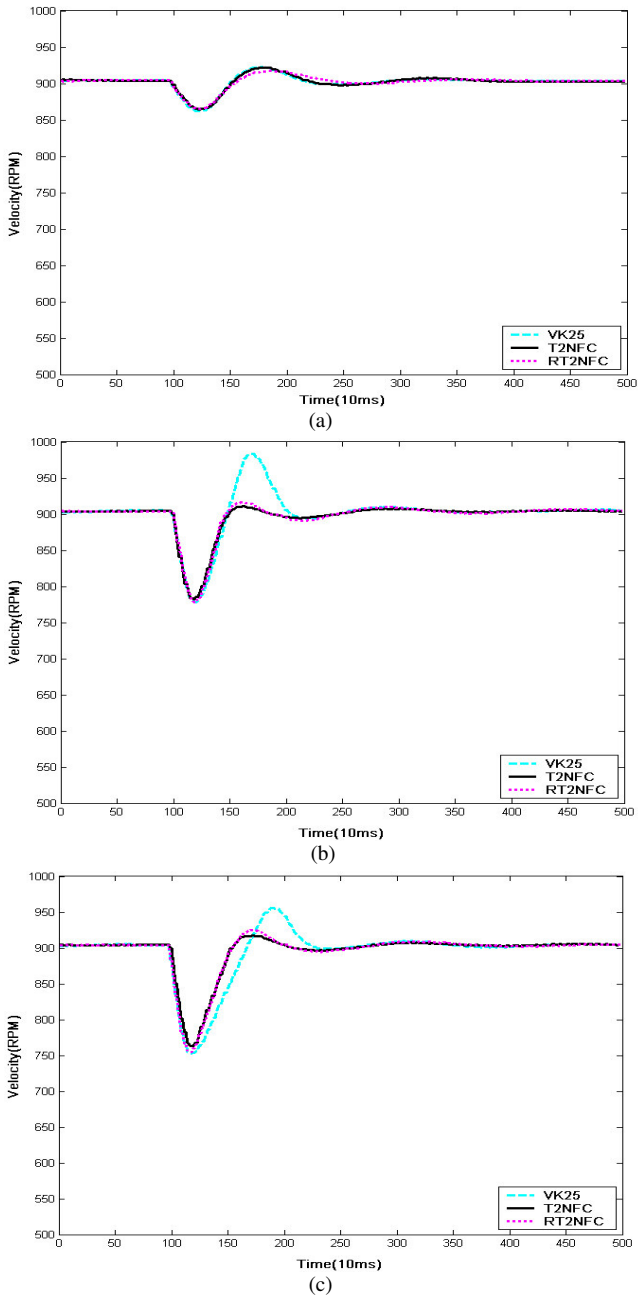


Fig. 5. Control response of the RT2NFC, T2NFC and Viking 25 to the uncertainties associated with load changes of: (a) 20% (b) 80% (c) 100 %

For the Viking 25, the results shown in Fig. 5b, Fig. 5c are unacceptable as they do not satisfy the desired standards, thus the common practice in such situations is to retune the controller which is a time consuming process [1].

On the other hand, both the RT2NFC and the T2NFC produced type-2 FLCs that handled effectively the uncertainties associated with the change of the load and operation condition to give a very good performance that has small overshoots/undershoots as well as short settling times. The performance of both the RT2NFC and the T2NFC satisfy the required standards and thus they will require no further tuning. Therefore the RT2NFC and T2NFC could be used effectively to produce accurate and robust speed controllers for marine diesel engines. This shows the power of type-2 FLCs as although the RT2NFC and T2NFC were trained to mimic the currently used Viking 25 controller, however as the level of uncertainties increase, the type-2 FLCs were able to handle the uncertainties and outperform the Viking controller that was used to train them. This demonstrates the potential type-2 FLCs hold for control applications.

The similarity in control response between the RT2NFC and the T2NFC is a valid indication that they will both produce the same outputs and responses. However, in the next subsection, we will show how the RT2NFC will produce a much faster response (especially as the number of used rules increase) which will enable real-time operations.

C. Computational times of the RT2NFC and the T2NFC

In this subsection, we will introduce a comparison between the computational times of the RT2NFC and the T2NFC. These comparisons were conducted on our electronic embedded controller where both the RT2NFC and T2NFC used the same rule base and MFs. In this comparison, the RT2NFC and the T2NFC were provided with a large set of input data (Error and Error Integral) which covered the entire universe of discourse and we measured the time required to generate crisp outputs using both the RT2NFC (that uses uncertainty bounds) and T2NFC (that uses type-reduction).

Fig. 6 illustrates the mean (solid lines) and standard deviations (dashed upper and lower error bars) of these calculated times (expressed in micro-seconds) for a varying number of fired rules. It is shown that the mean and the standard deviation intervals for the RT2NFC were always less then the T2NFC, for example in the instance of 25 rules firing (as shown in Fig. 6) the RT2NFC achieved a comparative reduction of 28.5% relative to the T2NFC. However, in fuzzy logic systems the precision of the approximation for the control function is directly related to the number of rules and MFs [21]. In marine diesel engines, the customer requested standards can entail increasing the control precision which will imply having more MFs and thus a larger rule base (implying larger number of fired rules). This large number of fired rules will be problematic for the T2NFC which will degrade its real time performance and thus its ability to produce an accurate and robust response especially when operating on an embedded platform. On the other hand, the RT2NFC will produce faster computation times compared to the T2NFC, where for example the RT2NFC is 48.8% and 54% faster than T2NFC

in the instance of 100 and 200 rules firing respectively and these computational savings will increase as the number of firing rules increase. This RT2NFC faster response will map to a better real-time response that will impact not only our ability to meet hard real-time deadlines but also to free the processor for executing other high priority components such as alarms and monitoring whilst still having the superior type-2 FLC control response.

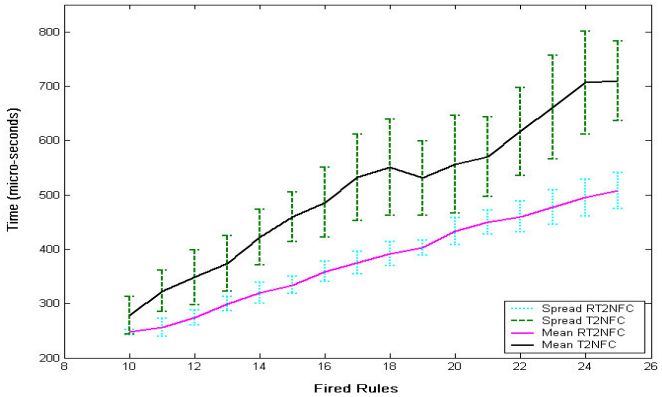


Fig. 6. Mean and Standard Deviation of RT2NFC and the T2NFC

VI. Conclusions

In this paper, we presented an embedded RT2NFC that avoided the computational overhead associated with type-reduction in type-2 FLCs by using the *Wu-Mendel* uncertainty bounds approximation. The RT2NFC used a neural based approach to learn the parameters of interval type-2 FLC for the speed control of marine diesel engines. The structure and equations defining the RT2NFC were discussed and the procedure required for tuning of the real time type-2 FLC was presented.

We have performed numerous experiments on a real diesel engine testing platform in which we compared the currently used Viking 25 commercial controller to our RT2NFC as well to the T2NFC that uses the iterative *Karnik-Mendel* type-reduction procedure. It has been shown that the under no or small levels of uncertainties the RT2NFC and T2NFC gave approximately the same performance as the Viking 25. However, under larger uncertainties both the RT2NFC and T2NFC produced a robust and accurate control response with short settling times and with small overshoots/undershoots thus outperforming the Viking 25. However, the T2NFC will have problems when the number of fired rules increase (which is associated with having a larger number of MFs and rules needed to provide more control precision for the engines) which will reduce the T2NFC real time performance and thus its ability to produce an accurate and robust response especially when operating on an embedded platform. It was shown that the T2NFC and the RT2NFC will generate very similar control responses. However, we have shown that the RT2NFC has significantly faster computational times than the T2NFC especially as the number of rules increase which will result in a better real-time response. Thus the RT2NFC will enable us to fully explore the potential of type-2 FLCs while helping to meet the hard real-time deadlines. We are currently working on further enhancements to the RT2NFC and testing it on real diesel engines to produce the first type-2 commercial real-time controller.

References

[1] Regulateurs Europa. (June 2003). Technical Papers and Data sheets. [Online]. Available: http://www.regulateurseuropa.com

[2] MAN B&W Technical Papers. (April 2004). Basics of Ship Propulsion [Online]. Available: http://www.manbw.com

[3] British Standards. (March 1997). BS 5514 Reciprocating internal combustion engines: Speed Governing (2nd ed.) [Online]. Available: http://www.bsonline.bsi-global.com/server/index.jsp

[4] J.Mendel and R.John, "Type-2 fuzzy sets made simple," *IEEE Trans. on Fuzzy Systems*, vol. 10, pp. 117-127, April 2002.

[5] J.Mendel and H.Wu, "Uncertainty versus choice in rule-based fuzzy logic systems," in *Proc. of IEEE Int. Conf. on Fuzzy Systems*, Honolulu, USA, May 2002, pp. 1336-1342.

[6] A.Amer, S.Eweda and M.Eweda, "Speed control of marine diesel engine using fuzzy approach part(II)", in *Proc. of the 14th Int. Conf. on Computer Theory and Applications*, Alexandria, Egypt, September 2004.

[7] D.Langbridge, S.Cater, and N.Mort, "Experiences with rule-based control algorithms in a teaching laboratory and a diesel engine test cell," in *IEE Colloquium on Exploiting the Knowledge Base: Applications of Rule Based Control,* London, UK, June 1989, pp. 1-5

[8] H.Hagras, "A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots," *IEEE Trans. on Fuzzy Systems*, vol. 12, no.4, pp. 524-539, August 2004.

[9] C.Lynch, H.Hagras, and V.Callaghan, "Embedded Type-2 FLC for Real-Time Speed Control of Marine & Traction Diesel Engines," in *Proc. of the IEEE Int. Conf. on Fuzzy Systems*, Reno, USA, May 2005, pp. 347-352.

[10] D.Wu and W. Tan, "A type-2 fuzzy logic controller for the liquid level process," in *Proc. of the 2004 IEEE Int. Conf. on Fuzzy Systems*, Budapest, Hungary, July 2004, vol. 2, pp. 953-958

[11] W.Tan and J.Lai, "Development of a type-2 fuzzy proportional controller," in *Proc. of the 2004 IEEE Int. Conf. on Fuzzy Systems*, Budapest, Hungary, July 2004, vol. 3, pp. 1305-1310

[12] P. Melin and O. Castillo, "A New Method for Adaptive Control of Non-Linear Plants using Type-2 Fuzzy Logic and Neural Networks," *Int. J. of General Systems*, vol. 33, pp. 289-304, June 2004.

[13] Q.Liang and J.Mendel, "Interval type-2 fuzzy logic systems: theory and design," *IEEE Trans. on Fuzzy Systems*, vol. 8, pp. 535-550, October 2000.

[14] J.Mendel, "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New directions," Upper Saddle River, NJ: Prentice-Hall, 2001.

[15] H. Wu and J. Mendel, "Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems," *IEEE Trans. on Fuzzy Systems*, vol.10, pp. 622-639, October 2002.

[16] C. Lee, J. Hong, Y. Lin, and W. Lai, "Type-2 Fuzzy Neural Network Systems and Learning," *Int. J. of Computational Cognition*, vol. 1, pp. 79-90, December 2003.

[17] C.Wang, C.Cheng, and T.Lee, "Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network (T2FNN)," *IEEE Trans. on Systems, Man and Cybernetics Part B: Cybernetics*, vol. 34, no. 3, pp. 1462-1477, June 2004.

[18] R.John and C.Czarnecki, "An Adaptive Type-2 Fuzzy System For Learning Linguistic Membership Grades," in *Proc. of the IEEE Int. Conf. on Fuzzy Systems*, Seoul, Korea, August 1999, vol. 3, pp. 1552-1556

[19] S.Haykin, "Neural Networks: comprehensive Foundation," 2nd ed., Prentice Hall, 1999.

[20] C. Lynch, H. Hagras and V. Callaghan, "Embedded Interval Type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines," in *Proc. of the IPMU 2006*, Paris, France, 2006, to be published

[21] A. Kandel and G. Langholz, "Fuzzy Control Systems," CRC Press, Sept 1993