A SIMULATION-IMPLEMENTATION METHODOLOGY OF A FUZZY LOGIC BASED CONTROL SYSTEM

Gilberto C. D. Sousa
Departamento de Engenharia Elétrica
Universidade Federal do Espírito Santo - UFES
29060-970 Vitória, ES

Bimal K. Bose
Electrical and Computer Engineering Department
The University of Tennessee, Knoxville
Knoxville, TN 37996-2100 - EUA

Marcelo G. Simões
Escola Politécnica da Universidade de São Paulo
PMC / Mecatrônica
Av. Prof. Mello Moraes, 2231
05508-900 São Paulo - SP

Abstract: **The paper presents a methodology for simulation study and real time implementation of a fuzzy logic based controller of a power electronic system. The fuzzy controller with the membership functions and rule base is developed and programmed for simulation in a high level language (such as SIMNON) in the same environment as the plant and other non-fuzzy controller simulation, and then extensively iterated until optimum performance is obtained. The optimized controller is then directly translated into assembly (or C) language and integrated with the other controllers for real time implementation. The methodology is illustrated in detail with an actual control system.**

I. INTRODUCTION

The application of fuzzy logic is increasing steadily in the control of power electronics and drive systems [1]. In fact, many power electronic systems possess non-linearities, time constants varying from a few micro-seconds to several seconds, or are not easily modeled by conventional control techniques. Therefore, they can benefit from the usage of fuzzy logic in the design of their controllers, since fuzzy logic allows the inclusion of intuitive knowledge in the control strategy, usually resulting in better performance than that obtainable with conventional control techniques.

In spite of the success of fuzzy control in the power electronics area, a few drawbacks still remain. While in linear control design the performance, including stability indices, can be predicted theoretically, fuzzy controller design still heavily relies on trial-and-error approach for fine-tuning the membership functions and rule base, although fuzzy neural networks and genetic algorithm techniques are being developed to help with the design procedure [12] [13]. Furthermore, a typical control system requires signal acquisition and conditioning, that usually involves linear filters. Again, due to the ease of implementation, other classical controllers might be present as well within the system, to implement simpler control functions that do not require the performance of a fuzzy controller. Consequently, extensive simulation study of the controller in the integrated environment of plant and its fuzzy controller, plus any additional non-fuzzy controllers that might be present, is extremely important.

Recently, a large number of fuzzy logic development tools have appeared in the market. Most of these tools permit easy and user-friendly development of fuzzy controller in C language, which is then compiled to generate the object code for real time control. The popular Togai TILSHELL, for example, permits development of fuzzy production rules and membership functions in user-friendly Fuzzy Programming Language (FPL). The FPL source code is then compiled into portable C source code which is then further compiled to target-specific object code for real time control. The controller can be iterated in the TILSHELL, but its performance needs to be tested in the actual system. If the plant is simulated in C language, then the C-based fuzzy controller can be integrated with the plant simulation and then iterated easily. Power electronic systems, particularly containing AC machines, are so complex that it is not easy to develop C-based simulation program. On the other hand, languages such as SIMNON, ACSL, etc. are particularly convenient for such simulation. Recently, a new version of MATLAB was released that supports a Fuzzy Logic Control Toolbox. While providing an adequate environment for modeling fuzzy controllers, the cost of such a system is still much higher than that of simulation languages such as SIMNON.

Unlike the traditional approach of fuzzy controller development and then iteration on the actual system, in this project we have proposed SIMNON simulation of the total system. The fuzzy controller is easily programmed in SIMNON, or any other system simulation tool, and iterated extensively until the system performance is optimum. Then, the controller is easily translated into time-critical assembly language of a digital signal processor (DSP), the TMS320C25 from Texas Instruments, Inc., integrated with other non-fuzzy controllers, and finally tested on a drive system to verify the performance [7].

II. MODELING OF THE FUZZY LOGIC CONTROLLER

Fuzzy logic control literature is quite comprehensive [3]-[5]. A number of inference methods have been proposed through the years. Nevertheless, the Sup-Min (Supremum-Minimum) composition method or its variants are dominant. A brief review of its structure will be attempted here for the sake of completeness.

A.. *Structure of the Fuzzy Logic Controller*

The general structure of a fuzzy control system is given in fig. 1, while the inference method is illustrated with the help of fig. 2.

The control signal (u) is inferred from the two input variables error (e) and change in error (ce), usually expressed as per unit (pu) signals. In a strict sense, the fuzzy controller is designed to process fuzzy quantities only. Therefore, all crisp input values must be converted to fuzzy sets before being used. This process is called fuzzification operation, and can be performed by considering the crisp input values as "singletons" (fuzzy sets that have membership value of 1 for the given input value and 0 for all other points in the universe of discourse), as shown in fig. 2. They are next compared to the fuzzy sets to determine the respective degree of membership (μ). Once the antecedent (IF part) of every rule has been evaluated, the consequent (THEN part) can be computed by SUP-MIN composition. The process is illustrated in fig. 2 for the case of two rules. Notice that the output membership function of each rule is given by the MIN operator, while the combined fuzzy output is given by the SUP operator.

The result of the fuzzy inference is a fuzzy set, and consequently, there is a need for converting the output of the fuzzy controller to a crisp (real) value required by the plant. This is called defuzzification operation, and can be performed by a number of methods of which the center of gravity and height methods are the most common ones [4]. While the center of gravity method is the most frequent in the literature, it requires the integration of two functions, and consequently, its real time implementation is time consuming. On the other hand, the height method (described later in eq. 5) is very simple to implement, and is selected for this application.

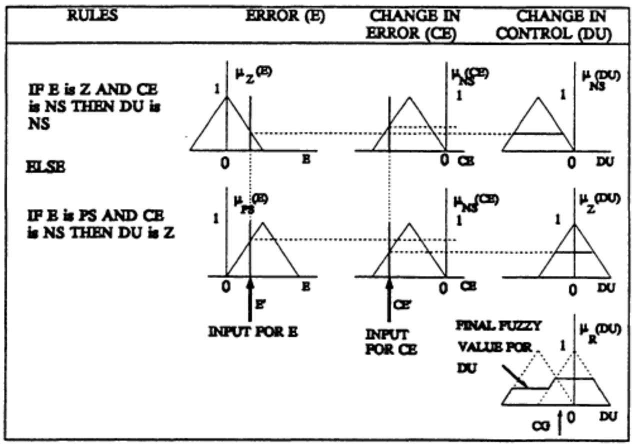

Fig. 2: Fuzzy composition by SUP-MIN principle.

B. *The Rule Base Evaluation Methodology*

Rule base evaluation is the core of the fuzzy inference process. It will be described here with the help of fig. 2 and fig. 3, where the most used triangular membership functions are selected, but the method can be easily extended to other types.

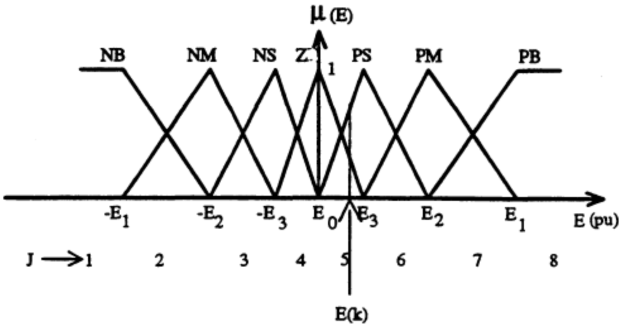

Fig. 3: Degree of membership evaluation.

Initially, the degree of membership of the input variables

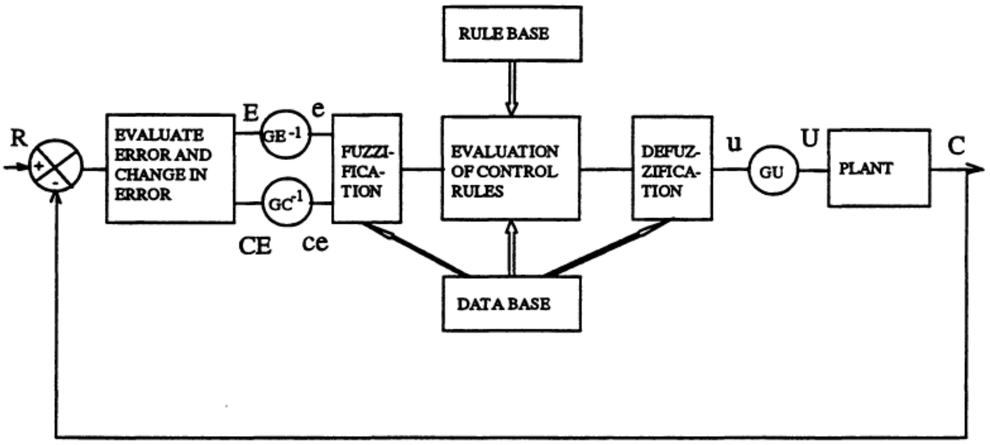

Fig. 1: Basic structure of a fuzzy controlled system.

with respect to each membership function should be obtained. From fig. 3, that represents the vast majority of practical applications of fuzzy controllers, it is evident that, at most, two membership functions will be affected; the degree of membership with respect to the remaining membership functions will be zero. Unnecessary computations can be avoided by first defining an interval index (J), that locates the relevant fuzzy sets (J=5 in fig. 3). The degree of membership of E(k) in the fuzzy set positive small (PS) is given by:

$$\mu_{PS}(E(k))=\frac{E(k)-E_0}{E_3-E_0} \qquad (1)$$

It is clear that for J=5, $E_0 \leq E(k) < E_3$, and the value of $\mu_{PS}(E(k))$ will fall in the [0,1] interval as expected. For any other interval, eq. 1 is rewritten to incorporate the proper constants (E_0 and E_3 in this case).

In the case of a 50% overlap between membership functions, the degree of membership for the other relevant membership function (Z in this example) is simply given by:

$$\mu_Z(E(k))=1-\mu_{PS}(E(k)) \qquad (2)$$

It is worth noticing that, by taking advantage of the particularities of the problem, great simplification in the rule base evaluation process can be achieved, with attendant improvement in computation speed.

Next, the implementation of the SUP-MIN inference method requires the evaluation of the antecedent of the relevant rules, using the MIN operator, as illustrated in fig. 2. For the case of two input variables, at most four rules will be relevant, i.e., will result in non-zero contribution to the control action. Let μ_{E1} and μ_{E2} be the two non-zero degrees of membership for E(k) (E_1=Z and E_2=PS in fig. 3). Conversely, let μ_{CE1} and μ_{CE2} be the two non-zero degrees of membership for a typical second fuzzy variable , change in error (CE(k)). The truth value for the first of the four fired rules will be given by:

$$\mu_{RA} = \mathrm{MIN}(\mu_{E1},\mu_{CE1}) \qquad (3)$$

The contribution of each fired rule must be retrieved from the rule base. In most practical cases of real time control implementation, the rule base is previously stored in the form of a simple look-up table, i.e., a singleton representation for the control variable is employed. In this case, the individual control action can be conveniently retrieved with the use of the interval indices previously computed. When the control variable is represented by piece-wise linear membership functions, the process will simply require the use of multiple look-up tables, but remains unchanged in its essence. The process is completed by performing the defuzzification operation, that is normally a discretized version of the center-of-gravity method.

The next two sections illustrate the methodology just described. Initially the simulation methodology will be presented, followed by a discussion of its real time implementation. Again, the methodology can be easily applied to any other system simulation software, as well as any real time programming language.

III. SIMNON PROGRAM DEVELOPMENT FOR FUZZY LOGIC CONTROLLERS

SIMNON is a very versatile system simulation tool, and has been successfully applied to converters and drive systems simulations [9]. A detailed description will be made here of its use in a case study, the optimum efficiency control. The basic features of the problem will be reviewed next.

A. The Fuzzy Logic Based Efficiency Optimization Control Problem

The induction motor vector control system with fuzzy logic efficiency optimization control is shown in fig. 4. It is a speed control system where the torque component of stator current reference ($i_{qs}^{*'}$) is obtained from the speed PI (proportional-integral) regulator. The flux component of stator current reference (i_{ds}^{*}) is derived either from the rated value (i_{dsr}) for transient operation (position 1 of the switch) or from the efficiency optimization process (position 2 of the switch). The synchronous frame reference currents (i_{qs}^{*} and i_{ds}^{*}) are initially transformed to the stationary reference frame currents (i_{qs}^{s*} and i_{ds}^{s*}) with the help of unit vectors ($\sin\theta_e$ and $\cos\theta_e$), and next to the abc frame to generate the actual reference currents (i_a^{*}, i_b^{*} and i_c^{*}) to be impressed to the induction motor by the converter and its current control loop.

As the indirect vector control method is being applied, the slip frequency (ω_{sl}) is derived from the torque current reference (i_{qs}^{*}) in a feed-forward manner. However, in this case the rotor flux is not constant, and an open loop observer is used to derive its estimate (ψ_r) that, combined with estimates of machine magnetizing inductance (L_m), rotor resistance and inductance (R_r and L_r) constitute the slip gain. The slip gain multiplied by the i_{qs}^{*} results in ω_{sl}. This is next added to the actual machine speed (ω_r) and then integrated to generate the θ_e angle that relates the stationary and synchronous rotating reference frames.

It is well known that the optimum efficiency operation of a drive system can be achieved by proper flux control. In fact, particularly under light load condition, and high speed, the rated flux level results in excessive core losses and lower efficiency. By properly adjusting (usually reducing) the flux level a better partition of machine losses between core and copper losses can be achieved, ensuing optimum efficiency. The optimum efficiency condition, however, varies with load torque and speed of operation, and is also sensitive to parameter variation due to temperature and saturation effects. As a result, methods that resort to pre-computed flux levels for a given operating condition yield sub-optimum efficiency operation only.

Unfortunately, changing the flux level also results in torque disturbances, that is unacceptable in high performance drives. To overcome this problem, a feed-forward torque compensation scheme has been added to provide a correcting term to the original $i_{qs}^{*'}$ component, such that virtually no torque pulsation occurs. The problem has been addressed in

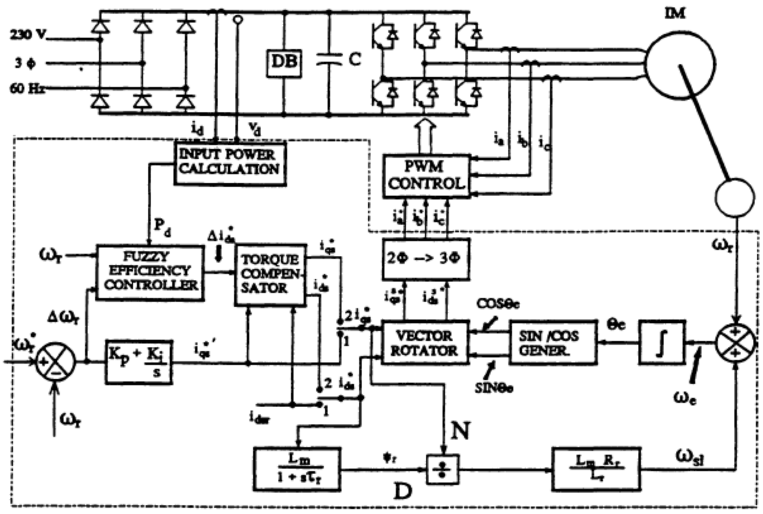

Fig. 4: Indirect vector control system incorporating fuzzy logic efficiency optimization control.

detail in an earlier paper [7], and only the fuzzy logic features will be stressed here.

Fuzzy logic based on-line search on the basis of measured input power is a powerful method to overcome these difficulties [7]. In fact, with closed-loop speed operation, the output power delivered to the load is constant. By monitoring the input power variation due to changes in the flux level, the system can be gradually driven to the optimum efficiency point. No model of the system is used, and as a result, no parameter sensitivity problem is present. The required knowledge to construct the fuzzy controller is quite simple: starting from an initial change in flux level, observe the variation in input power. If a reduction has occurred, proceed in the same direction; otherwise, reverse the search direction. Fuzzy logic provides the necessary framework to translate it to meaningful control action.

The proposed fuzzy efficiency controller shown in fig. 4 is illustrated in more details in fig. 5. The control signal, namely the step in d-axis current ($\Delta i_{ds}^*(k)$) is derived from the past control action ($L\Delta i_{ds}^*(k)=\Delta i_{ds}^*(k-1)$) and the observed variation in input power ($\Delta P_d(k)$), through fuzzy inference.

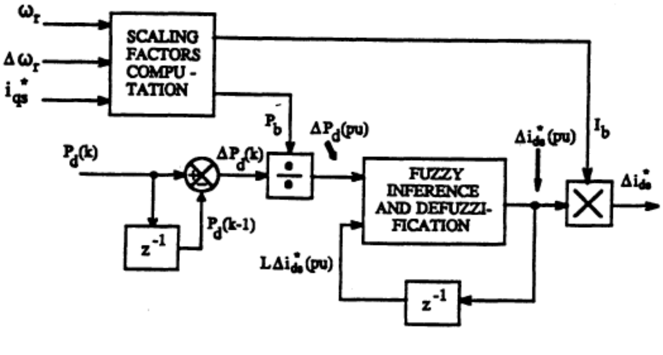

Fig. 5: Efficiency optimization control block diagram

The membership functions for the control variables are shown in fig. 6, whereas the universes of discourse have been normalized in the [-1,1] interval, through the use of suitable scaling factors (P_b and I_b). The latter are constructed from i_{qs}^* and ω_r signals, to help maintain optimum performance of the fuzzy efficiency controller in the entire operating range.

The rule base for the control action is listed in Table I. It essentially controls the convergence process by decreasing the step size $\Delta i_{ds}^*(k)$ when it detects that the system is reaching the optimum flux level, i.e., the input power reduction approaches zero. By using fuzzy logic, the search process can be speed up, without causing an overshoot. Consequently, as the optimum efficiency point is achieved faster, after a steady state condition has been detected, more energy can be saved than with conventional control methods.

B. Fuzzy Inference Simulation Methodology

The use of SIMNON language [10] in FLC implementation is somewhat complicated, mainly because SIMNON does not support branch instructions. Furthermore, the order of program execution is not completely controlled by the user, since during compilation the program sorts the instructions to optimize for execution time. However, the need to have all system components in a single simulation environment fully justifies the extra effort. Furthermore, once the basic fuzzy operations have been modeled, they can be used as macros in the development of more elaborated fuzzy controllers.

The methodology used in the actual derivation of fuzzy efficiency optimization SIMNON routine is explained here, with the help of fig. 6.

1. Compute input and output scaling factors, P_b and I_b, respectively, and the change in input power, ΔP_d (pu);

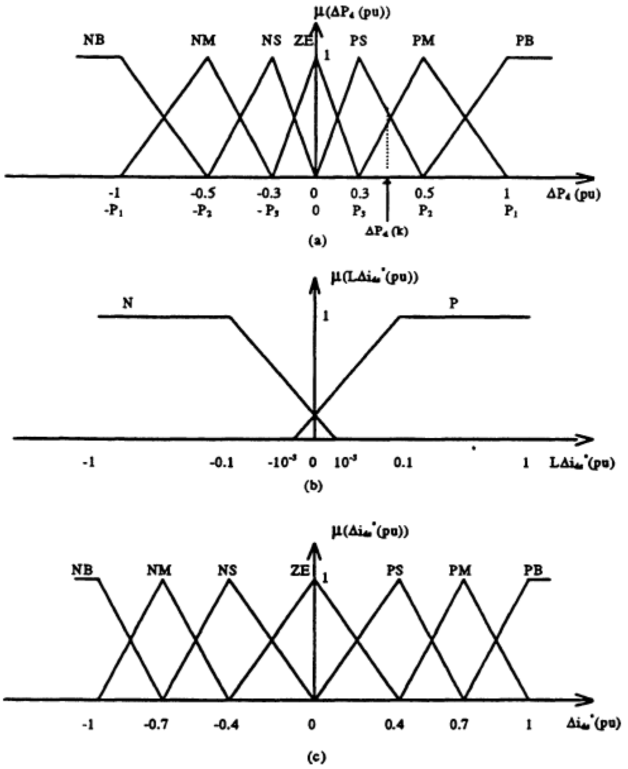

Fig. 6: Membership function for the fuzzy efficiency controller.

Table 1: Rule base for the FL controller.

$L\Delta i_{ds}$(pu) / ΔP_d(pu)	N	P
PB	PM	NM
PM	PS	NS
PS	PS	NS
ZE	ZE	ZE
NS	NS	PS
NM	NM	PM
NB	NB	PB

2. Make a preliminary computation of degree of membership for all fuzzy sets. For instance, the degree of membership of ΔP_d (k) in the fuzzy set PM, is given by:

$$\mu_{PM}(\Delta P_d(k)) = \frac{\Delta P_d(k) - P_3}{P_2 - P_3} \qquad (4)$$

3. Define the interval index J that identifies which fuzzy sets possess non zero degree of membership. SIMNON supports the conditional assignment of value to a variable, but does not allow a continuation line. This problem can be solved by defining some auxiliary variables (J_1, J_2,...) as indicated below:

   J=If ΔP_d<-P_1 Then 1 Else If ΔP_d<-P_2 Then 2 Else J_1

   J_1=If ΔP_d<-P_3 Then 3 Else If ΔP_d< 0 Then 4 Else J_2

4. Evaluate the degree of membership for the relevant fuzzy sets:

   μ_{P1} = If J<3 Then μ_{NB} Else If J<4 Then μ_{NM} ....Else μ_{PB}

   $\mu_{P2} = 1 - \mu_{P1}$

   For the situation illustrated in fig. 6, $\mu_{P1} = \mu_{PS}$ and $\mu_{P2} = \mu_{PM}$

5. Apply similar procedure to compute degrees of membership μ_{I1} and μ_{I2}, for the second input variable, $L\Delta i_{ds}^*$(pu).
6. Evaluate the antecedent of the relevant rules, using MIN operator. Typically, four rules are fired. The first rule is illustrated here:

   μ_{RA} = MIN (μ_{P1}, μ_{I1})

7. Retrieve the contribution of each fired rule, from the rule base:

   ΔI_a = If J<3 Then -I_1 Else If J<4 Then -I_2 Else...

   Due to the symmetry of this particular rule base, ΔI_b=-ΔI_a.

8. Calculate the new Δi_{ds}^* by height defuzzification method

$$\Delta i_{ds}^*(pu) = \frac{\mu_{RA}\,\Delta I_a + \ldots + \mu_{RD}\,\Delta I_d}{\mu_{RA} + \ldots + \mu_{RD}} \qquad (5)$$

After the development and tests of the SIMNON routines, a detailed performance evaluation study was conducted, for a 10 hp induction machine drive system. Fig. 7 shows the time domain optimum efficiency search curves at speed ω_r=0.25 pu and load torque T_L=0.1 pu. The optimum flux level is achieved in just four steps of i_{ds}^*. The search algorithm, however, imposes a minimum non-zero step size of Δi_{ds}^*, when ΔP_d approaches zero, in order to ensure true optimum operation for parameter variation or small change in load condition.

IV. REAL TIME FUZZY LOGIC CONTROL PROGRAM DEVELOPMENT

The complete drive control system, including the fuzzy efficiency controller, feed-forward torque compensation, speed control, and all input/output signals, was implemented in assembly language in a single TMS320C25 DSP based control board from Ariel Corporation. The real time fuzzy efficiency controller uses the same algorithm described earlier. Consequently, only some implementation aspects will be discussed here.

The algorithm basically derives the next control action (Δi_{ds}^*(k)) from the measured change in input power (ΔP_d(k)) and the last control action ($L\Delta i_{ds}^*$(k)). Initially, the interval index for each input variable is determined, thus defining the membership functions with non-zero degree of membership, as previously indicated in fig. 6, for the case of ΔP_d.

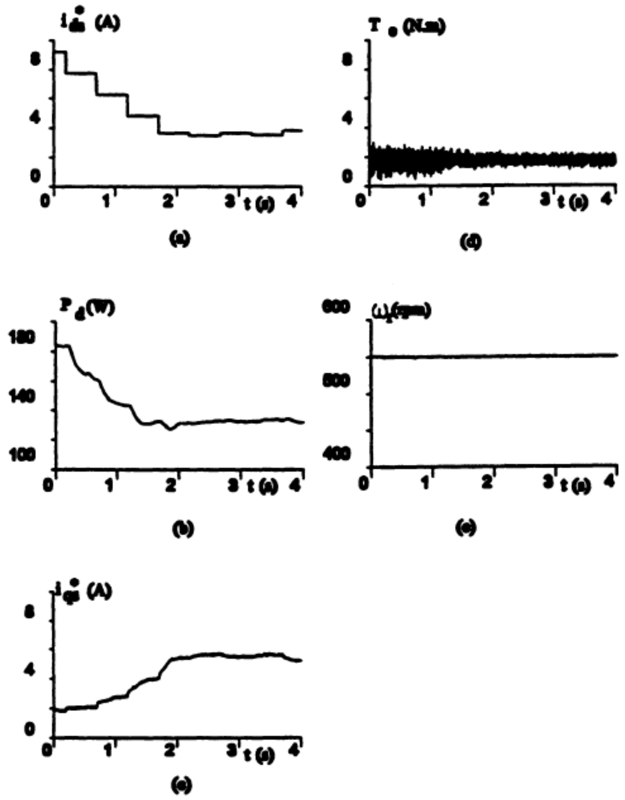

Fig. 7: Time domain simulated optimum efficiency curves at ω_r =0.25 pu and T_L =0.1 pu.

Since the DSP does not have a division instruction, but very efficiently supports a virtually single cycle multiplication instruction, whenever possible a division operation is replaced by a multiplication. This approach is used in the computation of the degrees of membership. For the particular value of $\Delta P_d(k)$ shown in fig. 6, we have:

$$\mu_{PM}(\Delta P_d)=(\Delta P_d(k)-P_3)(P_2-P_3)^{-1} \quad (6)$$

As P_2 and P_3 are constants, $(P_2\text{-}P_3)^{-1}$ was pre-computed and stored in data memory. Similar procedure was applied to all terms involved in the computation of the degree of membership for the remaining fuzzy sets. By doing so, the only true division operation necessary in the entire fuzzy efficiency controller was the defuzzification operation described in eq. 5.

The rule base evaluation requires the retrieval of the control signal associated with every fired rule. This task is readily accomplished with the use of interval indices and a memory resident look-up table containing the rule base. Let J be the index for ΔP_d , I the index for $L\Delta i_{ds}^*$, M and N the number of fuzzy sets for ΔP_d and $L\Delta i_{ds}^*$, respectively. It can be demonstrated that, for a two-dimensional rule base, the relative position (POS) of the first fired rule within the look up table is given by

$$POS=(I-1)M+(J-2) \qquad (7)$$

Similarly, the second rule is located at POS+M, the third one at POS+1, and the last one at POS+M+1. Once the rule base evaluation has been performed, the new control action can be computed by the height defuzzification method of eq. 5.

The implementation in Assembly language ensures optimized execution of the program, and consequently all the required control functions were implemented in a single DSP. Furthermore, as the language supports branch instruction, the identification of the relevant fuzzy sets is greatly simplified. By utilizing a functional description of the membership functions, the degrees of membership are precisely calculated, yielding finely adjusted control actions, in contrast with the discretization of the membership function approach, commonly reported in the literature [3], that tends to cause oscillations near the steady-state condition.

The operation of the fuzzy efficiency controller is illustrated in fig. 8. The drive system was initially in a steady-state condition, at a speed ω_r = 450 rpm (0.25 pu) and load torque T_L = 1.54 lbf ft (0.10 pu). At t = 0 the efficiency controller was enabled, and the magnetizing current command i_{ds}^* was adaptively reduced on the basis of measured DC link power as shown.

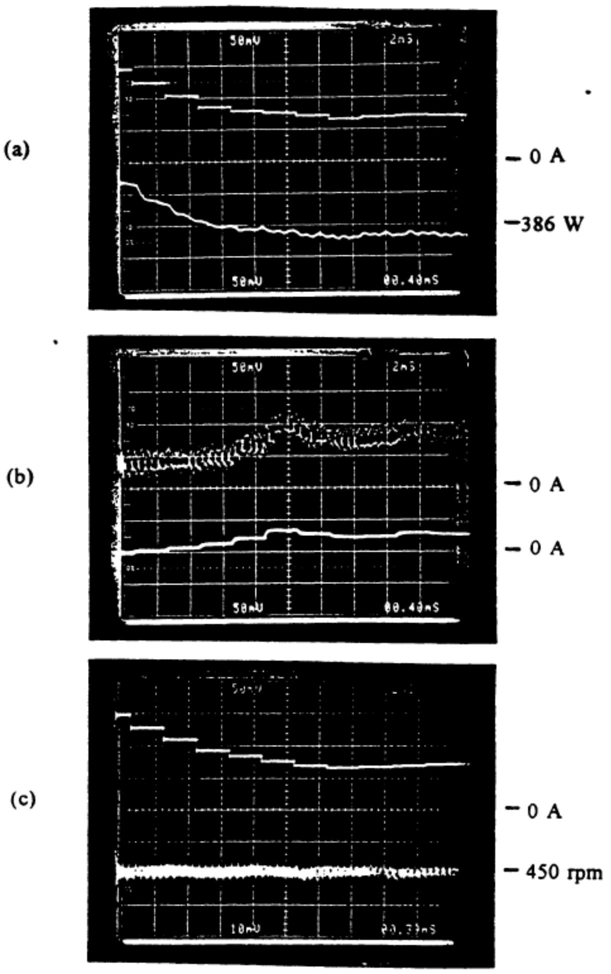

Fig. 8: Experimental search curves at ω_r=0.25 pu and T_L =0.10 pu.
a) Top: i_{ds}^* (3.33 A/div.); Bottom: P_d (58.5 W/div.);
b) Top: i_{qs}^* (1.67 A/div.); Bottom: $\Sigma\Delta i_{qs}^*$ (58.5 W/div.);
c) Top: i_{ds}^* (3.33 A/div.); Bottom: ω_r (61 rpm/div.);
Time scale: 2 sec./div.

The optimum operating point was achieved in only six steps. After that, the search algorithm remained active, with small Δi_{ds}^* steps, to ensure proper tracking in case the optimum point shifts due to parameter variation or slow changes in load torque. Good agreement with the simulation results was observed.

V. CONCLUSION

For large, complex power electronics and drive systems, available FLC development tools are not adequate, since they lack the capabilities of properly representing the plant and other non-fuzzy controllers. It has been demonstrated how SIMNON, a system simulation language, can be adapted to model a fuzzy controller. The methodology has the advantages that it is reasonably simple, does not require any expensive development tool, permits easy iteration in a simulation environment (without any fear of plant damage) and that can be directly translated into optimized control code for real time implementation. Real time implementation of fuzzy Logic controllers in C or Assembly languages might look time consuming at a first glance, but the use of a single, familiar development environment, combined with the availability of previously constructed basic building blocks can dramatically facilitate the design process and reduce design costs. Therefore, this alternative should be seriously considered when designing fuzzy logic controllers.

REFERENCES

[1]B. K. Bose, “Expert system, fuzzy logic and neural network applications in power electronics and motion control ”, Proc. of the IEEE, pp. 1303-1323, August 1994.

[2] G. C. D. Sousa and B. K. Bose, " A fuzzy set theory based control of a phase-controlled converter dc machine drive", IEEE/IAS Annual Meeting Conf. Rec., pp. 854-861, 1991.

[3]Y. F. Li and C. C. Lau, "Development of fuzzy algorithms for servo systems", IEEE Control System Magazine, Vol. 9, no. 3, pp. 65-72, 1989.

[4]C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller, Part I", IEEE Trans. on Systems Man and Cybernetics, Vol. 20, pp. 404-418, Mar./April 1990.

[5]C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller, Part II,", IEEE Trans. Systems Man and Cybernetics, Vol. 20, pp. 419-435 Mar./April 1990.

[6]Y. Jani, G. C. D. Sousa, W. Turner, R. J. Spiegel and P. J. Chappell, "Fuzzy efficiency optimization of ac induction motors", Proc. of the 3rd Fuzzy Logic Control Symposium, Houston, June 1-3, 1992.

[7]G. C. D. Sousa, B. K. Bose, J. G. Cleland, "Fuzzy logic based on-line efficiency optimization control of an indirect vector controlled induction motor drive.", IECON' 93 Conf. Rec., pp. 1168-1174, 1993.

[8]G. C. D. Sousa, B. K. Bose and Kyung S. Kim, "Fuzzy logic based on-line MRAC tuning of slip gain for an indirect vector-controlled induction motor drive", IECON' 93 Conf. Rec., pp.1003-1008, 1993.

[9]Jih-Sheng Lai, "SIMNON simulation for induction motor drives", Proc. of the 22 Southeastern Symposium on System Theory, pp. 488-493, 1990.

[10] H. Elmqvist, K. J. Astrom, and T. Schontal, SIMNON User's Guide for MS-DOS Computers, Version 1.0, 1986.

[11] G. Hill, E. Horsthotte, and J. Teichrow, TILShell User's Manual, Version 1.2, 1990.

[12] R. Nolan, P. Pillay, and T. Haque, "Application of genetic algorithms to motor parameter determination", IEEE/IAS Annual Meeting Conf. Rec., pp. 47-54, 1994.

[13] S. Thaler, "Fuzzy rule generation based on a neural network approach", Electronic Engineering, July 1993, pp. 43-50.

BIOGRAPHIES

Gilberto Costa Drumond Sousa was born in Faria Lemos, Minas Gerais, Brazil, on October 5, 1958. He received the B.E. degree in electrical engineering from the Federal University of Espirito Santo, Brazil, in 1981, and the M.S. degree from the Federal University of Santa Catarina, Brazil, in 1986. In 1989 he was awarded a scholarship by the Brazilian Agency for Research and Development (CNPq) to pursue doctoral studies at the University of Tennessee, Knoxville, USA, where he got his Ph.D. in 1993.

He has been a member of the faculty at the Electrical Engineering Department of the Federal University of Espirito Santo, since 1982, where he has taught several courses in power systems, electrical machines and more recently, power electronics and drives at graduate level. He has recently started to act as a research supervisor at UFES' Master Program in Electrical Engineering. His research interests comprehend power electronics and drives, machine modeling and power systems planing. He has published several papers at both national and international conferences, as well as some papers in IEEE transactions.

Dr. Sousa is a member of IEEE Industry Application Society, The Brazilian Power Electronics Society (SOBRAEP), and The Brazilian Automatic Control Society (SBA).

Bimal Kumar Bose received the B. E. degree from Calcutta University, Calcutta, India, in 1956, the M.S. degree from the University of Wisconsin, Madison, in 1960, and the Ph.D. degree from Calcutta University in 1966.

He was a member of the Faculty of Calcutta University, India, where he was awarded the Premchand Roychand Scholarship and the Mout gold medal for outstanding contributions. In 1971 he joined Rensselaer Polytechnic Institute, Troy, NY, as Associate Professor of Electrical Engineering where he was responsible for the power electronics program for five years. From 1976 to 1987, he was an Engineer at the General Electric Research and Development Center, Schenectady, NY, where he did extensive research in power converters, ac drives, microcomputer control, and the like. In 1987 he joined the University of Tennessee, Knoxville, as Professor of Electrical Engineering (Condra Chair of Excellence in Power Electronics). He is also the Distinguished

Scientist of the Power Electronics Applications Center, Knoxville. He has published more than 120 papers and holds 18 US patents. He is the author of Power Electronics and AC Drives (Prentice-Hall, 1986) and editor of several other IEEE press books.

Dr. Bose is Chairman of IEEE Industrial Electronics (IE) Society Power Electronics Council and has served as Associate Editor of the IE Transactions, as well as to other IEEE societies publications. He is also the recipient of several awards for his outstanding contributions to the Power Electronics and Drives technology.

Marcelo Godoy Simões was born in São Paulo, Brazil, on February 03, 1963. He received his BE degree in Electrical Engineering from "Escola Politécnica da Universidade de São Paulo" on December 1985. Upon graduation he worked for FDTE, a research institution that belongs to The University of São Paulo, for three years. He has been a professor at the University of São Paulo (Brazil) since 1989, and obtained his MS degree in July, 1990, at the University of São Paulo. He was awarded a Brazilian scholarship to pursue his doctoral degree at The University of Tennessee, Knoxville. During his doctoral studies at The University of Tennessee, he published several papers in IEEE conferences, some of them were accepted for publication in IEEE Transactions. He has also two patents pending for the new strategies presented in his Ph.D. dissertation for control of drives and performance enhancement of wind generation systems. He received the Doctor of Philosophy degree in December, 1995, and now he is working as a professor at The University of São Paulo (Brazil), doing research and development of systems on the application of artificial intelligence in power electronics, drives, and machine control.