Embedded Interval Type-2 Neuro-Fuzzy Speed Controller for Marine Diesel Engines

Christopher Lynch **Hani Hagras** **Victor Callaghan**

Department of Computer Science, University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, UK
Email: {clynch, hani, vic}@essex.ac.uk

Abstract

Marine diesel engines operate in highly dynamic and uncertain environments, hence they require robust and accurate speed controllers that can handle the uncertainties encountered in these environments. The current speed controllers for marine diesel engines are based on PID and type-1 Fuzzy Logic Controllers (FLCs) which cannot fully handle the uncertainties encountered in such environments. Type-2 FLCs can handle such uncertainties to produce a better control performance. However, manually designing a type-2 FLC is a difficult task. In this paper, we will introduce an embedded type-2 Neuro-Fuzzy Controller (T2NFC) which learns the parameters of interval type-2 FLC to control marine diesel engines. We have performed numerous experiments on a real diesel engine testing platform in which the T2NFC operated on an industrial embedded controller and handled the uncertainties to produce an accurate and robust speed controller that outperformed the currently used commercial engine controller, even though we have trained the T2NFC with data collected from the commercial controller.

Keywords: Interval type-2 FLC, Neuro-Fuzzy Systems

1 Introduction

Marine diesel engines are huge engines that are used in large container ships, high speed ferries, harbour tugs, offshore patrol crafts, etc. The engines vary in power and speed from a few hundred rpm and 90,000 kW to several thousand rpm and less than 10,000 kW. Figure 1 shows two of these huge engines namely the Wartsila-Sulzer RTA96-C and the MAN B&W RK 270.

Due to their vast sizes and large power outputs, marine diesel engines require *accurate* and *robust* speed control/governing. *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. Moreover, for applications such as power generation sets, significant speed deviation can cause the generation of incorrect frequencies resulting in loss of synchronisation between the generator and the associated power grid. *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. 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.

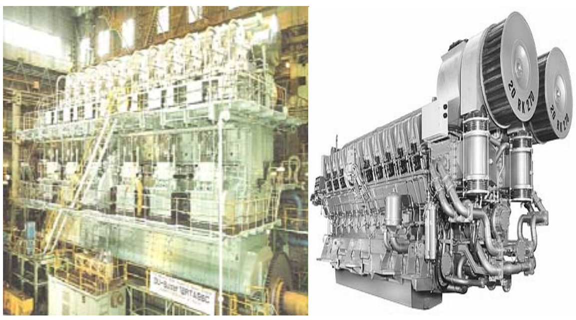


Figure 1: Examples of marine diesel engines: RTA96-C (left) and the RK270 (right).

Due to their simplicity and their suitability for the industrial embedded controllers, various forms of the PID controller have been used for speed control in marine diesel engines. It has been shown that FLCs and Fuzzy PID controllers can provide improved control over traditional PID controllers [1], [2]. As a result, FLCs have found use in the speed control of various marine diesel engines [1], [6]. However, there are many sources of uncertainty facing the FLC for the speed control of marine diesel engines; we list some of them as follows:

- Uncertainties in inputs to the FLC which translate to uncertainties in the antecedents Membership Functions (MFs) as sensors measurements are affected by high noise levels and their characteristics also change due to the diverse environmental conditions facing the engines [9].
- Uncertainties in control outputs which translate to uncertainties in the output MFs of the FLC. Such uncertainties can result from change of the actuators characteristics due to wear and tear or environmental changes [9].
- Linguistic uncertainties as the meaning of words that are used in the antecedents and consequents linguistic labels can be uncertain, as words mean different things to different people [12]. In addition, experts do not always agree and they often provide different consequents for the same antecedents. [13].
- Uncertainties associated with the change in engine operation and load conditions due to varying loads, weather and sea conditions, wind strength, hull fouling (growth of algae, sea grass and barnacles) and vessel displacement which is dependant on cargo. These uncertainties are considered the most dynamic and severe uncertainties that can affect both the input and output of the FLC to cause serious degradation in the performance of the engine [10]. 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 [10].

The above uncertainties can translate to uncertainties in the antecedents and/or consequents MFs [12]. 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* MFs that might be only valid under specific conditions [3], [11]. However, type-1 FLCs cannot fully handle or accommodate the high levels of linguistic and numerical uncertainties present in changing and dynamic environments such as the marine diesel engines environments [9]. Alternatively type-2 FLCs that use type-2 fuzzy sets have been proven to directly model and handle these uncertainties [9]. However, manually designing a type-2 FLC is a difficult task, particularly as the number of MFs and rules increase. In this paper, we will introduce a Type-2 Neuro-Fuzzy Controller (T2NFC) which learns the parameters of interval type-2 FLC to control marine diesel engines.

There has been other work that used neural based systems to learn the parameters of type-2 FLCs that were Mamdani based like [7], [14] or TSK based like [5]. However, according to the author's knowledge, this is the first work that develops on an embedded industrial controller a type-2 Neuro fuzzy system that is trained from the commercial existing controller to learn the parameters of an interval type-2 FLC for the real-time control of a heavy industrial application like marine diesel engines.

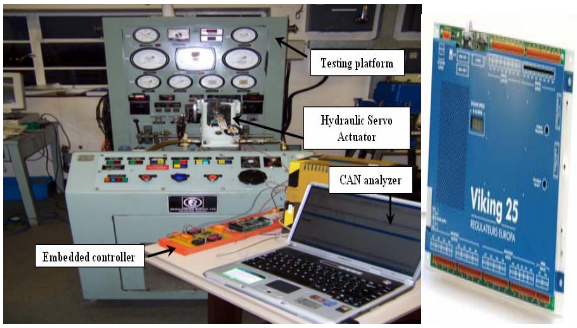
The remainder of this paper is organised as follow: In section 2, we will introduce the testing platform for marine engines. Section 3 will briefly review the interval type-2 fuzzy sets. Section 4 will present the T2NFC. Section 5 will present our experiments and results followed by the conclusions in section 6.

2 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 Figure 2(a) 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 in the engine with the ability to introduce the same uncertainty levels faced in the real engines.

The T2NFC was embedded on the 32-bit Texas Instruments TMS320F2812 150MHz industrial DSP which was programmed in ANSI C and Assembly. The embedded T2NFC controls the speed of the engine by controlling a hydraulic servo actuator which manages the rate of fuel delivery to the cylinders. 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 controller (shown in Figure 2(b)) 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.



(a) (b)

Figure 2: (a) The testing platform (b) Viking 25.

3 Interval Type-2 Fuzzy sets

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

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

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 shown in Figure 4(a) 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:

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

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

Where

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

When an input, x_k is located in a specific x-domain segment, we call its corresponding MF an *active branch* [11]. The MF is piece-wise differentiable i.e. each branch is differentiable over its segment domain [11]. 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:

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

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

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

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

4 The Type-2 Neuro-Fuzzy Controller

4.1 The T2NFC Structure

Figure 3 shows the structure of the proposed T2NFC which mimics the well known interval type-2 FLC reported in [3],[8],[11]. The T2NFC architecture is inspired from the T2FNN in [14], however the T2NFC is targeted mainly for control applications and in this paper we will address the correct equations and operations which were not correctly presented in [14].

In the T2NFC, Layer I 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 ($i=1 \ldots M$, 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* $[w_{lc}^i, w_{rc}^i]$ (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 [8], [11]:

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

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

Where * denotes the product t-norm. Layer IV is the *Type Reduction Layer* which takes us from the type-2 output sets of the inference engine to a type-1 set called the "type reduced set". We will use the "centre of sets" type reduction, as it has reasonable computational complexity that lies between the computationally expensive centroid type reduction and the simple height and modified height type reductions which have problems when only one rule fires [11]. The type reduced set using the centre of sets type reduction is expressed as an interval set defined by its left and right most points y_l and y_r which can be written as follows:

$$y_l = \frac{\sum_{i=1}^{L} \overline{f}^i w_l^i + \sum_{i=L+1}^{M} \underline{f}^i w_l^i}{\sum_{i=1}^{L} \overline{f}^i + \sum_{i=L+1}^{M} \underline{f}^i} \quad (12)$$

$$y_r = \frac{\sum_{i=1}^{R} \underline{f}^i w_r^i + \sum_{i=R+1}^{M} \overline{f}^i w_r^i}{\sum_{i=1}^{R} \underline{f}^i + \sum_{i=R+1}^{M} \overline{f}^i} \quad (13)$$

It is assumed that w_l^i, w_r^i are arranged in an ascending order such that $w_l^1 \le w_l^2 \ldots.. w_l^M$ and $w_r^1 \le w_r^2 \ldots.. w_r^M$. Where L and R could be obtained from the iterative Karnik-Mendel procedure [8], [11]. Layer V is the *Defuzzification Layer* which calculates the T2NFC crisp outputs by taking the average of y_l and y_r for each given output as follows:

$$y(\vec{x}) = \frac{y_l + y_r}{2} \quad (14)$$

The T2NFC operates as an interval type-2 FLC in the forward mode mapping crisp inputs to crisp outputs. In the next subsection, we will introduce the T2NFC backward mode which is used to learn the parameters of the interval type-2 FLC for the speed control of marine engines.

4.2 Learning the type-2 FLC Parameters

In order to make sure that the type-2 FLC learnt by the T2NFC is suitable for the speed control of marine diesel engines, we have to ensure that the its performance will be at least as good as the currently used Viking 25 commercial controller. Hence, the T2NFC will be trained with input/output data captured from the Viking 25 so that a satisfactorily trained T2NFC will produce a type-2 FLC that will have the same performance as the Viking 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 to give a very good performance as will be demonstrated in the experiments section.

For the i^{th} rule, the T2NFC learns the needed 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:

$$e^p = \frac{1}{2}\left[y(\vec{x}^p) - d^p\right]^2 \qquad p = 1,\ldots. P \quad (15)$$

By using α to represent the learning rate for the BP tuning then m_{k1}^i can be tuned as follows:

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

$$= m_{k1}^i(p) - \alpha\left[\frac{\partial e^p}{\partial y(\vec{x})}\frac{\partial y(\vec{x})}{\partial y_l}\frac{\partial y_l}{\partial m_{k1}^i} + \frac{\partial e^p}{\partial y(\vec{x})}\frac{\partial y(\vec{x})}{\partial y_r}\frac{\partial y_r}{\partial m_{k1}^i}\right]_p \quad (16)$$

Where:

$$\left.\frac{\partial e^p}{\partial y(\vec{x})}\right|_p = y(\vec{x}^p) - d^p \quad (17)$$

$$\left.\frac{\partial y(\vec{x})}{\partial y_l}\right|_p = \frac{1}{2} \quad (18)$$

$$\left.\frac{\partial y(\vec{x})}{\partial y_r}\right|_p = \frac{1}{2} \quad (19)$$

$$\left.\frac{\partial y_l}{\partial m_{k1}^i}\right|_p = \left[\frac{\partial y_l}{\partial \overline{\mu}_{F_k^i}(x_k)}\frac{\partial \overline{\mu}_{F_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial y_l}{\partial \underline{\mu}_{F_k^i}(x_k)}\frac{\partial \underline{\mu}_{F_k^i}(x_k)}{\partial m_{k1}^i}\right] \quad (20)$$

$$\left.\frac{\partial y_r}{\partial m_{k1}^i}\right|_p = \left[\frac{\partial y_r}{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)}\frac{\partial \overline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i} + \frac{\partial y_r}{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)}\frac{\partial \underline{\mu}_{\tilde{F}_k^i}(x_k)}{\partial m_{k1}^i}\right] \quad (21)$$

$$\frac{\partial y_l}{\partial \overline{\mu}_{F_k^i}(x_k)} =$$

$$\frac{\left(\sum_{i=1}^{L}\overline{f}^i + \sum_{i=L+1}^{M}\underline{f}^i\right)\left(w_l^i \prod_{j=1, j\neq k}^{n} \overline{\mu}_{\tilde{F}_j^i}\right) - \left(\sum_{i=1}^{L}\overline{f}^i w_l^i + \sum_{i=L+1}^{M}\underline{f}^i w_l^i\right)\left(\prod_{j=1, j\neq k}^{n} \overline{\mu}_{\tilde{F}_j^i}\right)}{\left(\sum_{i=1}^{L}\overline{f}^i + \sum_{i=L+1}^{M}\underline{f}^i\right)^2} \qquad i \le L$$

$$=0 \qquad i > L \quad (22)$$

i.e.
$$\frac{\partial y_l}{\partial \overline{\mu}_{F_k^i}(x_k)} = \begin{cases} \dfrac{\left(\prod_{j=1, j\neq k}^{n} \overline{\mu}_{\tilde{F}_j^i}\right)\left(w_l^i - y_l\right)}{\left(\sum_{i=1}^{L} \overline{f}^i + \sum_{i=L+1}^{M} \underline{f}^i\right)} & i \leq L \\ = 0 & i > L \end{cases} \quad (23)$$

Similarly:

$$\frac{\partial y_l}{\partial \underline{\mu}_{F_k^i}(x_k)} = \begin{cases} \dfrac{\left(\prod_{j=1, j\neq k}^{n} \underline{\mu}_{\tilde{F}_j^i}\right)\left(w_l^i - y_l\right)}{\left(\sum_{i=1}^{L} \overline{f}^i + \sum_{i=L+1}^{M} \underline{f}^i\right)} & i > L \\ = 0 & i \leq L \end{cases} \quad (24)$$

$$\frac{\partial y_r}{\partial \overline{\mu}_{F_k^i}(x_k)} = \begin{cases} \dfrac{\left(\prod_{j=1, j\neq k}^{n} \overline{\mu}_{\tilde{F}_j^i}\right)\left(w_r^i - y_r\right)}{\left(\sum_{i=1}^{R} \underline{f}^i + \sum_{i=R+1}^{M} \overline{f}^i\right)} & i > R \\ = 0 & i \leq R \end{cases} \quad (25)$$

$$\frac{\partial y_r}{\partial \underline{\mu}_{F_k^i}(x_k)} = \begin{cases} \dfrac{\left(\prod_{j=1, j\neq k}^{n} \underline{\mu}_{\tilde{F}_j^i}\right)\left(w_r^i - y_r\right)}{\left(\sum_{i=1}^{R} \underline{f}^i + \sum_{i=R+1}^{M} \overline{f}^i\right)} & i \leq R \\ = 0 & i > R \end{cases} \quad (26)$$

From the above equations, it is clear that the tuning process depends on the active branch that x_k^p belongs to and the situation of the rule i compared to L and R. So for example if If $i \leq L$ and $i \leq R$ and $x_k^p < m_{k1}^i$ (hence $x_k^p < \left(m_{k1}^i + m_{k2}^i\right)/2$), therefore substituting from (23),(5),(24),(6) into (20) (we will call this (20')) furthermore substituting by (25), (5), (26), (6) into (21) ((we will call this (21')) and then substituting by (17), (18), (20'), (19), (21') into (16) we get the following

$$m_{k1}^i(p+1) = m_{k1}^i(p) - \frac{1}{2}\alpha\left(y(\vec{x}^p) - d^p\right)$$
$$\left(\frac{\left(\prod_{j=1, j\neq k}^{n} \overline{\mu}_{\tilde{F}_j^i}\right)\left(w_l^i - y_l\right)}{\left(\sum_{i=1}^{L} \overline{f}^i + \sum_{i=L+1}^{M} \underline{f}^i\right)} (x_k^p - m_{k1}^i) N(m_{k1}^i, \sigma_{k2}^i; x_k^p) / {\sigma_{k2}^i}^2\right) \quad (27)$$

By following a similar procedure as above we can derive the BP update equation for m_{k2}^i. In the same manner the $\sigma_{k1}^i, \sigma_{k2}^i$ can be tuned, so for example if $x_k^p < m_{k1}^i$, $i \leq L$ and $i \leq R$, then σ_{k1}^i can be tuned as follows:

$$\sigma_{k1}^i(p+1) = \sigma_{k1}^i(p) - \frac{1}{2}\alpha\left(y(\vec{x}^p) - d^p\right)$$
$$\left(\frac{\left(\prod_{j=1, j\neq k}^{n} \underline{\mu}_{\tilde{F}_j^i}\right)\left(w_r^i - y_r\right)}{\left(\sum_{i=1}^{R} \underline{f}^i + \sum_{i=R+1}^{M} \overline{f}^i\right)} (x_k^p - m_{k2}^i)^2 N(m_{k2}^i, \sigma_{k1}^i; x_k^p) / {\sigma_{k1}^i}^3\right) \quad (28)$$

Where in (27) we must take care to use $w_l^1 \leq w_l^2 ..... w_l^M$ and in (28) use $w_r^1 \leq w_r^2 ..... w_r^M$. The type-2 FLC consequent parameters can be tuned in the same way by following the same procedure. So for example if $i \leq L$, w_l^i can be tuned as follows:

$$w_l^i(p+1) = w_l^i(p) - \frac{1}{2}\alpha\left(\left(y(\vec{x}^p) - d^p\right)\frac{\left(\prod_{j=1}^{n} \overline{\mu}_{\tilde{F}_j^i}\right)}{\sum_{i=1}^{L} \overline{f}^i + \sum_{i=L+1}^{M} \underline{f}^i}\right) \quad (29)$$

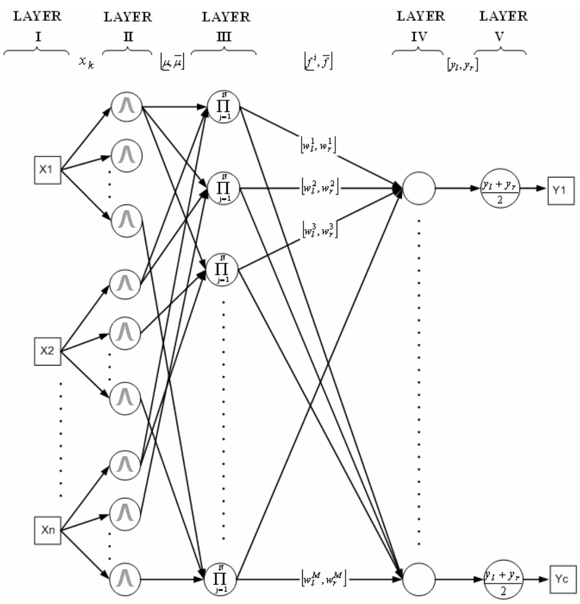
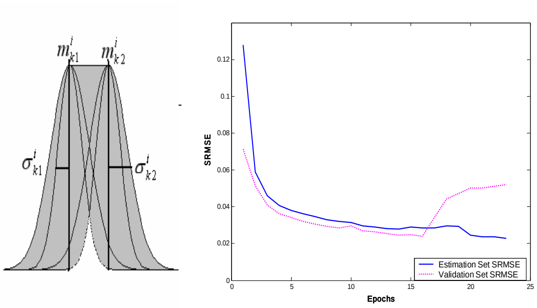


Figure 3: The Type-2 Neuro-Fuzzy Controller

As it is possible to over-fit the training data (10,000 samples) by monitoring of the Scaled Root Mean Squared Error (SRMSE) alone, we have split our training data into an estimation subset (6000 samples) and a validation subset (4000 samples). The estimation subset of examples is used to train the T2NFC in the usual way, except for a minor modification: the training session is stopped periodically (i.e. every so many epochs), and the network is tested on the validation subset [4]. The estimation learning curve decreases monotonically for each successive epoch, in contrast the validation learning curve decreases monotonically to a minimum at which point it start to increase as the training continues [4]. Training beyond this minimum point is essentially learning from noise contained within the training data [4]; thus stopping the training at this point will allow us to avoid over-fitting the training data. Therefore, we stop training the T2NFC at this minimum point to produce a type-2 FLC that can be used in real-time control. Figure 4(b) illustrates how the validation subset can be used as an indication to stop training where the training SRMSE decreases for each successive epoch whilst the validation SRMSE decreases until a minimum (epoch 16) at which point it then begins to increase for each successive epoch. Training was stopped at this point and the T2NFC operated in the forward mode to control the speed of the engine.

From the above discussion it is obvious that the learning of the type-2 FLC parameters within the T2NFC besides requiring to find the active branches of the MFs, it also necessitates to keep in memory track of L, R and the sorted consequents centroid list (such that $w_l^1 \le w_l^2 ..... w_l^M$ and $w_r^1 \le w_r^2 ..... w_r^M$) that were computed and used in the forward pass. These requirements were not considered in [14], in addition the BP tuning equations quoted in [14] were also erroneous.



(a) (b)

Figure 4: (a) Interval type-2 fuzzy which is represented by gaussian primary MF having uncertain standard deviation and uncertain mean. (b) The use of the validation set to stop the T2NFC training

5 Experiments and Results

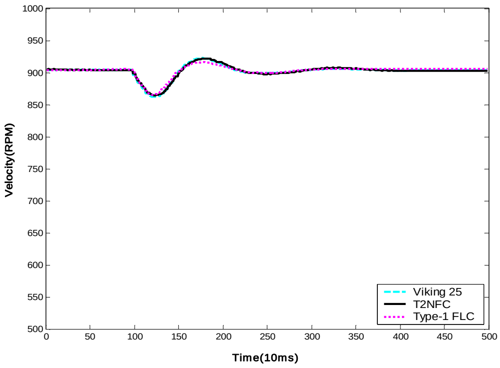
In this section, we will compare the T2NFC performance against the Viking 25 commercial controller as well as an expert designed type-1 FLC in handling the uncertainties and disturbances that are associated with the change of operation and load conditions.

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 controlled the Manipulating Variable (MV) which was used to drive a hydraulic actuator on the engine testing platform.

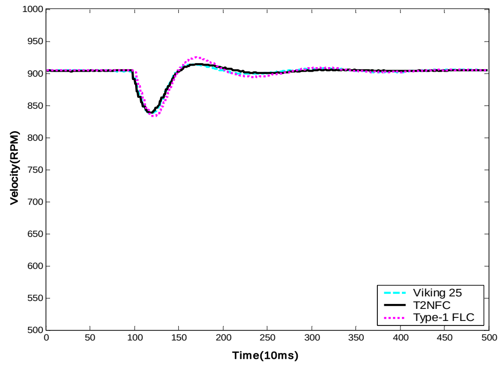
The experiments shown in this section are a representative subset of an extensive set of experiments we performed all of which repeatedly led to the same results shown below. All the experiments were performed on the engine testing platform presented in Section (2). Both the T2NFC and the type-1 FLC were coded in ANSI C and embedded in the DSP (discussed in Section (2)). For the engine testing platform, a set-point of 905 rpm was chosen to correspond with the requirements of medium speed diesel engines. We used the same noisy sensors and actuators utilised on the real engine and we introduced the same uncertainty levels faced in the real 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 $\pm$ 15% of the nominal set-point and must settle within $\pm$ 1% of the set-point within 4 seconds for up to a 100% load addition.

In the representative experiment shown in Figure 5, both the Viking 25 and type-1 FLC were tuned so that they can handle disturbances that were equivalent to 20% of the full load (which is common disturbance that can face the engines at normal sea condition). The data used by the T2NFC was obtained from the Viking 25.

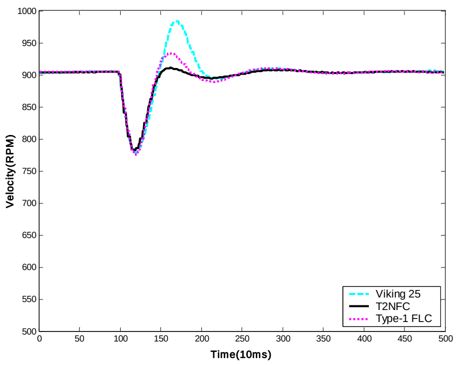
Figure 5(a) shows that the performance of the T2NFC is similar to the Viking 25 and type-1 FLC when introducing the disturbance of 20 % load that they were tuned to handle. However as the uncertainty associated with the change of load increases to 40 % , 80 % and 100% load as shown in Figure 5(b), Figure 5(c), Figure 5(d) respectively, the performance of both the Viking 25 and type-1 FLC degrades significantly producing large overshoots/undershoots as well as long settling times.



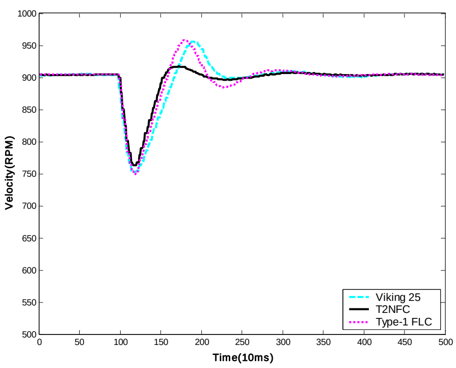
(a)



(b)



(c)



(d)

Figure 5: Experiments showing the response of the T2NFC, the Viking 25 and the type-1 FLC to load uncertainties associated with load changes of: (a) 20 % (b) 40 % (c) 80 % (d) 100.

For the Viking 25 and the type-1 FLC, the results shown in Figure 5(c) and Figure (d) 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,

On the other hand, the T2NFC produces a type-2 FLC that handles 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 the T2NFC satisfies the required standards and thus it will require no further tuning. Therefore the T2NFC could be used effectively to produce accurate and robust speed controller for marine diesel engines.

6 Conclusions

In this paper, we presented an embedded T2NFC for the speed control of marine diesel engines. The T2NFC was implemented on an industrial electronic embedded controller and was evaluated on a real diesel engine testing platform. We presented the structure of the T2NFC and the procedure it uses to learn parameters of the real-time type-2 FLC. We have shown that the BP tuning of the T2NFC is rather complicated and involves keeping track of the active branches of the MFs as well as keeping track of L, R and the sorted consequents centroid list that were computed and used in the forward pass. These requirements were not considered in [14], in addition the BP tuning equations quoted in [14] were also erroneous.

The T2NFC was trained by data collected from the commercial Viking 25 controller that is currently used in real diesel engines. Through the experiments that we conducted, we have shown the following:

- As the uncertainties increase the type-1 FLC and Viking 25 performance deteriorates to produce long settling times and large overshoots/undershoots that violates the given standards requiring them to undergo a time consuming retuning cycle.
- The T2NFC will give the same performance as the Viking 25 and type-1 FLC under no or small level of uncertainties.
- The T2NFC will handle the uncertainties to produce a robust and accurate control response that can recover quickly from disturbances with short settling times and with small

overshoots/undershoots thus outperforming the performance of the type-1 FLC and Viking 25.

- The T2NFC met the requirements of the British Standard BS5514 and did not require any further tuning making it an excellent option to effectively produce accurate and robust speed controllers for marine diesel engines.

This paper has shown the potential type-2 FLCs have for control applications in general where they can be learnt and tuned from data supplied by the existing controllers. However, the type-2 FLCs ability to handle uncertainties will enable them to give a more accurate and robust control performance than the existing controllers.

We hope that this paper will be a step towards the simplification of the type-2 FLC design process and hence its further deployment on embedded platforms in many more control applications. We are currently working on reducing the computational complexity of the T2NFC to allow for a faster response and hence its deployment on FPGA platforms to produce the first type-2 commercial control product.

Acknowledgements

We are pleased to acknowledge the funding support from MAN B&W Diesel Ltd. We are also pleased to acknowledge the support of all the staff at Regulateurs Europa, particularly Kevin Murray, Martin Birkin, Geoff Nunn and Chris Tupe. Finally we acknowledge our colleagues at the IIE research group.

References

[1] A.Amer, S.Eweda and M.Eweda, "Speed control of marine diesel engine using fuzzy approach part(II)", Proceedings of the 14th International Conference on Computer Theory and Applications, Alexandria, Egypt, Sept 2004.

[2] M.Golob and B.Tovornik, "Real time fuzzy PID controller structures", Computational Intelligence and Applications, World Scientific and Engineering Society Press, pages 205-210, USA, July 1999.

[3] H.Hagras, “A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots”, IEEE Transactions On Fuzzy Systems, volume 12, number 4, pages 524-539, August 2004.

[4] S.Haykin, "Neural Networks: A comprehensive Foundation", Prentice Hall, 1999.

[5] R.John and C.Czarnecki, "An Adaptive Type-2 Fuzzy System For Learning Linguistic Membership Grades", Proceedings of the IEEE International Conference on Fuzzy Systems, volume 3, pages 1552-1556, Seoul, Korea, August 1999.

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

[7] C. Lee, J. Hong, Y. Lin, and W. Lai, "Type-2 Fuzzy Neural Network Systems and Learning", International Journal of Computational Cognition, volume 1, pages 79-90, December 2003.

[8] Q.Liang and J.Mendel, "Interval type-2 fuzzy logic systems: theory and design", IEEE Transactions on Fuzzy Systems, volume 8, pages 535-550, October 2000.

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

[10] MAN B&W Technical Papers, "Basics of Ship Propulsion", [online], http://www.manbw.com/files/news/filesof3859/P254-04-04.pdf , April 2004.

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

[12] J.Mendel and R.John, "Type-2 fuzzy sets made simple", IEEE Transactions on Fuzzy Systems, volume 10, pages 117-127, April 2002.

[13] J.Mendel and H.Wu, "Uncertainty versus choice in rule-based fuzzy logic systems", Proceedings of IEEE International Conference on Fuzzy Systems, pages 1336-1342, Honolulu, USA, May 2002.

[14] C.Wang, C.Cheng, and T.Lee, "Dynamical Optimal Training for Interval Type-2 Fuzzy Neural Network (T2FNN)", IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics, volume 34, number 3, pages 1462-1477, June 2004.