Hybrid Intelligent Speed Control of Induction Machines Using Direct Torque Control

Fernando David Ramirez Figueroa and Alfredo Victor Mantilla Caeiros

Tecnologico de Monterrey, Campus Ciudad de Mexico,
Calle del Puente 222, Colonia Ejidos de Huipulco,
Tlalpan 14380, Distrito Federal, Mexico
{a00466442,amantill}@itesm.mx

Abstract. This paper presents a novel hybrid adaptive fuzzy controller for the regulation of speed on induction machines with direct torque control. The controller is based on a fuzzy system and PID control with decoupled gains. Genetic programming techniques are used for offline optimizations of the normalization constants of fuzzy membership function ranges. Fuzzy cluster means is introduced for online optimization on the limits of triangular fuzzy membership functions. Finally simulations in LabVIEW are presented validating the response of the controller with and without load on the machine; results and conclusions are discussed.

Keywords: Genetic Algorithms, Genetic Programing, Intelligent Control, Fuzzy Logic, Fuzzy C-means, FCM, Hybrid Intelligent Systems, DTC, Induction Machine.

1 Introduction

Induction machines are among the most widely used of all the electric motors. They are simple to build, rugged, and offer a reasonable asynchronous performance. Their torque-speed curve is controllable, they have a stable operation under load, and their efficiency is almost always satisfactory. Drives with high performance that control instantaneous electromagnetic torque (EMT) for induction motors (IM), have been in use for several decades. Direct torque control (DTC) was developed in the 1980's by Depenbrok [5], Takahashi and Noguchi [18].

Control techniques based on artificial intelligence have been very popular since the decade of the 1990's given that most of these techniques do not require complex mathematical models to be designed. Fuzzy controllers developed by Mamdani [12] and Takagi-Sugeno [17] are among the most popular systems. Other approaches consist of the use of Artificial Neural Networks [3, 14] to replace classical control regulators. Since the design of these controllers does not consider an optimal response and is normally based on human knowledge, genetic algorithms can be used to optimize response [7,10]. Numerous contributions have been made to DTC aided by Intelligent Control such as those presented in [4], [8], [9] and [13] among many others.

I. Batyrshin and G. Sidorov (Eds.): MICAI 2011, Part II, LNAI 7095, pp. 319–330, 2011.

Improvements proposed in [8] and [9] are made to the DTC control scheme by using AI techniques instead of the conventional DTC algorithm. The work made in [13] proposes an adaptive law based on fuzzy logic which adjusts the gain on the model and estimates speed.

In this paper a novel speed control for induction motors using direct torque control and intelligent control techniques based on fuzzy logic, evolutionary algorithms, and fuzzy cluster means is presented. The performance of the speed hybrid controller increases as different intelligent control techniques are combined.

The purpose of this work is to apply artificial intelligence techniques not previously used to motor control. Conventional DTC is used to control torque and flux in the machine and indirectly regulate speed. The Controller is based on fuzzy Mamdani systems and PID control. A non-interactive (decoupled gains) version of PID is presented and implemented; the fuzzy controller is initially designed based on human knowledge. Later, this controller is optimized using genetic programing; finally Fuzzy Cluster Means is used to enhance the online performance.

This paper is organized as follows: section 1 presents an introduction, section 2 describes the theory of the proposed controller, section 3 shows simulations, section 4 presents and discusses tests and results and finally section 5 describe conclusions.

2 Intelligent Speed Control of Induction Machines Using DTC

A technique to indirectly regulate the speed of IM is by regulating torque (diagram shown in Figure 1). DTC decouples the control of torque and flux in an induction machine, making it very similar to the control of a direct current machine [19]. The speed controller is highlighted with a block with the broadest line; it will regulate the electromagnetic torque reference of the DTC loop, hence, indirectly regulating the speed.

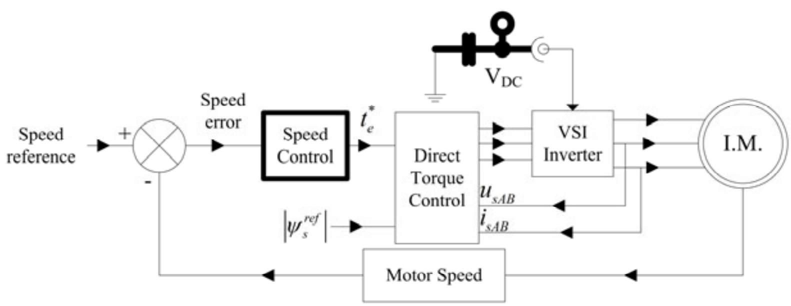

Fig. 1. Diagram of DTC for IM with speed loop using Intelligent Control

2.1 Basics of Direct Torque Control

In a DTC drive, flux linkage and electromagnetic torque are controlled directly and independently by selecting the optimum mode for the inverter. This selection is made to maintain flux and EMT errors within their respective hysteresis bands. This selection is also made to obtain a fast torque response, low inverter switching frequency, and low harmonics [5, 18]. The stator flux and EMT are restricted with their respective hysteresis bands, two and three levels respectively. The outputs of the comparators are used by the inverter switching table (Table 1), which also uses information of the sector of the stator flux space vector. In Table 1 the arrows indicate the increase or decrease of EMT or Flux, depending on the selected vector for action.

Table 1. Inverter switching table and their relation to flux and electromagnetic torque

	V_1	V_2	V_3	V_4	V_5	V_6	V_0 & V_7
$\lvert\psi\rvert$	↑↑	↑	↓	↓↓	↓	↑	---
EMT	↓	↑	↑	↓	↓↓	↓↓	↓

2.2 Mathematical Model of Induction Machine

The model of the induction machine is a dynamic vector model in a two axis DQ stationary reference frame, for a symmetrical squirrel cage induction machine. It is considered to be balanced and does not include the magnetic circuit model [15]. Here stator voltages are (1), rotor voltages (2). Flux linkages for stator and rotor are (3) and (4). EMT is described by (6) and finally stator, rotor and slip rotational speeds are related by (5).

$$v_{sDQ} = r_s i_{sDQ} + \frac{d\psi_{sDQ}}{dt} \tag{1}$$

$$v_{rDQ} = 0 = r_r i_{rDQ} + \frac{d\psi_{rDQ}}{dt} - \omega_r \psi_{rQD} \tag{2}$$

$$\psi_{sDQ} = L_s i_{sDQ} + L_m i_{rDQ} \tag{3}$$

$$\psi_{rDQ} = L_r i_{rDQ} + L_m i_{sDQ} \tag{4}$$

$$\omega_s = \omega_{slip} + \omega_r \tag{5}$$

$$T_{em} = 3P/2J\left(\psi_{sD} i_{sD} - \psi_{sQ} i_{sQ}\right) \tag{6}$$

2.3 Adaptable Genetically Enhanced Fuzzy PID Controller

This controller is based on a non-interactive (decoupled gains) improved version of the classical PID control law; it will considerably increase the performance of the original version [1]. The control law is implemented using (7) and the signal is sent to the plant with (8).

$$\dot{u} = k_i e + k_p \dot{e} + k_d \ddot{e} \tag{7}$$

$$u = \dot{u} + \Delta u \tag{8}$$

Each of the P, I, D gains are based in the Mamdani fuzzy system [12], shown in figure 2. The inputs and output of the controller are normalized to the [-2, 2] region and the outputs are scaled back to the operation range of the machine. Each fuzzy system (or gain) has three inputs, one output, triangular membership functions and three rules, generating a total of nine rules. The rules are shown in table 2: the general form of the rule is in (9).

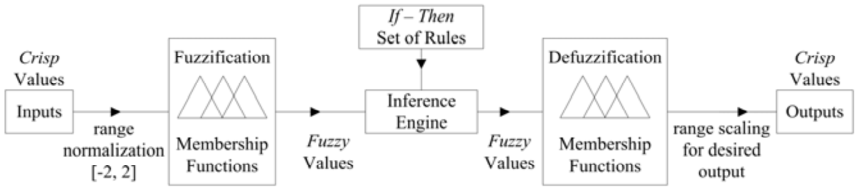

Fig. 2. Mamdani fuzzy inference system used for each of the PID gains in the control law

$$IF\ input\ is\ Inp_{MF}\ THEN\ output\ is\ Out_{MF}. \tag{9}$$

Where: *input* is a crisp value. *output*: is the crisp regulated value of EMT. Inp_{MF} or Out_{MF}: can be any of the membership functions (MFs) used in the fuzzification or defuzzification process. Once the response of each gain is calculated, the output is summed as in (7) and the control law sent to the plant is calculated with (8).

Table 2. Inputs, outputs and fuzzy rules of fuzzy PID controllers

Inputs	**Outputs for the three different gains**		
	Positive	**Zero**	**Negative**
Negative	Positive	Negative	Positive
Zero	Zero	Zero	Zero
Positive	Negative	Positive	Negative

$$Fitness = \frac{\sum \left(Speed_{ref} - Speed_{real}\right)^2}{\#Samples} \quad (10) \qquad \#Samples = \frac{TotalSimTime}{StepSimTime} \quad (11)$$

The fuzzy PID is genetically enhanced using GP in the input and output normalization constants... The fitness is measured with the mean squared error calculated with (10), which is the reference speed minus the real speed squared, all of these samples summed and divided by the number of samples. The number of samples is the inverse of the step simulation time multiplied by the total simulation time (11).

2.3.1 Genetic Programming as Optimization Techniques

Genetic Programing -GP- creates autonomous programs that evolve their structure; this technique provides a way for searching for the best or *fittest* program to solve a problem. GP genetically breeds a population of programs using Darwin's natural selection and biological inspired operators. These programs are represented by trees. In GP the tree representation eliminates the problem of fixed size chromosomes making the search in a more organized manner [2, 10]. It also makes the search more organized, thus, being able to solve more complex problems than genetic algorithms.

Genetic Programming Algorithm Steps

GP is very similar to Genetic Algorithms because it contains biologically inspired operators. However, the differences cover some of the weaknesses of original GAs, as will be further described. The stages of the algorithm are [10]:

- *1. Initialization*: Elements such as number of generations, population size, probability of crossing, and mutating individuals, are created and initialized.
- *2. Selection*: Selection is made by measuring the performance of individuals aiming to maximize the performance of the population. The performance of individuals is measured using a *Fitness Function*. This function is a certain task where each individual is evaluated [16]. The tournament selection method is used, whereby a few randomly selected individuals compete in several tournaments and the winner of each tournament is selected for crossover.
- *3. Crossover*: This operation mates individuals by combining segments of chromosomes. Branches on trees are interchanged, diversifying the size of individuals and generating new ones.
- *4. Mutation*: When the mutation operation is performed, one of the branches of the individual is mutated, generating a new form in the branch and a new individual.
- *5. End conditions*: The algorithm will cease to operate if conditions, such as iteration number or a certain fitness value, are fulfilled. In the contrary case, the algorithm returns to *selection* to continue with the evolutionary process.

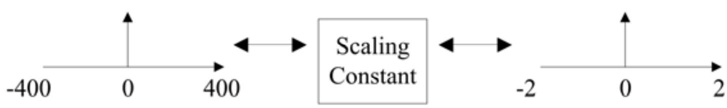

Fig. 3. Enhancements to the Fuzzy PID Controller

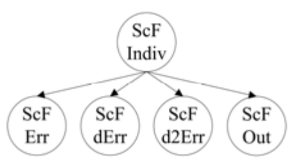

Fig. 4. Scaling factor optimization individual tree form

Normalization Factors Optimization

Normalization factors are very important since they dictate what portion of the decision table is used. They change membership functions uniformly over domains thus, changing the controllers gain over the whole domain uniformly. The scaling factors (Figure 3) are codified using 8 bit unsigned integers and later 16 bit unsigned integers to make the range of search wider.

The optimization process for scaling factors is done using a constant for each gain, along with different sizes for the number representation. The individual represented by a tree is shown in Figure 4. It is basically a tree with four branches; each one represents a scaling factor for the three inputs and one output.

2.3.2 Adjusting Membership Function for Optimal Response

The form of the membership functions is optimized online. Using the FCM algorithm the controller is able to adjust the form of its MFs online and with this information the limits of the triangular MFs are adjusted.

Fuzzy Cluster Means

Clustering methods split a set of N elements $X=\{x_1,\ldots,x_n\}$ into c groups or clusters $c=\{\mu^1,\ldots\mu^n\}$. Fuzzy set theory provides a natural way to describe clustering methods in a more realistic form using FCM. The FCM algorithm is described as follows: Fuzzy partition matrices M, for c classes and N data points are defined by three conditions $M\in\{U\in V_{cN}|(12)\}$:

$$\begin{aligned}&\forall\ 1\le i\le c \quad \mu_{ik}\in[0,1],\quad 1\le k\le N;\quad \sum_{k=1}^{c}\mu_{ik}=1 \quad \forall \quad 1\le k\le N\\&\forall \quad 1\le i\le c \quad 0<\sum_{K=1}^{c}\mu_{ik}<N\end{aligned} \tag{12}$$

Algorithm 1. Fuzzy Cluster Means algorithm

1. Fix x and m, set p = 0 and initialize $U^{(0)}$.
2. Calculate fuzzy centers for each cluster $V^{(p)}$. using (16)
3. Update fuzzy partition matrix $U^{(p)}$ using (15)
4. If $||U^{(p)}-U^{(p-1)}||<\in$ then $j=+1$ and return to step 2.

FCM (algorithm 1) will maximize the distance between the centers of the clusters and minimize the distances of the elements of a same cluster. The FCM criteria function is shown in (13). d_{ik} is the inner product norm -distance- (14). A is a positive definite matrix and m the weighted exponent: $m\in[1,\infty)$. By assigning values to m and c and defining the working sets, (U,V) can be a global minimum of $J_m(U,V)$ if (15) and (16) are fulfilled [16]. Parameter m determines the fuzziness of the clusters; if $m = 1$ the algorithm becomes the crisp k-means version, and $m = \infty$ the algorithm is as fuzzy as possible, usually $m = 2$ [16].

$$J_m=(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}\mu_{ik}^{m}d_{ik}^{2} \tag{13}$$

$$d_{ik}^2 = \|x_k - v_i\|_A^2 \tag{14}$$

$$\forall\ 1 \le i \le c\ \ 1 \le k \le N \quad u_{ik} = 1 \Big/ \sum_{j=1}^{c} \left(\|x_k - v_i\| / \|x_k - v_j\| \right)^{2/(m-1)} \tag{15}$$

$$\forall\ \ 1 \le i \le c \quad v_j = \sum_{k=1}^{N} u_{ik}^m x_k \Big/ \sum_{k=1}^{N} u_{ik}^m \tag{16}$$

The response of the controller can then be optimized depending on the input response and for that a simple algorithm is proposed and explained in Algorithm 2. Its execution can be graphically appreciated in figure 5. The form of the input MFs is varied only in their centers, so the internal action of the controller is not changed and erroneous response is induced. Nevertheless, adjusting the form of the output MFs in their entire limits will have a positive impact, since it will take action on the crisp output response of the controller.

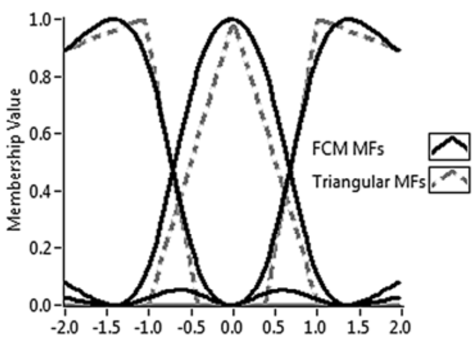

Fig. 5. Membership functions for the Adaptable Fuzzy PID Controller

Algorithm 2. Algorithm for the adaption of membership functions for optimization response

1. Generate a set of data in the range of action of the membership functions
2. Eliminate data from the initial set around a certain point
3. Adjust the form of the triangular membership functions using FCM

3 Simulation of the System

The implementation is made in LabVIEW [11] with the aid of the Intelligent Control Toolkit for LabVIEW [16]. For the PID controller the pre-optimization values are set as follows: the inputs are normalized by dividing them by 377, the top speed of the rotor [rad/s] and the output scaled back with a constant of 200. The values for these constants are set empirically based upon knowledge from the user; the simulation program is shown in Figure 6. The program that makes the optimizations using GP is shown in figure 7. The fitness function used is a simulation of the induction motor model with the DTC control scheme and the fuzzy PID loop controlling the speed, similar to the program in figure 6.

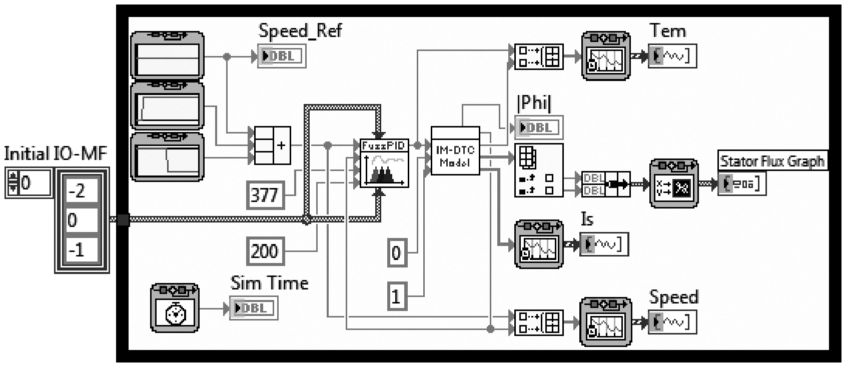

Fig. 6. DTC and IM loop with Fuzzy PID Controller in LabVIEW

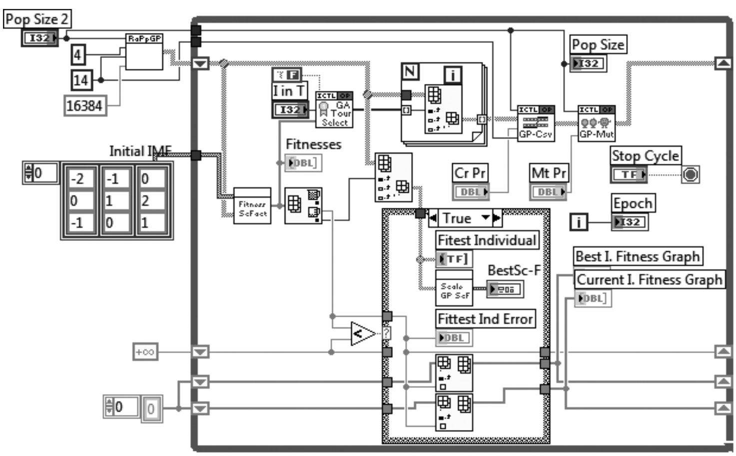

Fig. 7. of Genetic Programing block diagram system made in LabVIEW

4 Results

Parameters consulted in [15] for the induction motor model used in the simulations are shown in Table 4. Simulations have different speed references, as it is shown in Table 3, for a total of 10 s of simulation time. The solver used is Euler with a fixed step time of 50 µs. The GP optimizations had a fitness function with a period of time of 10 seconds, a step time of 0.2 ms and the Euler solver. Results are compared against a conventional PID with proportional gain of 1, integral gain of 10000 and derivative gain of 10000.

When the fuzzy controller is executed with the initial values prior to optimization the fitness value is 24674. Results of the GP are shown in Table 5; the fittest individual found had a fitness value of 17119.8. Figure 9 shows the results of this individual compared to a classical PID for the test presented in Table 3. The optimized controller (IC) takes the machine to the desired reference faster and more accurately than the classic PID (C-PID); thus showing that the response of the optimized controller is much better. These results show that GP is able to successfully optimize the overall response of the controller.

Table 3. Speed references test

Time Range [s]	Speed [rad/s]
[0, 1]	50
[1, 5]	300
[5, 10]	100

Table 4. Parameters for motor simulations where *p:* parameter and *v:* value

P	V	P	V	P	V	P	V
Rs	0.435 Ω	*Lm*	0.06931 H	σ	0.05531	*f*	60 Hz
Rr	0.816 Ω	*Ls, Lr*	0.07131 H	*Tm*	1 Nm	*Poles*	2

Table 5. Scaling factors optimization results using GP. *# Bits:* Bits used to represent the search space. *Pop Size:* Size of the population. *I in T:* Individuals used in tournament selection. *M Pr:* Probability of mutation. *Epochs:* Generations executed. *Fitness:* Fitness of the best individual found during that execution. *ScF Err:* Scaling Factor for the error. *ScF dErr:* Scaling Factor for the first derivate of the error. *ScF d2Err:* Scaling Factor for the second derivate of the error. ScF Out: Scaling Factor for the output. *Cross Probability* is 0.9 for all the executions.

# Bits	Pop Size	I in T	M Pr	Epochs	Fitness	ScF Err	ScF dErr	ScF d2Err	ScF Out
10	12	3	0.02	147	21399.71	266	0.0022	424	988
10	12	3	0.035	352	20655.92	585	0.0020	0.0020	988
10	16	4	0.025	60	21399.7	195	0.0025	214	834
13	24	6	0.01	35	19261.2	7182	0.0004	0.0003	5168
13	***14***	***4***	***0.015***	***82***	***17119.8***	***512***	***84***	***0.0002***	***8144***
13	14	4	0.015	194	19247.7	6114	0.0003	0.0005	1822
16	24	6	0.01	206	19250.93	55378	3.3E-5	5.7E-4	54277

Once GP optimizations were executed the adaptable algorithm on the controller was allowed to operate and be tested. The adjusted membership functions are shown in Figure 8, the left shows the input MFs and the right the output MFs. The shapes of the input MFs are optimized with the information of the fuzzy centers coming from FCM, the beginning and ending limits of the zero cluster.

The limits of the output MFs are also optimized with the centers coming from FCM, the ending limit of negative cluster, beginning and ending of zero cluster, and beginning of the positive cluster. The results for the test presented in Table 3 are shown in Figure 9; as it can be seen the response of the controller is optimal, being able to set the speed of the motor to the desired reference and update the limits of its membership functions on the fly.

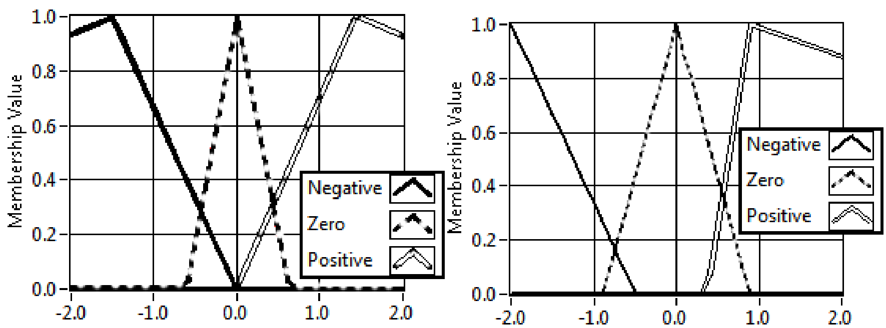

Fig. 8. Triangular membership functions adjusted with adaptable MF algorithm

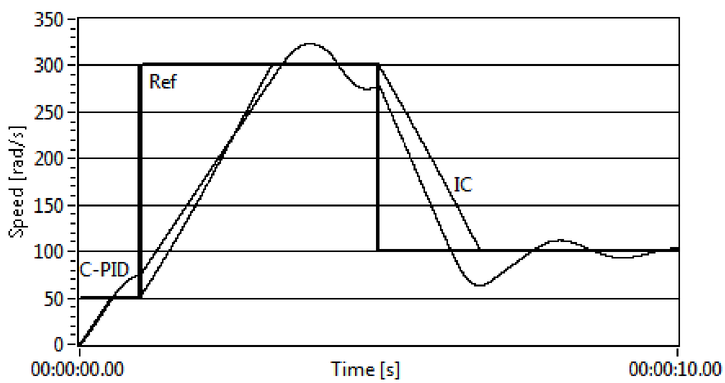

Fig. 9. Speed test for Adjustable fuzzy controller without load

Although the response of the controller has been optimized successfully under no load to the machine, it is interesting to analyze what would happen if load is introduced to the motor. For that reason a simple test is shown in Table 6; after the controller has reached a desired speed, different loads are introduced to the machine.

Table 6. Speed references test with load

Time Range [s]	Torque Load [Nm]
[4.2, 5.5]	50
[5.5, 9]	80

As can be appreciated in Figure 10, when the first load of 50Nm is introduced, the adjustable controller response does not change, while the response of the non-adjustable controller drops, but remains constant.

When the 80Nm load is set the response of all the controllers drops; however, after some moments the adjustable controller is able to compensate and the response of the machine increases. The non-adjustable controller is not able to compensate and the response goes down until the load is withdrawn. After the load is withdrawn both controllers compensate and correct the response. A small steady state error can be seen in the normal fuzzy controller, while the optimized controller does not show the same steady state error. Finally, the response of the classic PID (C-PID) is of the chart limits, with very high oscillations and not being capable of compensating the load.

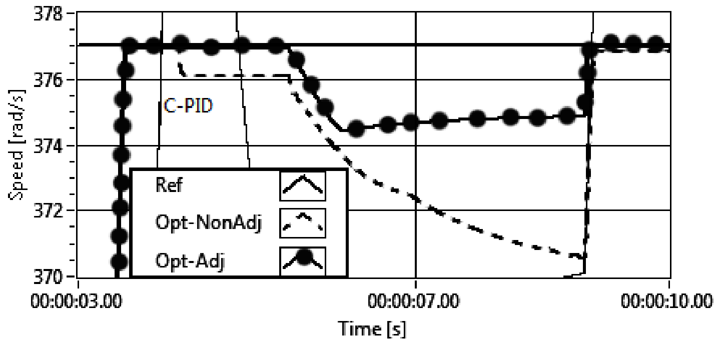

Fig. 10. Speed test for Adjustable fuzzy controller with load

5 Conclusions

This work presents a novel approach on intelligent control for the speed control on induction machines validated through simulations. Most of these applications require complex computational capabilities, which are getting easier to obtain at lower costs. The proposed PID decoupled topology was validated and can be easily transferable to an embedded system. Difference gives more information because they represent a reason of change. Digital systems saturate integrators and differences can be easily computed with subtractions.

The optimizations based on GP proved to effectively optimize the fuzzy controllers. GP can deal with complex forms of optimizations; these optimizations have to be carried offline due to the nature of evolutionary algorithms and their delayed response due to large search spaces.

The online optimization proposed proved to correctly compensate unknown uncertainties introduced to the system and further improve the response of the controllers. Furthermore, this optimization can be executed online without requiring great computational load, as it is executed with a 1 ms period, much greater to the smaller -lower than 25 µs- required executing DTC [6].

References

1. Astrom, K., Hagglund, T.: PID Controllers: Theory, Design and Tuning. Instrument Society of America, USA (1995)
2. Bourmistrova, A., Khantis, S.: Control system design optimisation via genetic programming. In: IEEE Congress on Evolutionary Computation, CEC 2007, Singapore, pp. 1993–2000 (2007)
3. Cheng-Zhi, C., Guang-Hua, W., Qi-Dong, Z., Xin, W.: Optimization Design of Fuzzy Neural Network Controller in Direct Torque Control System. In: Third International Conference on Machine Learning and Cybemetics, Shanghai, vol. 1, pp. 378–382 (2004)
4. Stephen, C.: Developing commercial applications of intelligent control. IEEE Control Systems Magazine 17(2), 94–100 (1997)
5. Depenbrock, M.: Direkte Selbtregelung (DSR) für hochdynamische Drehfeldantribe mit Stromrichterschaltung. ETZ A 7, 211–218 (1985)
6. Dufoo, S., Pacas, M.: Predictive Direct Torque Control of an Induction Machine with Unsymmetrical Rotor. In: IEEE International Conference on Industrial Technology, pp. 1851–1856 (2010)
7. Goldbergh, D.E.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Massachusetts (1989)
8. Grabowski, P.Z., Blaabjerg, F.: Direct Torque Neuro-Fuzzy Control of Induction Motor Drive. In: 23rd International Conference on Industrial Electronics, Control and Instrumentation, vol. 2, pp. 557–562 (1998)
9. Li, H.: Fuzzy DTC for Induction Motor with Optimized Command Stator Flux. In: 8th World Congress on Intelligent Control and Automation, Jinan, China, pp. 4958–4961 (2010)
10. Koza John, R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
11. LabVIEW, graphical programming language, `http://www.ni.com/labview`
12. Mamdani, E.H.: Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. Institution of Electrical Engineers, Control & Sciences 121(12), 1585–1588 (1974)
13. Sui, M., Zhang, K., Yang, J.: An Improved Sensorless DSVM-DTC of Induction Motor Based MRAFC. In: 7th World Congress on Intelligent Control and Automation, pp. 775–780 (2008)
14. Pedro., P.C., Rivas, J.J.R.: A Small Neural Network Structure Application in Speed Estimation of an Induction Motor Using Direct Torque Control. In: IEEE 32nd Annual Specialists Conference on Power Electronic, vol. 2, pp. 823–827 (2001)
15. Pedro, P.C., Javier, S.: Maquinas Electricas y Tecnicas Modernas de Control. Grupo, A. (ed.), Mexico (2008)
16. Pedro, P.-C., Fernando, D.: Ramirez-Figueroa. In: Intelligent Control Systems with LabVIEW. Springer, London (2009)
17. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 15, 116–132 (1985)
18. Takahashi, I., Noguchi, T.: Quick torque response control of an induction motor using a new concept. IEEE J. Tech. Meeting on Rotating Machines, paper RM84-76, 61–70 (1984)
19. Peter, V.: Sensorless Vector and Direct Torque Control. Oxford University Press (2003)