18

An Embedded Type-2 Fuzzy Controller for a Mobile Robot Application

Leonardo Leottau[1] and Miguel Melgarejo[2]
[1]*Advanced Mining Technology Center (AMTC),*
Department of Electrical Engineering, University of Chile, Santiago
[2]*Laboratory for Automation, Microelectronics and Computational Intelligence (LAMIC),*
Faculty of Engineering, Universidad Distrital Francisco José de Caldas, Bogotá D.C.
[1]*Chile*
[2]*Colombia*

1. Introduction

Fuzzy logic systems (FLS) have been used in different applications with satisfactory performance (Wang, 1997). The human perception cannot be modelled by traditional mathematical techniques, thus, the introduction of fuzzy set (FS) theory in this modelling has been suitable (John & Coupland, 2007). When real-world applications are treated, many sources of uncertainty often appear. Several natures of uncertainties would influence the performance of a system. It is independent from what kind of methodology is used to handle it (Mendel, 2001).

Type-1 Fuzzy logic systems (T1-FLS) have limited capabilities to directly handle data uncertainties (Mendel, 2007). Once a type-1 membership function (MF) has been defined, uncertainty disappears because a T1-MF is totally precise (Hagras, 2007). Type-2 fuzzy logic systems (T2-FLS) make possible to model and handle uncertainties. These are rule based systems in which linguistic variables are described by means of Type-2 fuzzy sets (T2-FSs) that include a footprint of uncertainty (FOU) (Mendel, 2001). It provides a measure of dispersion to capture more about uncertainties (Mendel, 2007). While T2-FSs have non-crisp MFs, T1-FSs have crisp membership grades (MGs) (John & Coupland, 2007).

A representation of the inference model for T2-FLS is depicted in Figure 1 (Mendel, 2007). It begins with fuzzification, which maps crisp points into T2-FSs. Next, inference engine computes the rule base by making logical combinations of antecedent T2-FS, whose results are implicated with consequent T2-FS to form an aggregated output type-2 fuzzy set. Afterwards, Type-Reduction (TR) takes all output sets and performs a centroid calculation of this combined type-2 fuzzy set, which leads to a type-1 fuzzy set called *type-reduced set*. That reduced set is finally defuzzyfied in order to obtain a crisp output (Mendel, 2001; Karnik & Mendel 2001). The computational complexity of this model is reduced if interval type-2 fuzzy sets are used (Mendel, 2001), it is convenient in the context of hardware implementation in order to make softer the computational effort and sped up the inference time.

Type-2 fuzzy hardware is a topic of special interest, since the application of T2-FLS to particular fields that demand mobile electronic solutions would be necessary. Some recent

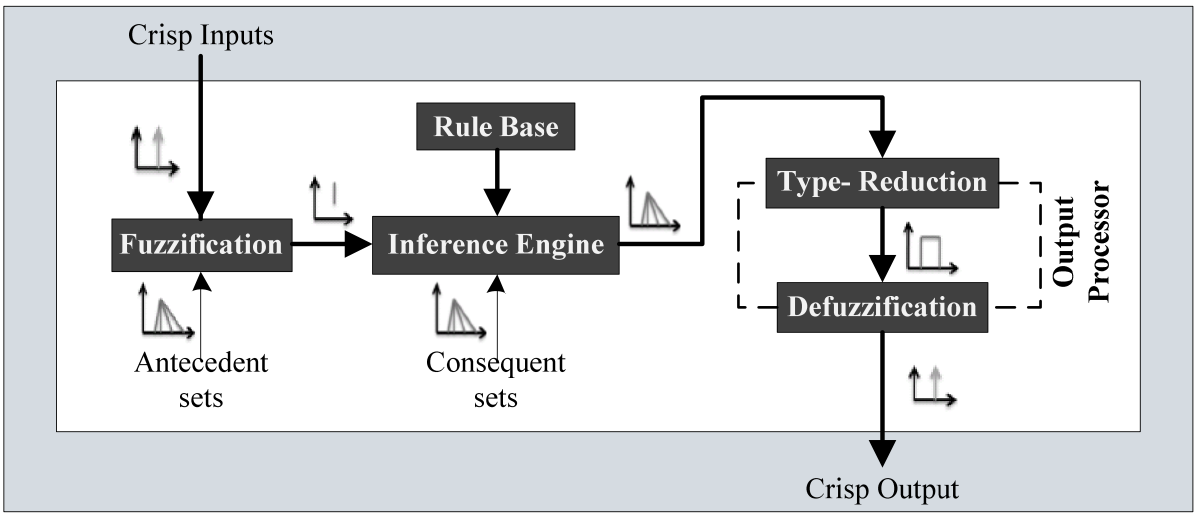


Fig. 1. Type-2 fuzzy system.

applications of T2-FLS have been developed in fields like robotics, communication and control systems among others (Castro et al., 2007; Hagras, 2007; Melgarejo & Peña, 2004, 2007; Torres & Saez, 2008). It is worth to think about the possibility of embedding T2-FLS handling these applications in order to achieve better communication speeds in smaller areas (Leottau & Melgarejo, 2010a).

Control and robotics are one of the most widely used application fields of fuzzy logic. The advantages of type-2 fuzzy logic controllers (T2-FLCs) over type-1 fuzzy logic controllers (T1-FLC) have also been demonstrated and documented in (Castro et al., 2007; Figueroa et al., 2005; Hagras 2004, 2007; Martinez et al., 2008; Torres & Saez, 2008). Although this kind of works presents improvements in the T2-FLCs performance, it is necessary to propose methodologies where these advances can be reflected in design processes as (Hagras 2008, Leottau & Melgarejo, 2010b; Melin & Castillo, 2003; Wu & Tan, 2004) and hardware implementation approaches over embedded devices as (Hagras, 2008; Leottau & Melgarejo, 2010a; Melgarejo & Peña, 2004, 2007). In this way T2-FLS would become widely used in different applicative contexts.

This work presents the implementation of an Interval Type-2 Fuzzy Logic Controller (IT2-FLC) for tracking the trajectory of a mobile robot application. The FLC is design according to the approach proposed in (Leottau & Melgarejo, 2010b), which involves some of the T1-FLC and T2-FLC properties. A hardware implementation of the designed T2-FLC is carried out over a digital signal controller (DSC) embedded platform. Different tests are performed for evaluating the performance of the IT2-FLC with respect to a T1-FLC. Simulation and emulation (i.e. with the embedded controller) results evidence that the IT2-FLC is robust to type reducer changes and exhibits better performance than its T1-FLC counterpart when noise is added to inputs and outputs. The IT2-FLC outperforms the T1-FLC in all tested cases, taking the interval time square error (ITSE) as performance index.

The chapter is organized as follows: Section 2 presents an overview of hardware implementation of IT2-FLCs. Section 3 describes the design approach followed for developing the IT2-FLC and section 4 describes its hardware implementation. Section 5 presents test, obtained results and discussion. Finally, conclusions and future work are presented in Section 6.

2. Hardware Implementation of an IT2-FLC for robot mobile applications

A FLC is a control strategy whose decisions are made by using a fuzzy inference system. Particularly, an IT2-FLC uses IT2-FSs to represent the inputs and/or outputs of the T2-FLS and these sets are completely characterized by their Footprints of Uncertainty (Mendel, 2001). So, this kind of FLC can model the uncertainty by means of the FOUs. In this way, uncertainties in sensors, actuators, operational changes, environmental changes, noise can be considered (Hagras, 2007; Mendel 2001). A FOU is described by its upper and lower MFs. Each MF is composed by as many membership grades (MGs) as discretization levels are considered in antecedents and consequents universes of discourse (Baturone et al., 2000).

This work considers two alternatives for computing the MGs of IT2-FSs: memory based approach and function computing method, which are involved in fuzzyfier and inference engine stages. These two alternatives are the most commonly used in fuzzy hardware implementations (Baturone et al., 2000). Moreover, centroids and center of sets methods are considered for TR stage. As TR algorithm, the Enhanced Karnik-Mendel algorithm is used (Wu & Mendel, 2009). For the sake of clarity, figure 1 depicts the stage structure of an IT2-FLC. These mentioned methods and alternatives are briefly introduced below:

Function Computing Approach (FCA):

This method carries out a direct computation of MF by using numeric algorithms that avoid constructing and storing look up tables (Leottau & Melgarejo, 2010a). It reduces memory usage and facilitates the implementation of MF. However, its execution could require several machine cycles depending on the complexity of MF (Baturone et al., 2000).

Memory Based Approach (MBA):

This approach stores the MGs of every input value into a memory. This strategy is executed considerably fast, because it uses the input value as the pointer to the memory and to directly retrieve the MG (Leottau & Melgarejo, 2010a).

Centroid Type-Reduction (Cent):

The centroid TR combines all the rule-output T2-FSs using union and then finds the centroid of this T2-FS (Karnik & Mendel, 2001). Therefore, this method presents high accuracy but low computational performance (Mendel, 2001). Cent TR is chosen in order to obtain the best accurate output while the hardware platform is forced to the highest computational effort.

Center-of-sets type-reduction (CoS):

In center-of-sets TR, each type-2 consequent set is replaced by its centroid. Then, the weighted average of these centroids is found, being the weight of each centroid the degree of firing of it corresponding rule (Mendel, 2001). CoS TR is chosen because its convenient trade-off between accuracy and computational cost. It is more accurate than Heights method (Mendel, 2001) and computationally less expensive than Cent TR.

Enhanced Karnik Mendel Algorithm (EKM):

The EKM algorithm is an iterative procedure to obtain the generalized centroid of an T2-FS. It uses statistically defined values as initialization points to reduce the amount of iterations that are necessary, in this way it converges monotonically and super-exponentially fast (Wu & Mendel, 2009). In addition, as it has been mentioned in (Leottau & Melgarejo, 2010a), EKM is the fastest algorithm reported up to that date for finding the generalized centroid of a inferred IT2-FS over DSC technology.

3. Proposal to design an IT2-FLC for a truck backer-upper

A proposal for designing an IT2-FLC for tracking the trajectory of a mobile robot application involving some of the T1-FLC and T2-FLC properties is introduced in (Leottau & Melgarejo, 2010b) and briefly explained in this section. T2-FSs are used for the initial modelling taking advantage of its capability to handle linguistic uncertainty because a large number of T1-FSs are embedded. T1-FSs are used in order to make easier a fine tuning, taking into account their small amount of parameters. Finally, returning to T2-FLSs is proposed for handling the uncertainty that appears in the final system performance. The truck backer-upper (Castro et al., 2007; Nguyen & Widrow 1989, Wang, 1997) is a non-linear control problem, whose model is depicted in Figure 2. Backing a trailer truck is a difficult task which requires a great deal of practice. Usually a truck driver backing, going forward, backing again, etc. if forward movements are not permitted, a more difficult problem emerges. Thus, the problem treated in this chapter is to control the steering of a trailer truck for tracking a trajectory, where only are allowed backing movements.

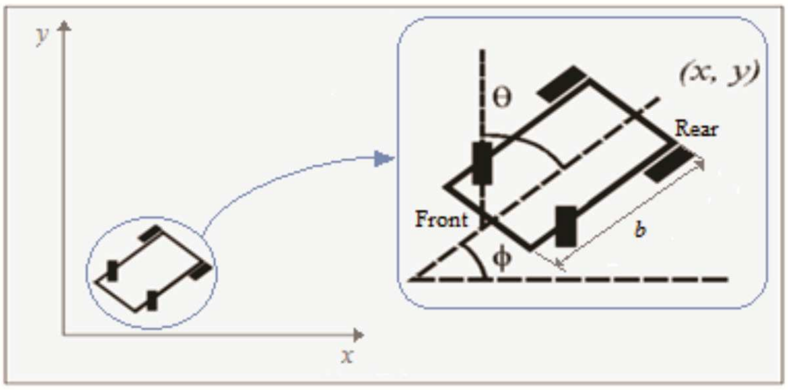


Fig. 2. The simulated truck.

3.1 Modeling the system dynamics: Truck mathematical model

In this step, the control process is studied in order to find and determine the model of the system dynamics. There are different strategies to do it (Kuo & Golnaraghi, 1996; Mendel, 2007; Wang, 1997). One of the most commonly used is by obtaining the mathematical model using physical schemes such as this case (Wang, 1997).

This problem is proposed in (Wang, 1997). The mathematical model that describes the system to control is shown below, where b is the length of the truck.

$$x(t+1) = x(t) + cos[\varphi(t) + \theta(t)] + sin[\theta(t)]sin[\varphi(t)] \tag{1}$$

$$y(t+1) = y(t) + sin\big(\varphi(t) + \theta(t)\big) - sin[\theta(t)]cos[\varphi(t)] \tag{2}$$

$$\varphi(t+1) = \varphi(t) - sin^{-1}[2sin(\theta(t))/b] \tag{3}$$

Since this work is focused for academic proposes, b=4 inch is assumed, however as it is handled in (Wang, 1997), units of that length can be considered depending on the operational environment. The truck position is determined by three state variables φ, x and y, where φ is the angle of the truck regarding the horizontal line as shows in Figure 2. The steering angle θ is the control variable to the truck which only moves backward. For the sake of simplicity, y does not have to be considered as state variable, thus position in the vertical axis is not controlled.

3.2 Initial design of an IT2-FLC for the truck

In this step, parameters as feedback variables, inputs (antecedents), outputs (consequents), universe of discourse, control action, rule base and the shape of MFs are defined. Using IT2-FS in this stage is proposed in order to obtain a preliminary design including linguistic uncertainty associated to ambiguity in the knowledge of the control process (Leottau & Melgarejo, 2010b).

Inputs of the controller are $(Ex, E\varphi \; and \; \Delta\varphi)$, where $Ex = Xref - x(t)$ is the error of x position regarding the reference trajectory *Xref*. $E\varphi = \varphi ref - \varphi(t)$ is the error of φ angle regarding the reference *φref* which is fixed to *φref*=$\pi/2$. $\Delta\varphi = \varphi(t) - \varphi(t-1)$ is the changing of φ. The output is θ, which is limited to $[-2\pi/9\,,2\pi/9]$. The universe of discourse is assumed as $Ex \in [-20, 20]$, $E\varphi$ and $\Delta\varphi \in [-\pi, \pi]$ and $\theta \in [-\pi/3\,,\pi/3]$.

Based on intuitive reasoning, a preliminary design is obtained. Two antecedent sets for each input are defined, one to determine positive input values and their respective complement set for negative values. Eight rules are obtained by combining the antecedent sets and using four consequent sets are proposed: Positive high (+ +), positive small (+), negative small (-) and negative high (- -). The obtained rule base is shown below:

$$\begin{aligned}
&Rule\ 0\!: if\ Ex^{-}\ \ and\ E\varphi^{-}\ and\ \Delta\varphi^{-}\ then\ \theta^{+}\\
&Rule\ 1\!: if\ Ex^{-}\ \ and\ E\varphi^{-}\ and\ \Delta\varphi^{+}\ then\ \theta^{-}\\
&Rule\ 2\!: if\ Ex^{-}\ \ and\ E\varphi^{+}\ and\ \Delta\varphi^{-}\ then\ \theta^{-}\\
&Rule\ 3\!: if\ Ex^{-}\ \ and\ E\varphi^{+}\ and\ \Delta\varphi^{+}\ then\ \theta^{--}\\
&Rule\ 4\!: if\ Ex^{+}\ \ and\ E\varphi^{-}\ and\ \Delta\varphi^{-}\ then\ \theta^{++}\\
&Rule\ 5\!: if\ Ex^{+}\ \ and\ E\varphi^{-}\ and\ \Delta\varphi^{+}\ then\ \theta^{+}\\
&Rule\ 6\!: if\ Ex^{+}\ \ and\ E\varphi^{+}\ and\ \Delta\varphi^{-}\ then\ \theta^{+}\\
&Rule\ 7\!: if\ Ex^{+}\ \ and\ E\varphi^{+}\ and\ \Delta\varphi^{+}\ then\ \theta^{-}
\end{aligned} \tag{4}$$

Since negative values in antecedent sets are handled as the complement of their respective positive values, in practical terms $\mathrm{Ex}^{-}, \mathrm{E\varphi}^{-}$ and $\Delta\varphi^{-}$ respectively are equivalent to $\overline{\mathrm{Ex}^{+}}$, $\overline{\mathrm{E\varphi}^{+}}$ and $\overline{\Delta\varphi^{+}}$.

3.3 Initialization the IT2-FLC for the truck

In order to evaluate the IT2-FLC performance, it is convenient to define a figure of merit. In this way, it is possible to carry out a preliminary tuning, searching for an acceptable response. In this case, an exhaustive tuning is not carried out because to achieve stability is enough (Leottau & Melgarejo, 2010b). It could be carried out using simulations.

Determine an initial wide for FOUs in order to include the uncertainty in the system modelling is the first task of this initialization step. The main sources of uncertainty that would be present in this implementation are:

- Uncertainties in inputs to the FLC. The sensors measurements would be affected by noise levels or by variant conditions of observation (Hagras, 2007), taking into account that these kind of mobile robots applications usually are employed in outdoor environments (Hagras, 2004).

- Uncertainties in control outputs. Actuators characteristics could change with time. The mobile robots mechanics often are susceptible to wear or tear (Hagras, 2007).
- Uncertainties associated with changing operational conditions (Mendel, 2007).

The sources of uncertainty mentioned above are included in the final design. But, as it is mentioned in step 3.2 of this procedure, FOUs in this part of design, makes reference to linguistics uncertainties. This is the main advantage of use IT2-FS here, because the initial tuning would be based in some knowledge of the control process.

As it is proposed in (Leottau & Melgarejo, 2010b), it is used the middle of universe as center of antecedent sets location and the 10% of universe size as a measure for uncertainty. As universe size of $E\varphi$ and ΔE is 2π, the FOU wide of this IT2-FSs is initially defined as $2\pi \cdot 10\% = 0.2\pi$ and center of sets are located at zero. Please, see Figure 7.a for more clarity.

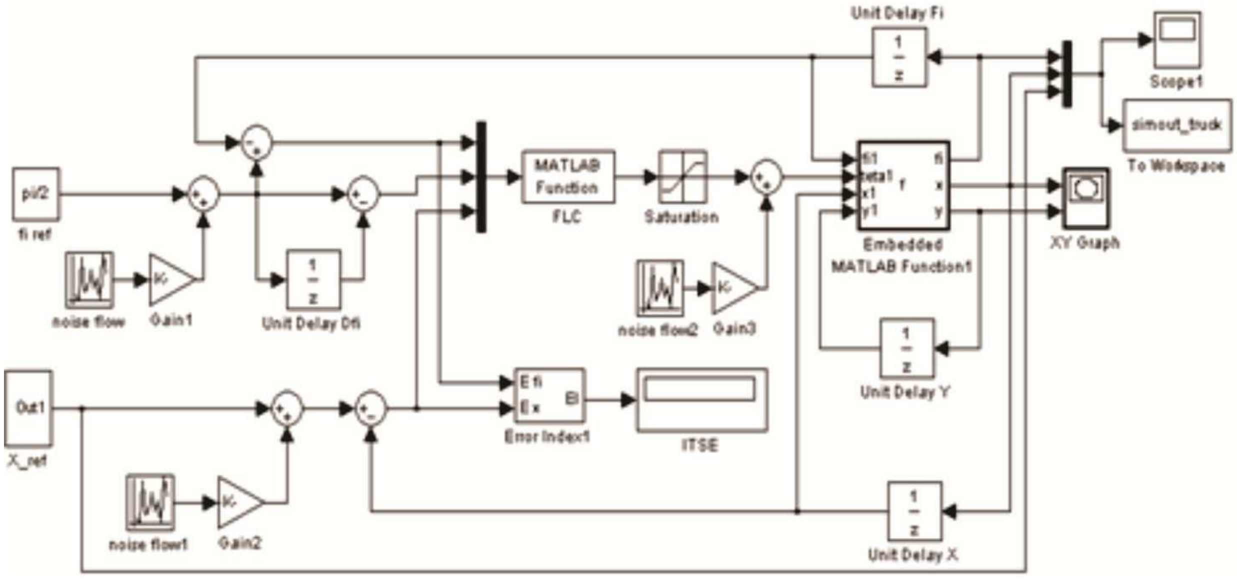


Fig. 3. Simulink model for the truck system.

The whole system is simulated using Matlab®, fuzzy Toolbox and Simulink. In addition, the IT2-FLC is designed and simulated with the IT2 Fuzzy Toolbox by the Tijuana Institute of Technology and Baja California Autonomous University (Castro et al., 2007). The Simulink model for the truck system is depicted in Figure 3.

As a figure of merit, the integral time square error (ITSE) is used (Dorf & Bishop, 1998). The objective of this example application is not to achieve an optimal performance in the controller. Thus, tuning parameters is carried out using the prove-error method, looking that the mobile robot follows a defined trajectory (see Figure 6).

3.4 The reduction of dimensionality

The reduction of dimensionality consists of converting the IT2-FLC into a T1-FLC. It is possible compressing the FOU in the IT2-FSs up to transform it in a T1-FS as is shown in Figure 4. The main objective of this procedure is making easier the tuning of fuzzy sets parameters (Leottau & Melgarejo, 2010b). The amount of parameters used for modelling a T2-FS is greater than for a T1-FS. In this way, tuning these parameters in a T1-FLC is easier and faster (Mendel, 2001).

In this application, the embedded T1-FS that are located in the middle of each FOU is used as a tuning start point. Once the IT2-FSs are converted to T1-FSs, the MF parameters' tuning is carried out using Microsoft Excel® solver tool that uses the Generalized Reduced Gradient Method as non linear optimization algorithm (Frontline Systems, 2010). A screen of used Excel® spreadsheet is shown in Figure 5.

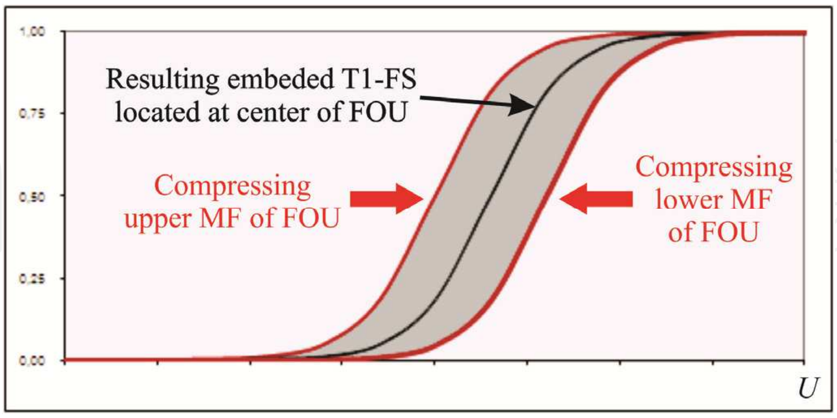
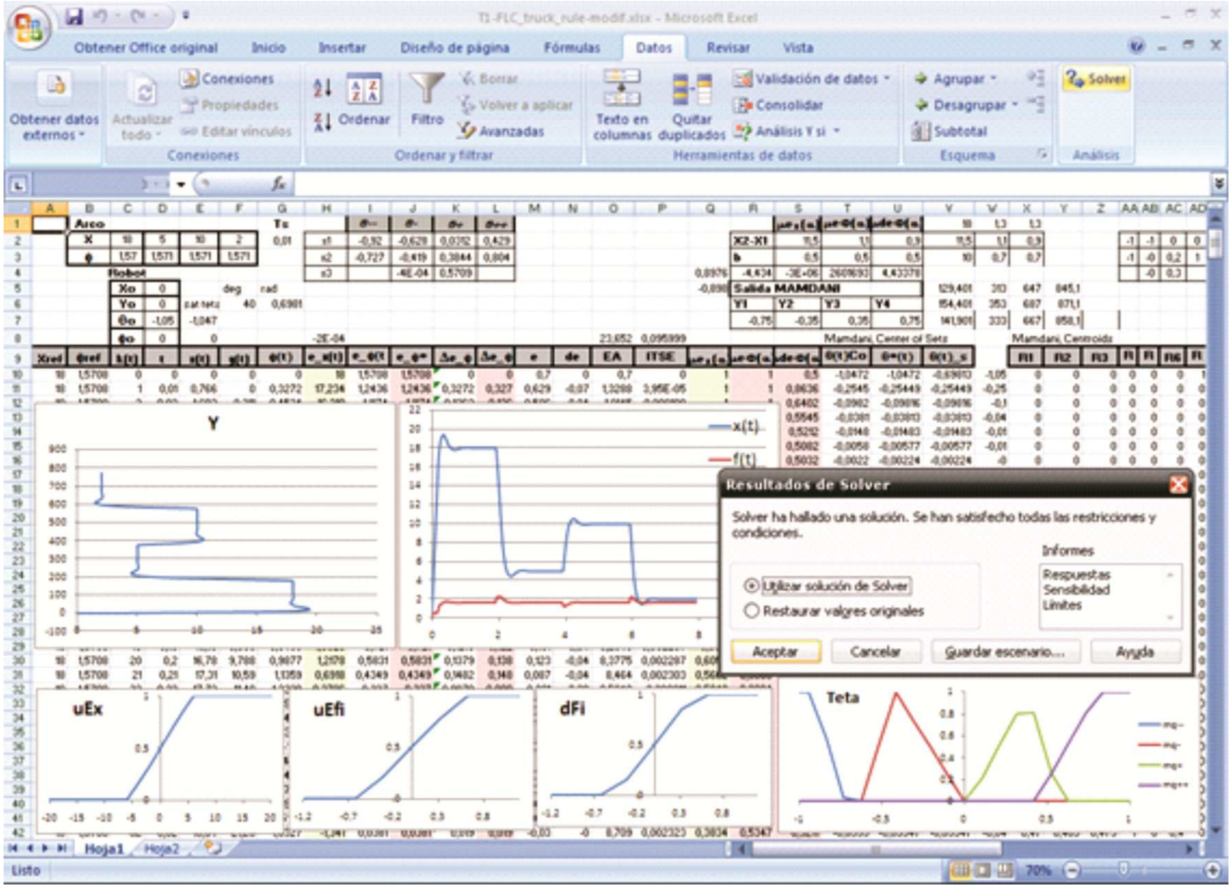


Fig. 4. Reducing the dimensionality of an IT2-FS.



Resultados de Solver

Solver ha hallado una solución. Se han satisfecho todas las restricciones y condiciones.

◉ Utilizar solución de Solver
○ Restaurar valores originales

Informes
Respuestas
Sensibilidad
Límites

Aceptar | Cancelar | Guardar escenario... | Ayuda

Fig. 5. A screen of used Excel® spreadsheet for MF parameters' tuning.

3.5 Returning to Type-2

Once the T1-FLC parameters are tuned, returning to T2-FLS scheme using the same uncertainties defined in step 3.3 is proposed in (Leottau & Melgarejo, 2010b), but in this case, locating the FOUs over their embedded T1-FSs obtained in previously tuning. So, By using the tuned T1-FSs obtained in step 3.4 as the central embedded T1-FS as in Figure 4, the same FOUs wide resulting by step 3.3 are included, so the FLC returns to its initial interval type-2 nature. Then a fine tuning is carried out again by using the prove-error method taking into account the system response obtained by simulation.

4. Hardware implementation of the IT2-FLC for the truck

Designed IT2-FLC showing in section 3 has the following characteristics: three inputs ($N=3$), two interval type-2 fuzzy sets (IT2-FSs) by input ($M=2$), six antecedent sets ($M_A=M \cdot N=6$), eight rules ($R=8$), four IT2-FSs in the consequent ($M_C=4$). It must be taken into account that negative values in antecedent sets are handled as the complement of their respective positive values, so in practical terms it is defined one IT2-FSs by input ($M^{\dagger} = 1$ and $M_A^{\dagger} = 3$). It is applicable for example for calculating the data memory usages. Universe of discourse' sizes have been defined in section 3.2. Its discretization levels in antecedents (D_A) and consequent (D_C) must be defined according computational and hardware resources available on the embedded platform to this implementation.

4.1 Available embedded platform for implementing the IT2-FLC

The available embedded platform for implementing the IT2-FLC is the Digital Signal Controller DSC-56F8013 evaluation board (Freescale Semiconductor, 2010). Next, some hardware resources of DSC-56F8013 are listed in Table 1.

Word length	16-bit
Core frequency	Up to 32 MIPS at 32MHz
Program Flash Memory	8KB (4KWord)
RAM	2KB (1KWord)
PWM module	One 6-channel
ADC	Two 3-channel 12-bit
Communication Interfaces	One Serial Communication Interface (SCI) with LIN slave functionality. One Serial Peripheral Interface (SPI). One Inter-Integrated Circuit (I2C) Port.
Quad Timer	One 16-bit Quad Timer.
On-Chip Emulation	JTAG/Enhanced On-Chip Emulation (OnCE™) for unobtrusive, real-time debugging.

Table 1. DSP-56F8013 basic features.

4.2 Determining computational model for implementing the IT2-FLC for the truck system

A summary of computational models for implementing the embedded IT2-FLC is shown in Table 2. The rest of this section explains why these computational models are chosen and how can be determined some of its features.

A methodological proposal for implementing interval type-2 fuzzy processors (IT2-FP) over DSC technology is reported in (Leottau & Melgarejo, 2010a). There, several computational

	IT2-FLC with centroid TR	IT2-FLC with CoS TR
Fuzzyfier	FCA	MBA
U_A =D_A	250	250
Inference Engine	FCA	-
U_C	1000	1000
D_C	100	4
RAM Consumption	456 Bytes (22.8%)	44 Bytes (0.28%)
DATA Mem. Consumption	4600 Bytes (57.5%)	136 Bytes (0.017%)

Table 2. Computational models features.

models have been characterized and tested over an IT2-FLS with the following characteristics: two inputs (*N=2*), three IT2-FSs by input (*M=3*), six antecedent sets (M_A=*M·N*=6), nine rules (*R=9*), nine IT2-FSs in the consequent (M_C=*9*). Since characteristics of that IT2-FLS are similar and even more complex than the IT2-FLC designed for this mobile robot application, following considerations are based in that methodological proposal.

4.2.1 Processing time

According inference times reported in (Leottau & Melgarejo, 2010a) for a DSC-56F8013, a complete inference is carried out between 500uS and 60mS for the fastest (MBA, Dc=10) and the slowest (FCA, Dc=1000) computational model respectively. That times can be used as references for determine the IT2-FLC sample time (*Ts*) for the truck system. Among this time, the embedded FLC samples the inputs variables for computing a new crisp output.

Since θ output is limited to ±40º, it is necessary to define a sampling time such that θ can rotate at least 80º. If it is used a Hitec servo motor (Hitec Servo Motors, 2010) that handling operating speed from 0.05 sec/60º, defining a sampling time of *Ts=100mS* is enough, maintaining a convenient extra range.

4.2.2 Computational model with centroid type-reduction

Computational model chosen for the IT2-FLC with centroid TR is implemented with FCA for fuzzyfier and inference engine. Although (Leottau & Melgarejo, 2010a) evidences that processors implemented with FCA are slower than those implemented with MBA, this is not a problem taking into account that selected sampling time is enough. On the other hand, available DSC56F8013 offers limited memory resources, with that respect (Leottau & Melgarejo, 2010a) evidence that FCA is the computational model with less data memory consumption. In this way, we consider that cost-benefit trade-off of this computational model is convenient for this application.

4.2.3 Computational model with CoS type-reduction

Computational models with CoS TR offer small data memory consumption but lower accuracy regarding methods as Centroid (Mendel, 2001). On the other hand, computational models based on MBA offer the highest memory consumptions and the highest accuracy (Leottau & Melgarejo, 2010a). Thus, by combining CoS TR with a fuzzyfier based on MBA can be a convenient trade-off between accuracy and data memory usage. Thus, computational model chosen for the IT2-FLS with CoS TR is implemented with MBA for the fuzzyfier.

4.2.4 Interconnecting the emulated mobile robot plant with the embedded IT2-FLC

The truck system is emulated by using Matlab® over Simulink, thus it is necessary interconnect the DSC56F8013 with the computer in order to communicate the embedded IT2-FLC with the truck system. Since 56F8013 evaluation board include a RS-232 interface, for easy connection to a host processor the serial port is used. Since emulation is carried out over a general propose processor which serial port baud-rate is limited to 115200, it is selected a sped of 57600 bauds in order to not to force the platform.

As it is mentioned in section 3, in a context of simulation, a block of Matlab® fuzzy toolbox or it2fuzzy toolbox is included in the Simulink model for simulating the T1-FLC and the IT2-FLC respectively (Figure 3). In order to interconnect the truck system plant with the IT2-FLC embedded in the DSC56F8013, these blocks must be replaced by a Matlab function that carries out the RS-232 serial port communication.

4.2.5 Universe sizes and discretización levels regarding RS-232 communication

Universes of discourse in the process are defined as: $Ex \in [-20, 20]$ $(U_{Plant\ X} = 40)$, $E\varphi$ and $\Delta\varphi \in [-\pi, \pi](U_{(Plant\ \varphi \text{ and } \Delta\varphi)} = 2\pi)$ and $\theta \in [-\pi/3, \pi/3]$ $(U_{(Plant\ \theta)} = 2\pi/3)$.

RS-232 interface has been chosen for connecting the embedded IT2-FLC with the truck system emulated by using Matlab® over Simulink. Since RS-232 interface can send and receive up to one Byte per time (2^8=256 levels), a universe of discourse between [0, 249] is defined in order to transmit just one byte per input, so U_A=D_A=250. Resolution of input universes is determined as: Input_res=U_{Plant}/D_A. Thus, Ex_res=40/250=0.16, Eϕ_res=2π/250≃0.025 and Δϕ_res=2π /250≃0.025.

Output universe is defined between [-π/3, π/3]. If U_C is the output universe size in the embedded IT2-FLC, calculating output resolution by using D_A=U_C, θ_res= 2π/(3·250)=0.00837=0.48°. Since computational models chosen particularly offer a low accuracy and output is limited to ±40°, we consider that 0.48° is a low resolution. Thus, a universe of discourse between [0, 999] is defined, so U_C=1000 and θ_res= 0.00209=0.12°.

It is necessary to transmit from the PC serial port to the IT2-FLC embedded on the DSC56F8013 three bytes, one per input of IT2-FLC. Then, it is necessary to receive one data as its output. Since the output universe of the embedded IT2-FLC has been to a thousand points. In order to receive the output of the IT2-FLC from DSC56F8013, it is necessary to divide this result. So, units, tens and cents are transmitted as three independent bytes those must be concatenated by a Matlab RS-232 function before to be injected to the truck plant.

Universes of discourse for the embedded IT2-FLC are defined in the previous paragraphs as and. So, it is necessary a scalization as:

$$V_{FLC} = V_{Plant} \cdot U_{FLC}/U_{Plant} + U_{FLC}/2 \tag{5}$$

Where V_{FLC} and U_{FLC} are respectively the scaled value and the universe size that handles the embedded FLC and U_{Plant} and V_{Plant} are respectively the universe size and the un-scaled value that handles the truck plant.

4.2.6 Discretization levels in the consequent regarding available memory resources

According (Leottau & Melgarejo, 2010a), the RAM usage expressed in Words for the inference engine with centroid type-reduction is $2{\cdot}D_C{+}2{\cdot}R$ and for all discussed fuzzyfiers is $2 \cdot M_A$. Since DSC56F8013 offers 1KWord in RAM, the maximum discretization levels in the consequent regarding RAM resources can be calculated as:

$$D_{C(max)} = (RAM_{available} - 2 \cdot M_A - 2 \cdot R)/2 = 486 \quad (6)$$

On the other hand, the DATA memory usage for the inference engine with centroid TR and MBA is $2{\cdot}D_C{\cdot}M_C$ and $2{\cdot}M_A{\cdot}D_A$ for the fuzzyfier. Since DSC56F8013 offers 4KWord in Flash memory, the maximum discretization levels in the consequent regarding DATA memory resources can be calculated as:

$$D_{C(max)} = (Flash_{available} - 2 \cdot M_A^{\dagger} \cdot D_A)/(2 \cdot M_C) = 312 \quad (7)$$

Thus, with centroid TR and MBA for fuzzyfier and inference engine, *Dc* must be less than 312. On the other hand, RAM usage for inference engine with CoS TR is *2·R=16 Words.* Equation 7 evidences that *Dc* must be than three hundred twelve points. So, *Dc* is defined as one hundred taking into account the study carried out in (Leottau & Melgarejo, 2010a). This study evidence that by using *Dc=100,* results maintaining a trade-off between accuracy, memory consumption and processing time.

5. Tests, results and discussion

5.1 Obtained IT2-FLC for the truck backer-upper

The obtained IT2-FLC for tracking the trajectory of the truck Backer-Upper is presented in this sub-section. The IT2-FLC has been designed following the procedure described throughout section 3. Preliminary response of *x* and *φ* obtained in step 3.3 is presented in Figure 6. This response has been obtained with the IT2-FSs resulting in the initialization step. These IT2-FSs and resulting T1-FSs tuned in step 3.4 are shown in Figure 7.a. The final IT2-FSs obtained after step 3.5 are shown in Figure 7.b.

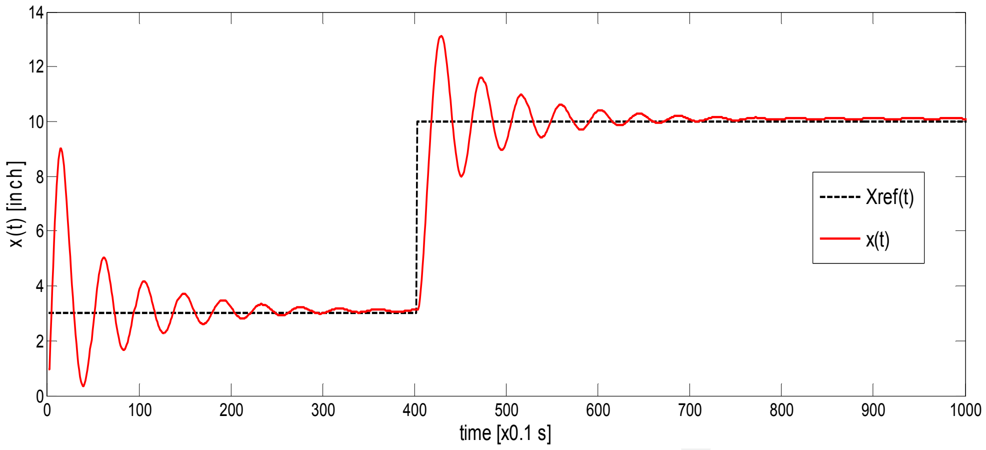
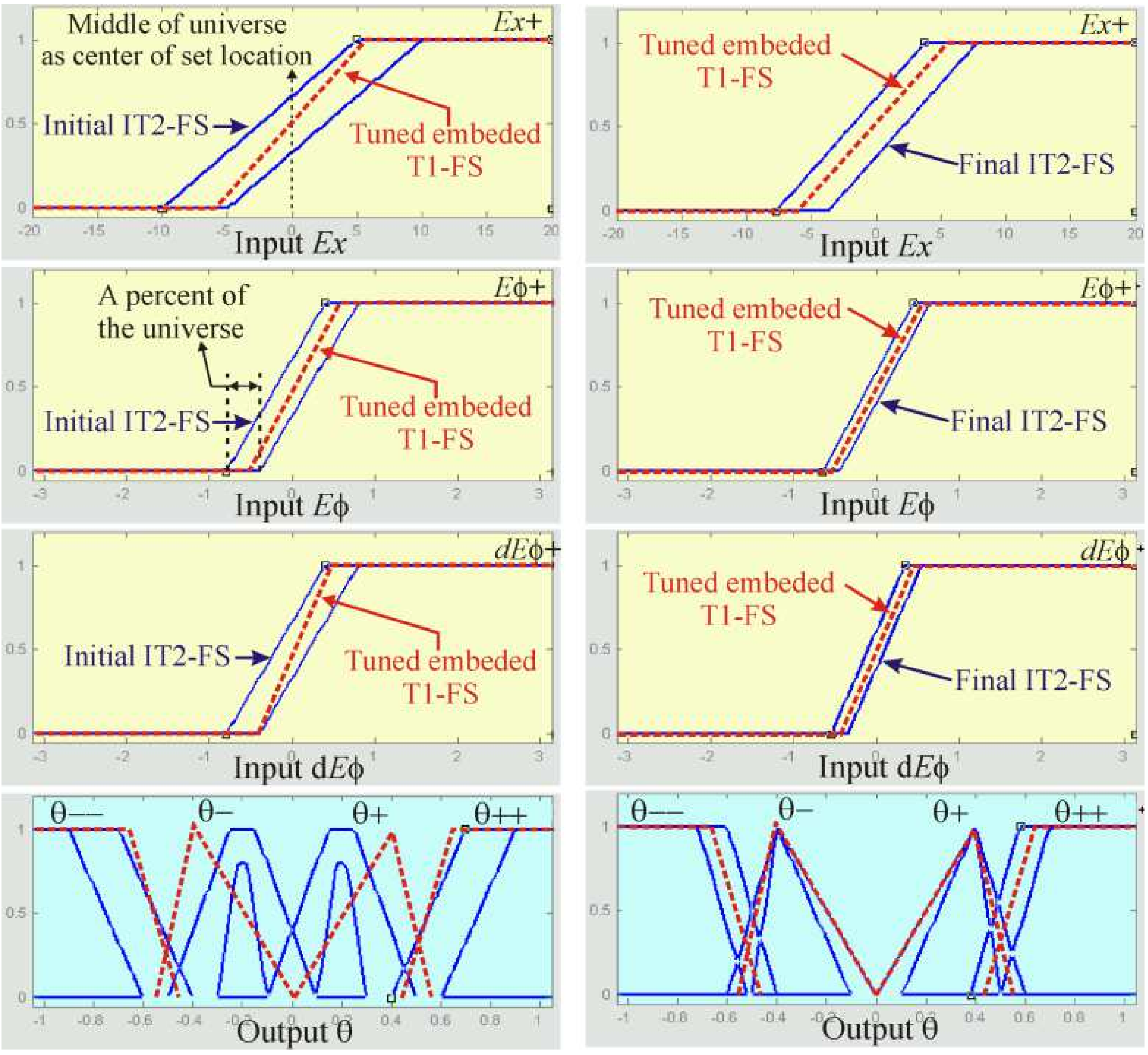


Fig. 6. IT2-FLC design procedure. Preliminary response of *x* and obtained in step 3.3.

5.2 Tests

Figure 8 resumes the approach outlined throughout this chapter applied for designing and implementing an IT2-FLC for the truck system. After modelling and defining parameters steps in 3.1 and 3.2, initialization and the first design in step 3.3 is carried out by simulation with Matlab® over Simulink. Subsequent to reduction of dimensionality in step 3.4, the tuning of the T1-FLC is carried out by using Excel®. Then, tuned T1-FLC is tested over simulation and emulation. Later than returning to Type-2 and the final fine tuning in step



a) b)

Fig. 7. IT2-FLC design procedure. a) IT2-FSs first design in step 3.3 (Blue lines) and tuned T1-FSs resulting in step 3.4. (Red dotted lines) b) IT2-FSs final implementation after step 3.5 (Blue lines) and newly tuned T1-FSs resulting in step 3.4. (Red dotted lines).

3.5, the resulting IT2-FLC is tested over simulation and emulation too. As it is mentioned before, the simulations are carried out by using fuzzy and it2-fuzzy toolboxes and emulations are carried out over the 56F8013 board whose programming is debugged and loaded by using the Codewarrior® suite for DSC56800E v8.2.3.

Different tests are achieved for evaluating the final performance of designed T1-FLC and IT2-FLC. A combined trajectory in x is used as reference form emulating different operating points to the FLC. In addition, random noise generators are inserted in the inputs and outputs of FLC in order to emulate some sources of uncertainty such as sensors measurements and changes of actuators characteristics. The FLCs are tested under three different noise environments: (1) without noise, (2) moderate noise with amplitude as 0.2% of maximum value that respective variable can take and (3) high noise with amplitudes as 1%. E.g. Ex input can take values within $|0, 20|$, then the amplitude of moderate noise is: $A_{mN}(Ex)$ =20 x 0.2% = 0.04. In order to emulate a stochastic nature of uncertainty, every noise environment is tested five times changing the initial seed of the random noise generator.

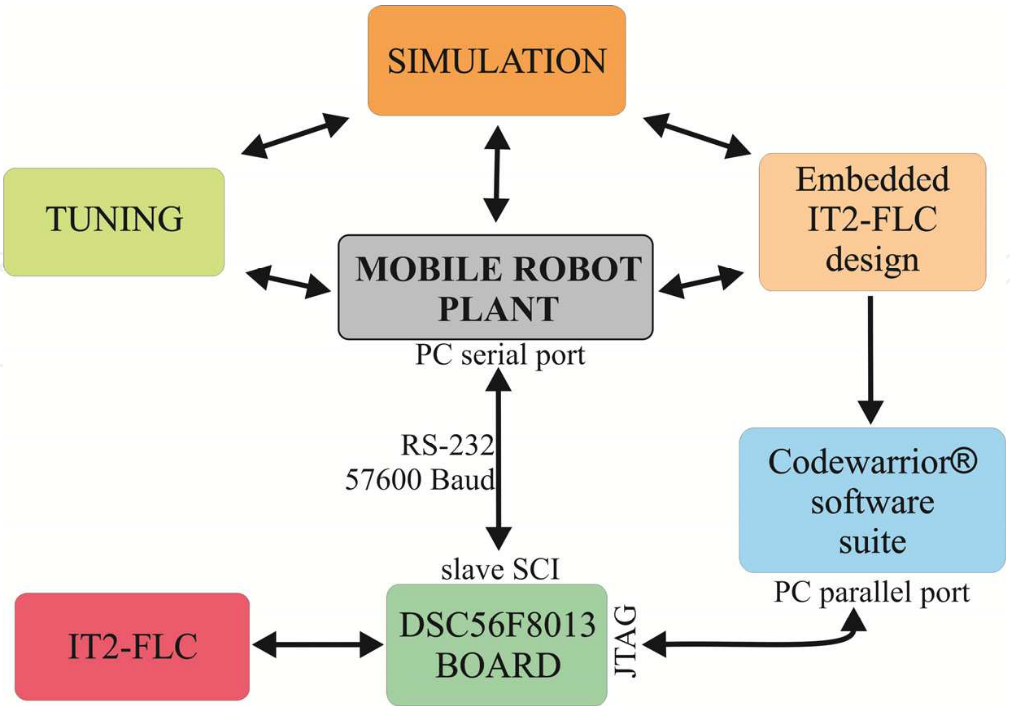


Fig. 8. Conceptual diagram of proposed approach for implementing the embedded IT2-FLC for the truck system.

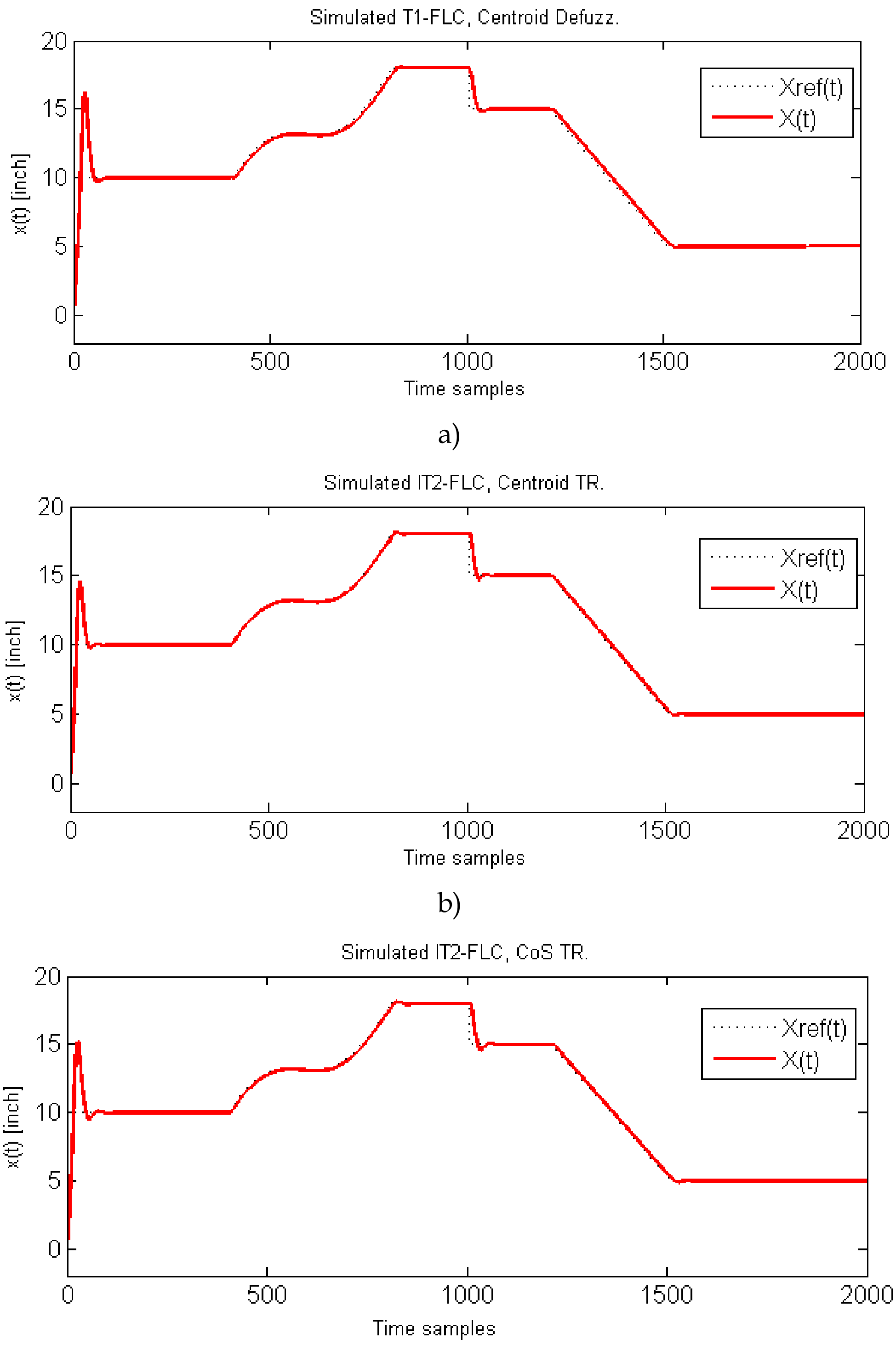
As it is mentioned in section 2, there are several type reducer and defuzzyfier methods. In order to determine the impact of that in the final performance of designed T1-FLC and IT2-FLC, a test is carried out changing the TR or defuzzyfier method in each FLC. Under simulation, some of available methods in its respective Matlab® toolbox (*fuzzy* for type-1 and *it2fuzzy* for type-2) are tested: centroid and center of sets (CoS) by the T2-FLC and centroid, bisector and mean of maximum (Mom) by T1-FLC. Under emulation, considered methods in section 2 are tested: Centroid and CoS.

The tests procedure is described as follows:

1. Set a TR or defuzzification method for the FLC.
2. Set a noise environment.
3. Run one complete simulation or emulation and register ITSE.
4. Repeat the same simulation (steps 2-3) five times with a different initial noise seed in order to obtain the average and standard deviation of ITSE.
5. Repeat steeps 2-4 for the three noise environments considered.
6. Repeat steeps 1-5 for the TR and defuzzification methods considered.
7. Repeat the procedure for the T1-FLC and the IT2-FLC over simulation and over emulation.

5.3 Results

Results for x and φ are presented in Figure 9 to Figure 12. Figure 9 and 10 show the response of FLCs tested without noise, under simulation and emulation respectively. Figure 11 shows the response of simulated FLCs and Figure 12 shows the response of hardware implemented FLCs, both cases tested under the worst noise condition, emulating a high uncertainty environment. Since all FLCs are tested under three different noise environments and five different initial seeds for each noise environment, obtained results for ITSE as performance indices are presented as an average and its standard deviation. It is shown in tables 3 and 4.

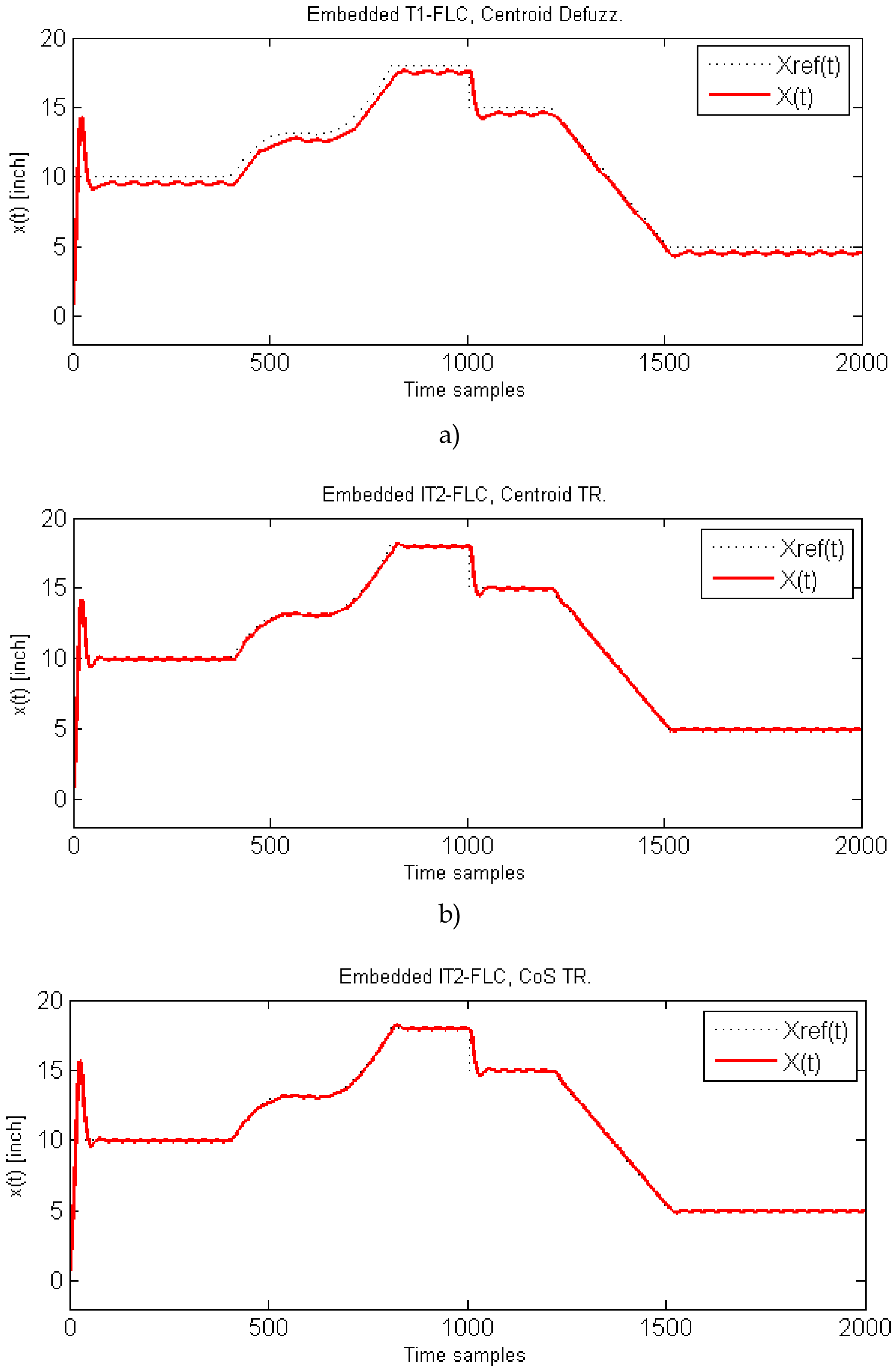


a)

b)

c)

Fig. 9. Without noise and under simulation, trajectory for *x* of: *a)* T1-FLC with Centroid defuzzification. *b)* IT2-FLC with Centroid type-reduction. *c)* IT2-FLC with CoS type-reduction. (*time*=*Time_samples/Ts*, where*Ts*=*0.1s*)

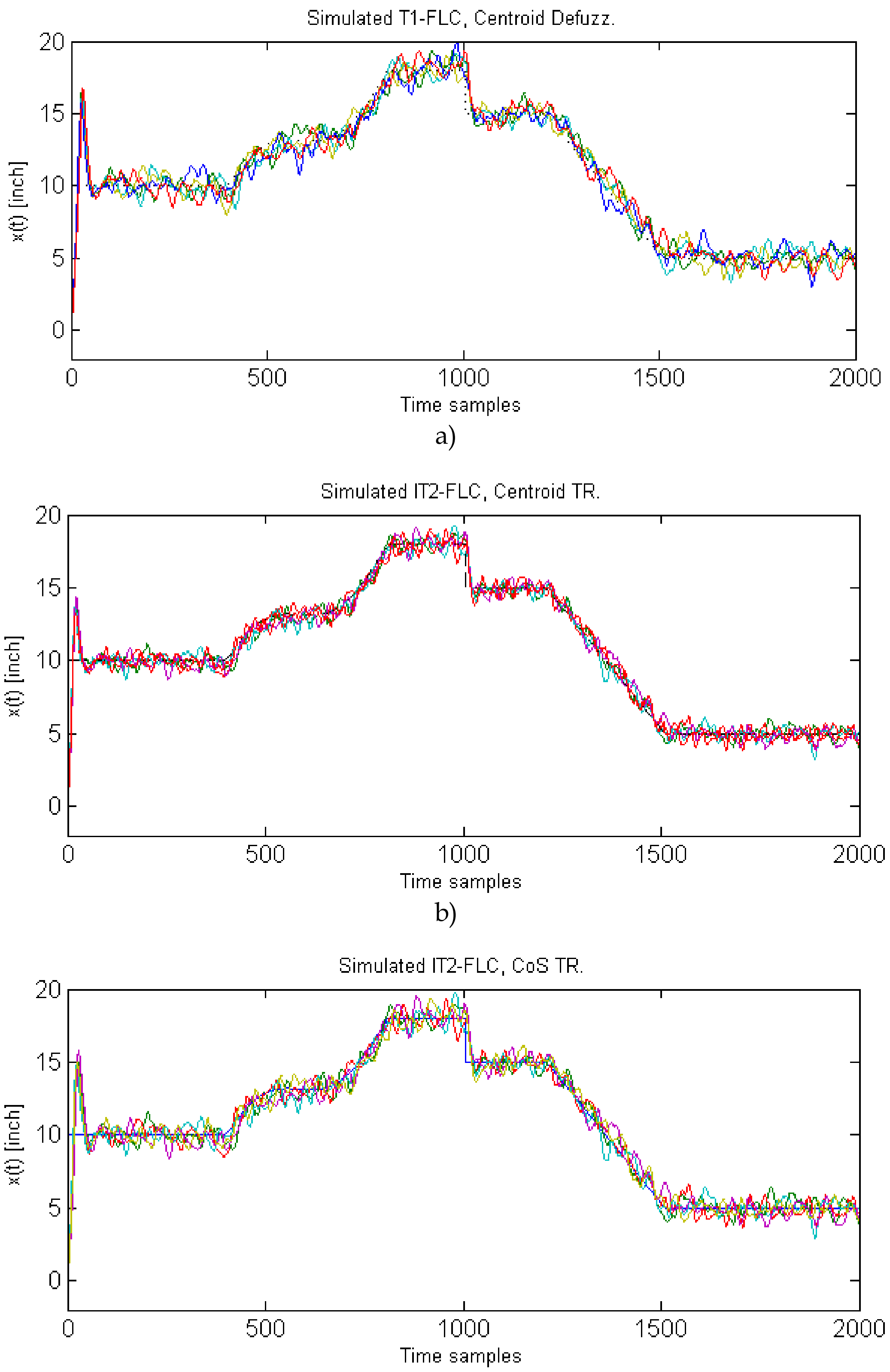


a)

b)

c)

Fig. 10. Without noise, trajectory for x of hardware implemented: a*)* T1-FLC with centroid defuzzification. b*)* IT2-FLC with centroid type-reduction. c*)* IT2-FLC with CoS type-reduction. (*time*=*Time_samples/Ts*, where*Ts*=*0.1s*)

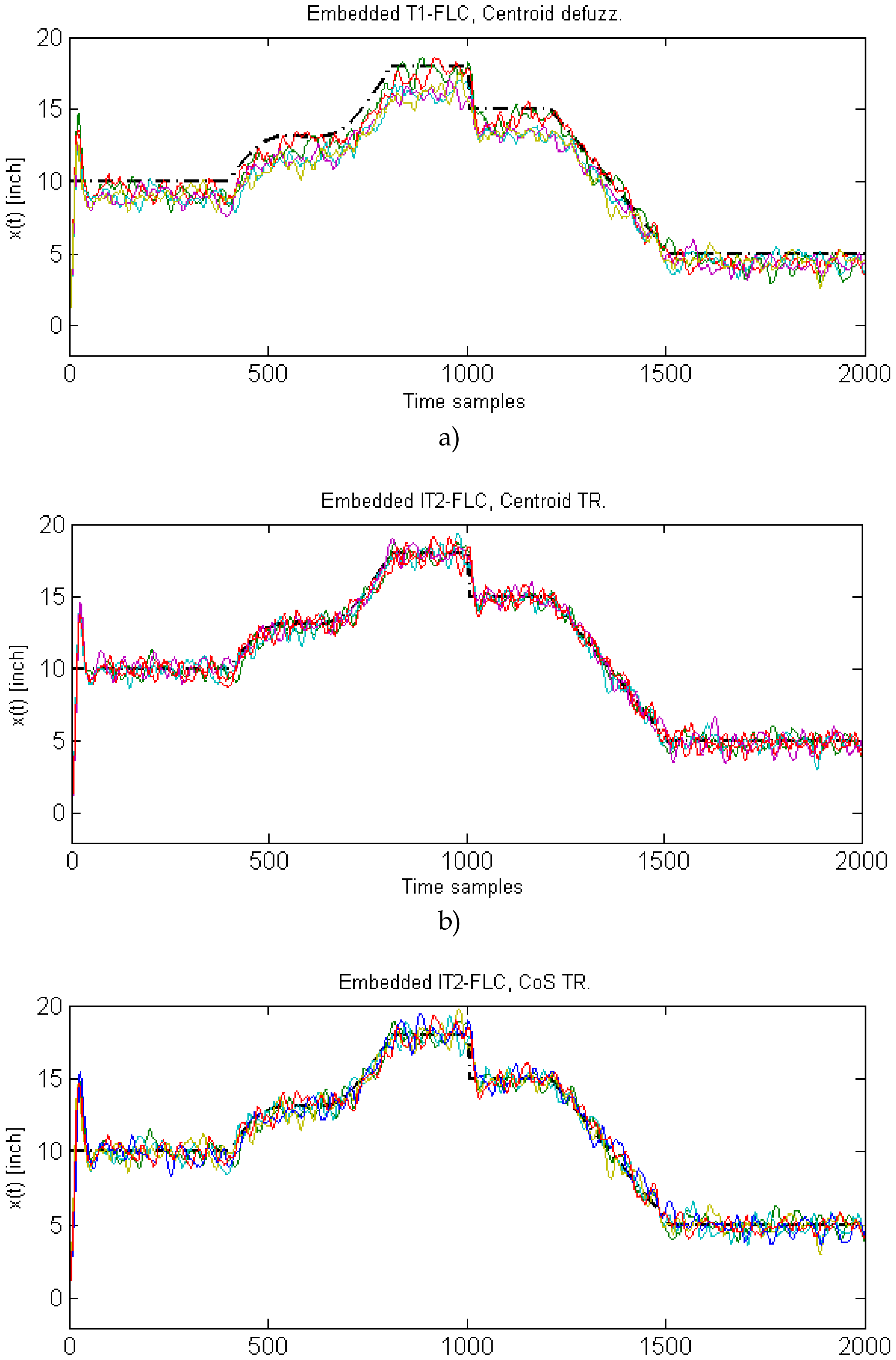


a)

b)

c)

Fig. 11. Under the worst noise condition and simulation, trajectory for *x* of: *a)* T1-FLC with centroid defuzzification. *b)* IT2-FLC with centroid type-reduction. *c)* IT2-FLC with CoS type-reduction. (*time*=*Time_samples/Ts*, where*Ts*=*0.1s*)



a)

b)

c)

Fig. 12. Under the worst noise condition, trajectory for *x* of Hardware implemented: a*)* T1-FLC with Centroid defuzzification. b*)* IT2-FLC with centroid type-reduction. c*)* IT2-FLC with CoS type-reduction. (*time*=*Time_samples/Ts*, where*Ts*=*0.1s*)

Noise level	**Cent.**		**Bisector**		**MoM**		**Cent. Hardware**		**CoS Hardware**	
	ITSE	*Desv.*	*ITSE*	*Desv.*	*ITSE*	*Desv.*	*ITSE*	*Desv.*	*ITSE*	*Desv.*
Zero	18.48	-	†	-	††	-	68.025	-	†††	-
Mod.	22.27	2.10	†	-	††	-	84.84	5.625	†††	-
High	98.93	14.71	†	-	††	-	137.38	13.94	†††	-

† Does not reach the reference.
†† Response turns unstable.
††† Does not reach the reference and presents oscillations.
Since tests have evidenced that the T1-FLCs with MoM and Bisector defuzzyfier and the embedded T1-FLC with CoS defuzzyfier do not present a successfully performance, response for these FLCs have not shown in Figs 9 to 12.

Table 3. Results of ITSE for the T1-FLC.

Noise level	**Cent.**		**CoS**		**Cent. Hardware**		**CoS Hardware**	
	ITSE	*Desv.*	*ITSE*	*Desv.*	*ITSE*	*Desv.*	*ITSE*	*Desv.*
Zero	11.23	-	13.73	-	11.7	-	13.77	-
Mod.	13.78	0.93	18.02	1.70	13.84	0.85	18.37	0.87
High	44.57	4.81	77.41	10.49	62.122	6.45	82.92	8.76

Table 4. Results of ITSE for the IT2-FLC.

5.4 Discussion

Without noise and using centroid TR, Simulated IT2-FLC ITSE is about 39.2% better than simulated T1-FLC ITSE and embedded IT2-FLC ITSE is about 82.8% better than embedded T1-FLC ITSE. Under moderate noise and using centroid TR it can be said that: Simulated T2-FLC ITSE is about 39.4% better than simulated T1-FLC ones and embedded T2-FLC ITSE is about 83.69% better than embedded T1-FLC. Under high noise and using centroid TR Simulated T2-FLC ITSE is about 54.9% better than simulated T1-FLC ones and embeddedT2-FLC ITSE is about 54.78% better than embedded T1-FLC ones.

ITSE of simulated IT2-FLC with centroid type reduction is between 18% (without noise) and 42% (with high noise) better than the IT2-FLC with CoS. Besides, ITSE of hardware implemented IT2-FLC with centroid type reduction is between 15% (without noise) and 25% (with high noise) better than the IT2-FLC with CoS.

The ITSE of the simulated IT2-FLC with Cent type reducer is between 0.4% (with moderate noise) and 28% (with high noise) better than the embedded IT2-FLC whit the same type reducer. Besides, the ITSE of the simulated IT2-FLC with CoS type reducer is between 0.3% (without noise) and 6.7% (with high noise) better than the embedded IT2-FLC whit the same type reducer. This difference is caused by the integer numeric format that handles the hardware platform which causes lost in the accuracy of embedded FLCs regarding their models tested over simulation.

Taking into account the comparison presented above, it is possible to consider that the IT2-FLC outperforms the T1-FLC in all cases. In addition, by looking Figure 11 and 12, it can be observed that the T2-FLC presents fewer oscillations around the reference, being the IT2-FLC implemented with centroid type reduction, those with better performance.

T2-FLC shows a good immunity to type reducer changes, which is a great advantage taking into account the reduction of computational complexity and inference time when strategies as CoS or heights are implemented.

6. Conclusions

An IT2-FLS for controlling the trajectory of a truck backer-upper mobile robot application and its embedded implementation has been presented in this chapter. The IT2-FLC has been designed based on the approach proposed in (Leottau & Melgarejo, 2010b) and implemented following methodological considerations reported in (Leottau & Melgarejo, 2010a) to the hardware implementation of IT2-FLS over DSC technology.

Two computational models have been selected taking into account the available and demanded hardware and computational resources. Several tests have been carried out in order to evaluate and to compare the performance of developed T2-FLCs and a T1-FLC. Simulated and emulated results evidence that the IT2-FLC is robust to type reducer and defuzzyfier changes and exhibits better performance than a T1-FLC when noise is added to inputs and outputs emulating some sources of uncertainty.

As future work, it is considered to test the developed IT2-FLC with a real mobile robot platform in order to carry out a comparative study of simulation, emulation and real platform performance. By this way, it is possible to extend the design and implementation methodology to other applications, involving in a more formal way a modelling of uncertainty.

7. References

Baturone, I.; Barriga, A.; Sanchez, S.; Jimenez, C.J. & Lopez, D.R. (2000). *Microelectronics Design of Fuzzy Logic-Based Systems*, CRC Press LLC, London.

Castro, J.; Castillo, O. & Melin, P. (2007). An Interval Type-2 Fuzzy Logic Toolbox for Control Applications, *Proceedings of the 2007 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2007,* London, UK.

Dorf, R. & Bishop, R. (1998). *Modern Control Systems*, Addison-Wesley.

Figueroa, J.; Posada, J.; Soriano, J.; Melgarejo, M. & Rojas, S. (2005). A Type-2 Fuzzy Logic Controller For Tracking Mobile Objects In The Context Of Robotic Soccer Game, *Proceedings of the 2005 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2005*.

Freescale Semiconductor. (January 2010). *Digital Signal Controller 56800/E Reference*. 29.01.2010, Available from: Http://www.Freescale.Com/Dsc

Frontline Systems. (March 2010). *Smooth Nonlinear Optimization*. 17.03.2010, Available from: http://www.solver.com/technology4.htm

Hagras, H. (2004). A Type-2 Fuzzy Logic Controller for Autonomous Mobile Robots, *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2004*, Budapest, Hungary.

Hagras, H. (2007). Type-2 FLC's: a New Generation of Fuzzy Controllers, *IEEE Computational Intelligence Magazine*, February 2007, pp. 30-43.

Hagras, H. (2008). *Developing a type-2 FLC through embedded type-1 FLCs,* Proceedings of the 2008 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008, Hong Kong.

Hitec Servo-Motors, *Servo-motors catalogue*. 05.08.2010, Available from: http://www.hitecrcd.com/products/servos/index.html

John, R. & Coupland, S. (2007). Type-2 Fuzzy Logic A Historical View, *IEEE Computational Intelligence Magazine*, 2,1, pp. 57-62.

Karnik, N. & Mendel, J. (2001). Centroid of a Type-2 Fuzzy Set, *Information Sciences*, 132, pp. 195-220.

Kuo, B. & Golnaraghi, F. (1996). *Automatic Control Systems*, Prentice Hall.

Leottau, L. & Melgarejo, M. (2010a). A Methodological Proposal for Implementing Interval Type-2 Fuzzy Processors over Digital Signal Controllers, *Journal of Applied Computer Science Methods*, v.2-1, June 2010, pp.61-81.

Leottau, L. & Melgarejo, M. (2010b). A Simple Approach for Designing a Type-2 Fuzzy Controller for a Mobile Robot Application, *Proceedings of the North American Fuzzy Information Processing Society's NAFIPS 2010*, Toronto, Canada, July 2010.

Nguyen, D. & Widrow, B. (1989). The Truck Backer-Upper: An Example of self-learning in Neural Networks. *Proceedings of the International Joint Conference in Neural Networks*, pp. II-357-363, June 1989.

Martinez, R.; Castillo, O. & Aguilar, L. (2008). Optimization with Genetic Algorithms of Interval Type-2 Fuzzy Logic controllers for an autonomous wheeled mobile robot: A comparison under different kinds of perturbations, *Proceedings of the 2008 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008.*

Melgarejo, M.; Garcia, A. & Pena-Reyes, C. (2004). Computational Model and architectural proposal for a hardware Type-2 Fuzzy System, *Proceedings of the 2nd IASTED conference on Neural Networks and Computational Intelligence*, Grindewald, Switzerland.

Melgarejo, M. & Pena-Reyes, C. A. (2007). Implementing Interval Type-2 Fuzzy Processors, *IEEE Computational Intelligence Magazine*, 2,1, pp. 63-71.

Melin, P. & Castillo, O. (2003). A new method for adaptive model-based control of non-linear plants using type-2 fuzzy logic and neural Networks, *Proceedings of the 2003 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2003,* St. Louis, MO.

Mendel, J. (2001). Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions, Prentice Hall, New Jersey.

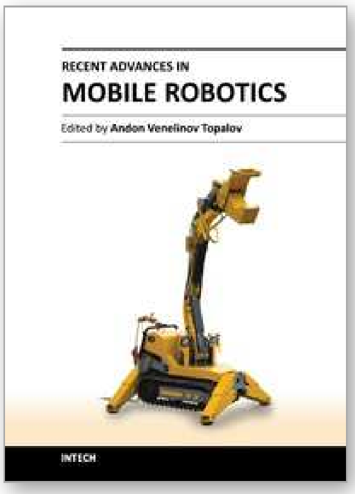
Mendel, J. (2007). Advances In Type-2 Fuzzy Sets and Systems, *Information Sciences*, 177,1, pp. 84-110.

Torres, P. & Saez, D. (2008). Type-2 fuzzy logic identification applied to the modelling of a robot hand, *Proceedings of the 2008 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2008.*

Wang, L. X. (1997). *A Course in Fuzzy Systems and Control*, Prentice Hall, New Jersey.

Wu, D. & Mendel, J. (2009). Enhanced Karnik-Mendel Algorithms, *IEEE Transactions on Fuzzy Systems*, 17, pp. 923-934.

Wu, D. & Tan, W. (2004). A type-2 fuzzy logic controller for the liquid-level process, Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2004.



Recent Advances in Mobile Robotics
Edited by Dr. Andon Topalov

ISBN 978-953-307-909-7
Hard cover, 452 pages
Publisher InTech
Published online 14, December, 2011
Published in print edition December, 2011

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821