

Diciembre 2006  
Volumen 2  
Número 1

# RIEE & C

---

---

---

**Revista de Ingeniería Eléctrica, Electrónica y Computación**

Revista publicada por el Departamento  
de Ingeniería Eléctrica y Electrónica del  
Instituto Tecnológico de Sonora



# RIEE&C

**Revista de Ingeniería Eléctrica, Electrónica y Computación**

**Editor en jefe:**

Javier Pérez Ramírez

**Grupo editor ejecutivo:**

José Antonio Beristáin Jiménez

Juan Carlos Murrieta Lee

Manuel Domitsu Kono

**Editores asociados:**

Adolfo Espinoza Ruiz

Adolfo Soto Cota

Andrés Othón Pizarro Lerma

Armando Ambrosio López

Armando García Berumen

Eduardo Romero Aguirre

Enrique Aragón Millanes

Erica Ruiz Ibarra

Gabriel Nuñez Román

Héctor Hernández López

Ismael Murillo Verduzco

Joaquín Cortez González

José Manuel Campoy Salguero

Juan José Padilla Ybarra

Moisés Rodríguez Echevarría

Rafael León Velázquez

**Equipo editorial:**

Jessica Beltrán Márquez

Vanessa Miranda

Alfredo Quintero Núñez

# Single Phase DC/AC Bi-Directional Converter with High-Frequency Isolation

Beristáin J. José A., Bordonau F. Josep, Busquets M. Sergi, Rocabert S. Joan and Murillo V. Ismael

**Abstract**— This paper introduces a comprehensive study of a dc/ac inverter with high-frequency isolation and bidirectional power flow capability. The topology is based on a high-frequency dc/ac converter, a high-frequency transformer, an ac/ac converter and a low-pass filter. In this scheme no additional dc-link stages are required. A LQR+I control approach and a modulation strategy for the commutation of the ac/ac converter switches is presented. Simulation and experimental verifications are provided.

**Key words**— High Frequency Isolation, Bi-directional power flow, soft commutation and renewable energy.

## I. INTRODUCTION

Dc/ac inverter systems present a number of applications: induction motor drives, uninterruptible power supplies, photovoltaic energy systems, fuel cell systems, small wind power systems, etc. In many cases, isolation between the input and the output of the inverter is required, because of operating specifications or for security reasons. Significant contributions to conversion systems with HF (High Frequency) isolation can be found in [1]-[5]. The advantage of the idea studied in this paper is to have compact high-frequency isolation without any additional stage (usually a dc stage and a filter is required) and a simple control. The topology is divided in three parts: a high frequency dc/ac converter, a high-frequency transformer and an ac/ac converter.

A comprehensive study of the converter with galvanic isolation is presented in this work. The topology is used in [1],[4],[5]. A LQR+I control method that allows regulating the voltage and the output frequency is presented. The advantages of this structure are that it operates with output power up to 1 kW

Manuscript received July 8, 2006. This work was supported by the Ministerio de Ciencia y Tecnología, Madrid, Spain, under Grant TIC-2002-03036, DPI 2000-0006-P4-02 and by ITSON-PROMEP, México..

Beristáin J. José A. is with the Instituto Tecnológico de Sonora, Electric and Electrónica Engineering Department; Av. Antonio Caso S/N Col. Villa ITSON; Ciudad Obregón, Sonora, México; C.P. 85138; Tel: (644) 4109000, ext. 1200; Fax: (644) 4109001.(e-mail bantonio@itson.mx).

Bordonau F. Josep is with the Technical University of Catalonia, Electronic Engineering Department; Av. Diagonal 647, Despatx 9-24, Campus Sud - Edif. H, CP 08028 Barcelona, Sapain. (email josep.bordonau@upc.edu).

Busquets M. Sergi is with the Technical University of Catalonia, Electronic Engineering Department; DESPATX 212, Campus Terrasa- Edif. TR2C. Colom, CP 108222, Terrasa, Spain (email Sergio.busquets@upc.edu).

Rocabert S. Joan is with the Technical University of Catalonia, Electronic Engineering Department; Av. Diagonal 647, Despatx 9-24, Campus Sud - Edif. H, CP 08028 Barcelona, Sapain. (email rocabert@upc.edu).

Murillo V. Ismael is with the Instituto Tecnológico de Sonora, Electric and Electrónica Engineering Department; Av. Antonio Caso S/N Col. Villa ITSON; Ciudad Obregón, Sonora, México; C.P. 85138; Tel: (644) 4109000, ext. 1200; Fax: (644) 4109001.(e-mail imurillo@itson.mx).

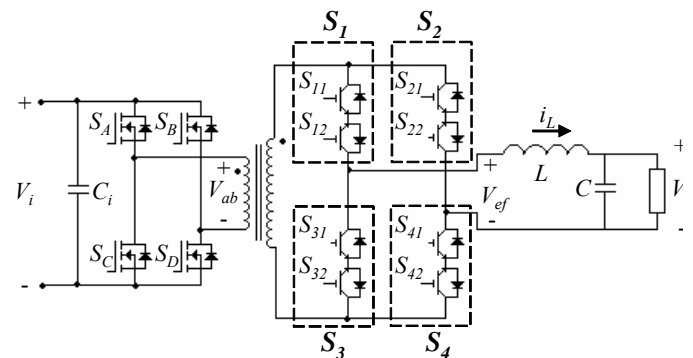


Fig. 1 Dc/ac converter with high-frequency isolation.

and has lower complexity (only two conversion stages are used, and without the need of dc-link filter). The LQR+I control provides an effective way to deal with multivariable controls typical in hybrid energy systems.

## II. DESCRIPTION AND CIRCUIT CONFIGURATION

The objective of the single-phase HF-Link inverter is to generate a sinusoidal voltage with galvanic isolation and bidirectional power flow capability. The topology used in this work is presented in Fig.1.

### A. Inverter description

The single-phase inverter generates a high-frequency ac voltage at the transformer primary. It generates a high-frequency PWM voltage by means of closed-loop bipolar control. Bipolar voltage is needed because of the use of a HF transformer.

Fig. 1 shows the basic configuration of the high-frequency link inverter. The devices  $S_A$ - $S_D$  form an inverter circuit that feeds the transformer with a high-frequency ac voltage providing isolation between the dc source and the load.

The devices  $S_{11}$ - $S_{12}$ ,  $S_{31}$ - $S_{32}$ ,  $S_{21}$ - $S_{22}$  and  $S_{41}$ - $S_{42}$  are bidirectional switches and form an ac/ac converter. When the energy flows from the dc source, the ac/ac converter operates as a positive or negative rectifier. The output filter eliminates the high frequency harmonics and delivers a sinusoidal voltage to the load. The load has been modeled as a resistor, but the results can be easily extended to others applications.

Table I outlines the 4 different dc/ac converter switching states and their associated output voltage presented at primary of the transformer,  $V_{ab}$ .

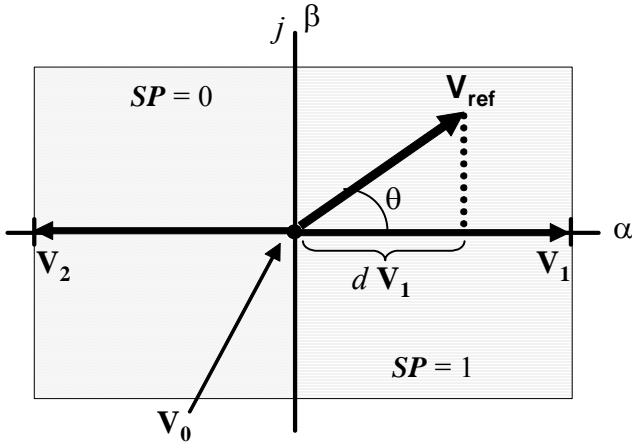
Table II shows the 4 different ac/ac converter switching states and their associated output voltage presented at input of low-pass filter,  $V_{ef}$ .

TABLE I.- DC/AC SWITCHING CONVERTER STATES

$S_A$	$S_B$	$S_C$	$S_D$	$V_{ab}$
ON	OFF	OFF	ON	$V_i$
ON	ON	OFF	OFF	0
OFF	OFF	ON	ON	0
OF	ON	ON	OFF	$-V_i$

TABLE II.- AC/AC SWITCHING CONVERTER STATES

$S_1$	$S_2$	$S_3$	$S_4$	$V_{ef}$
ON	OFF	OFF	ON	$V_{ab}$
ON	ON	OFF	OFF	0
OFF	OFF	ON	ON	0
OF	ON	ON	OFF	$-V_{ab}$

Fig. 2 State vector diagram of the dc/ac converter:  $V_{ref}$ , output reference voltage;  $d$ , duty ratio;  $sp$ , output voltage sign.

### B. Modulation Strategy

Different modulations strategies can be derived. In order to show the converter operation, a state vector modulation is chosen. State vector modulation is easily implemented in DSP devices.

Fig. 2 describes how the state vectors are defined in this system. The relative position of the reference vector in the right or left half-plane define if the  $V_1$ ,  $V_0$  or the  $V_2$  state vectors are used, see how there are three different state vector for the converter.

The state vectors are defined as:

$$\begin{aligned} V_1 &= tr V_i + j0 \\ V_2 &= -tr V_i + j0 \\ V_0 &= 0 + j0 \end{aligned} \quad (1)$$

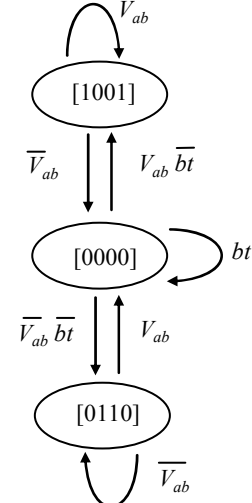
where

$tr$  is the turn ratio of the HF transformer.

The projection of the output reference vector defines the duty-ratio  $d$  of the state vector. The zero vector is applied the rest of the period.

The amplitude ( $m$ ) of the output voltage is

$$m = \sqrt{2} V_{o,rms} < tr V_i \text{ and with } f_o \text{ frequency}$$

Fig. 3 Commutation strategy for the dc-ac converter if square output voltage is needed. Where  $bt$  is the blanking time,  $[S_x]=[S_A S_B S_C S_D]$  and  $V_{ab}$  is the primary transformer voltage.

The reference vector is defined by:

$$\begin{aligned} \mathbf{V}_{ref} &= m e^{j\theta} \begin{cases} \theta = \omega t \\ \omega = 2\pi f_o \end{cases} \quad (2) \\ |\mathbf{V}_{ref}| &= \sqrt{2} V_{o,rms} = m \end{aligned}$$

Different state vector modulation strategies can be derived. The nearest two vectors are chosen to show the inverter operation. The duty ratio is defined as follows:

$$d = d_1 + d_2$$

If  $Sp=1$  the nearest two vectors are  $V_1$  and  $V_0$  and

$$d_1 = \frac{m \cos \theta}{tr V_i}, \quad d_0 = 1 - d_1 \text{ and } d_2 = 0 \quad (3)$$

If  $Sp=0$  the nearest two vectors are  $V_2$  and  $V_0$  and

$$d_2 = -\frac{m \cos \theta}{tr V_i}, \quad d_0 = 1 - d_2 \text{ and } d_1 = 0 \quad (4)$$

In (3)-(4), the duty-ratios  $d_1$  and  $d_2$ , define the projection of the reference vector (Fig. 2) for  $SP=1$  and  $SP=0$  respectively. The duty-ratio  $d_0$  correspond to the zero vector. The HF inverter on the DC side is commanded by the duty-ratio  $d$ .

### B. Commutation strategy

The commutation strategy must satisfy some rules:

- A short circuit of the dc source is not permitted.
- The current through the inductors cannot be interrupted.
- A commutation strategy verifying these rules and taking advantage of the possibilities for soft switching is proposed.

The commutation strategy for the dc/ac converter is defined by the modulation strategy. If the modulation is applied to the ac-ac converter, the commutation strategy for the dc-ac converter is

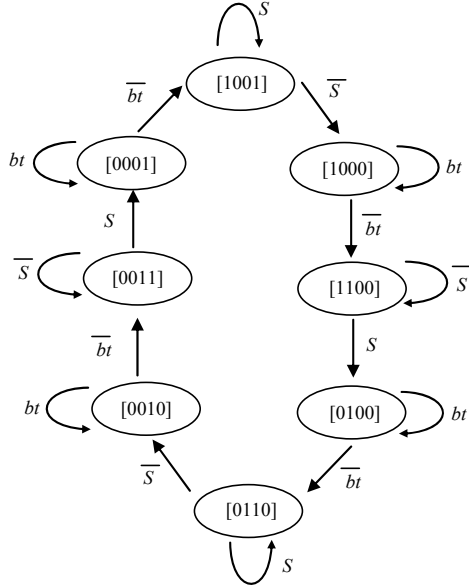


Fig. 4 Commutation strategy if PWM is applied to dc-ac converter. Where  $S$  is a switching function associated to the duty ratio,  $d$ .

defined as in Fig. 3. If the modulation is applied to the dc-ac converter, the commutation strategy is defined as in Fig. 4.

The commutation strategy for the ac/ac converter is based on a four step technique [8]. The decision about how to commute the ac/ac converter is based on the modulation strategy.

In order to generate the control signals of the ac/ac converter four input signals are required: the inductor current sign ( $i_L$ ), the output voltage sign ( $sp$ ), the transformer voltage ( $V_{ab}$ ) and the pulse-width modulation signal associated to the duty-ratio  $d$  indicating the instantaneous value of the output voltage ( $S$ ). The variable  $S$  is the switching function.

Fig. 5 shows one example of transition between  $S_1$  ON to  $S_1$  OFF.

A PLD implements the commutation sequence of the inverter and the ac/ac converter modulation and control scheme is implemented in a DSP.

Voltage spikes caused by the transformer secondary leakage inductance are clamped and recovered to the dc source by an energy feedback circuit presented in [4].

### III. LARGE-SIGNAL AND SMALL-SIGNAL MODELS

Fig. 6 shows the switching model of the converter. The assumptions for the analysis are:

- Ideal semiconductors.
- Ideal HF transformer.
- Ideal dc input source.
- The losses in the system are neglected.
- A resistive load is considered.
- The switching frequency is significantly greater than the output fundamental frequency and the system cut-off frequencies.

The commutation function  $S_x(t)$  is used to define the switching state of each converter switch:

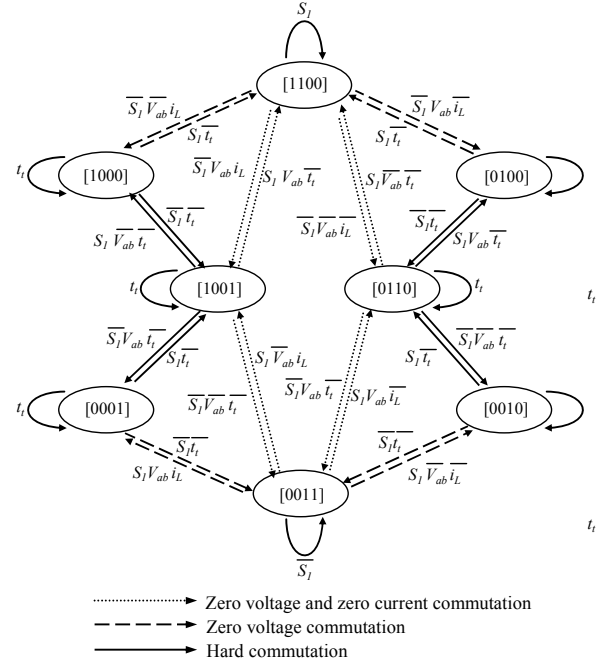


Fig. 5 Commutation between  $S_1$  ON,  $S_3$  OFF and  $S_1$  OFF and  $S_3$  ON.

$$S_x(t) = \begin{cases} 0, & \text{switch open} \\ 1, & \text{switch closed} \end{cases} \quad (5)$$

In order to avoid the short-circuit of the input capacitive filter and the open circuit of the primary (in Fig. 6), it is necessary to verify that:

$$S_A + S_C = 1, \quad S_B + S_D = 1 \quad (6)$$

Additionally to avoid a short circuit in the secondary of the transformer and to avoid opening the output filter inductor, the following restrictions must be fulfilled:

$$S_{11} + S_{31} = 1, \quad S_{21} + S_{41} = 1, \quad S_{12} + S_{32} = 1, \quad S_{22} + S_{42} = 1, \quad (7)$$

$$(S_{11} \cdot S_{12}) + (S_{31} \cdot S_{32}) = 1 \quad \text{and} \quad (S_{21} \cdot S_{22}) + (S_{41} \cdot S_{42}) = 1$$

If the particular case where  $V_{ab}$  is a pure square signal, the expression of the input voltage of the LC filter using switching functions (see Fig. 3) is:

$$V_{ef} = (S_{ap} - S_{bp})(S_{ec} - S_{fc}) tr V_i \quad (8)$$

where  $tr$  is already defined as the turns ratio of the high-frequency isolation transformer.

Analyzing the circuit, the state equations are obtained.

$$L \frac{d}{dt} i_L = -V_o + V_{ef} \quad (9)$$

$$C \frac{d}{dt} V_o = i_L - \frac{V_o}{R} \quad (10)$$

The state-space model of the system is expressed in (11)-(15). It represents the discrete time-variant large-signal model of the dc/ac converter.

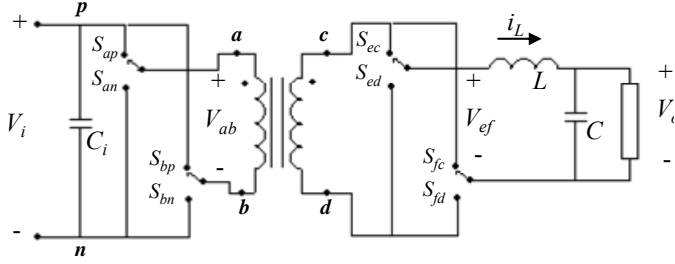


Fig. 6 Equivalent switching model of the converter.

$$\frac{d}{dt} [\mathbf{x}] = \mathbf{A} [\mathbf{x}] + \mathbf{B} [\mathbf{u}] \quad (11)$$

$$\text{where: } [\mathbf{x}] = [i_L \ V_o]^T \quad (12)$$

$$\mathbf{A} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \quad (13)$$

$$\mathbf{B} = \begin{bmatrix} \frac{trV_i}{L} \\ 0 \end{bmatrix} \quad (14)$$

$$[\mathbf{u}] = [(S_{ap} - S_{bp})(S_{ec} - S_{fc})] \quad (15)$$

The use of the moving average technique avoids the high-frequency analysis and focuses in the low frequency components of (11). The filtering effect of inductor  $L$  and capacitor  $C$  justify the approach.

If the switching period is  $T_s$ , then the moving average of any variable  $x$ , called  $\hat{x}$  is defined as:

$$\hat{x}(t) = \frac{1}{T_s} \int_{t-T_s}^t x(\tau) d\tau \quad (16)$$

The duty cycle is defined as in (17):

$$d = \hat{S} = \frac{1}{T_s} \int_{t-T_s}^t (S_{ap} - S_{bp})(S_{ec} - S_{fc}) d\tau \quad (17)$$

The averaged state-space equation in Direct-Quadrature (DQ) coordinates is obtained using the average method and applying the DQ transformation [7], resulting in (18), where  $\omega = 2\pi f$  and  $f$  is the output voltage frequency. The averaged time-invariant large-signal model is in (18). For notation simplicity, from now on, all the time-domain variables are assumed to be averaged.

$$\frac{d}{dt} \begin{bmatrix} \hat{i}_{Ld} \\ \hat{i}_{Lq} \\ \hat{v}_{Od} \\ \hat{v}_{Oq} \end{bmatrix} = \begin{bmatrix} 0 & -\omega & -\frac{1}{L} & 0 \\ \omega & 0 & 0 & -\frac{1}{L} \\ \frac{1}{C} & 0 & -\frac{1}{RC} & -\omega \\ 0 & \frac{1}{C} & \omega & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} \hat{i}_{Ld} \\ \hat{i}_{Lq} \\ \hat{v}_{Od} \\ \hat{v}_{Oq} \end{bmatrix} + \begin{bmatrix} \frac{tr\hat{V}_i}{L} & 0 \\ 0 & \frac{tr\hat{V}_i}{L} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{d}_d \\ \hat{d}_q \end{bmatrix} \quad (18)$$

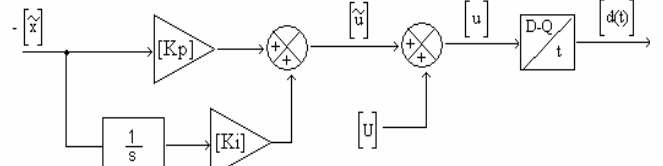


Fig. 7 Control scheme for the converter.

From (18), the steady-state values can be obtained forcing all time-derivatives to zero.

$$D_d = \frac{tr}{V_i} \left[ \frac{\omega L}{R} V_{oq} + V_{od} (1 - \omega^2 LC) \right] \quad (19)$$

$$D_q = \frac{tr}{V_i} \left[ -\frac{\omega L}{R} V_{od} + V_{oq} (1 - \omega^2 LC) \right] \quad (20)$$

$$I_{Ld} = \omega C V_{oq} + \frac{V_{od}}{R} \quad (21)$$

$$I_{Lq} = -\omega C V_{od} + \frac{V_{oq}}{R} \quad (22)$$

Applying the small-signal analysis to (18), the averaged time-invariant small-signal model is shown in (23).

$$\frac{d}{dt} \begin{bmatrix} \tilde{i}_{Ld} \\ \tilde{i}_{Lq} \\ \tilde{v}_{Od} \\ \tilde{v}_{Oq} \end{bmatrix} = \begin{bmatrix} 0 & -\omega & -\frac{1}{L} & 0 \\ \omega & 0 & 0 & -\frac{1}{L} \\ \frac{1}{C} & 0 & -\frac{1}{RC} & -\omega \\ 0 & \frac{1}{C} & \omega & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} \tilde{i}_{Ld} \\ \tilde{i}_{Lq} \\ \tilde{v}_{Od} \\ \tilde{v}_{Oq} \end{bmatrix} + \begin{bmatrix} \frac{trV_i}{L} & 0 \\ 0 & \frac{trV_i}{L} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{d}_d \\ \tilde{d}_q \end{bmatrix} \quad (23)$$

## V. CONTROL SCHEME

For closed-loop control, a linear quadratic regulator [LQR+I] in the DQ domain is proposed [6],[7].

LQR+I is used for closed-loop control since it allows obtaining an optimum multivariable controller and since all the state variables that are used for feedback can be easily sensed. The scheme is shown in Fig. 6.

The cost function for the design of the LQR regulator defines a trade-off between the output voltage error, the integral of the output voltage error and the control variables effort in order to avoid saturation. Indeed, three variables have been taken into account in the multivariable cost function.

## VI. SIMULATION AND EXPERIMENTAL RESULTS

A Simulation was done with the following specifications: output voltage  $V_o = 110$  Vrms, input voltage  $V_i = 24$  Vdc, output power  $P_{omax} = 160$  W, output voltage frequency  $f = 50$  Hz,  $f_s = 20$  kHz,  $L = 2$  mH and  $C = 10$  uF.

The large-signal model in (14) allows simulating the converter. The operation output voltage and inductor current in closed-loop are shown in Figs. 8, 9.

An experimental prototype has been implemented using functional modules. It is shown in Fig. 10. Experiments have been performed in order to verify the characteristics of the proposed circuit.

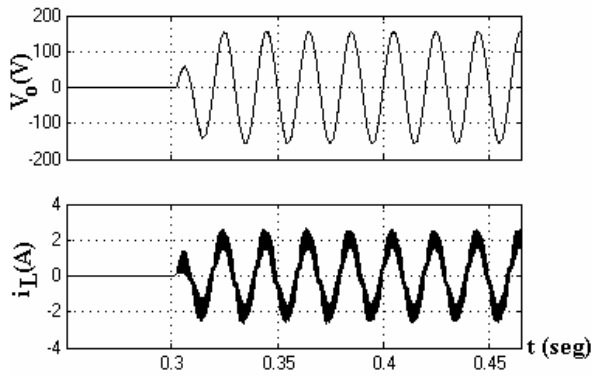


Fig. 8 Closed-loop simulation waveform:  $V_o$  and  $i_L$ . Start-up of the converter.

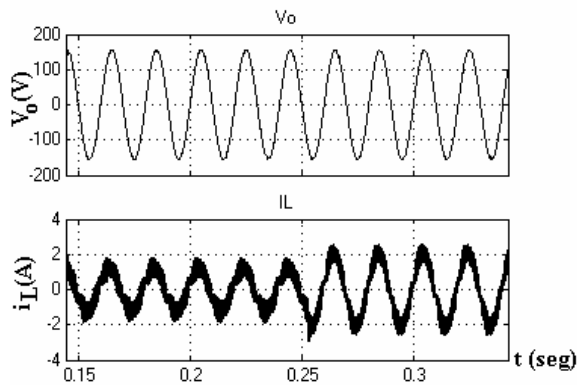


Fig. 9 Closed-loop simulation waveforms:  $V_o$  and  $i_L$ . Power step from 50% to 100% of rated power.

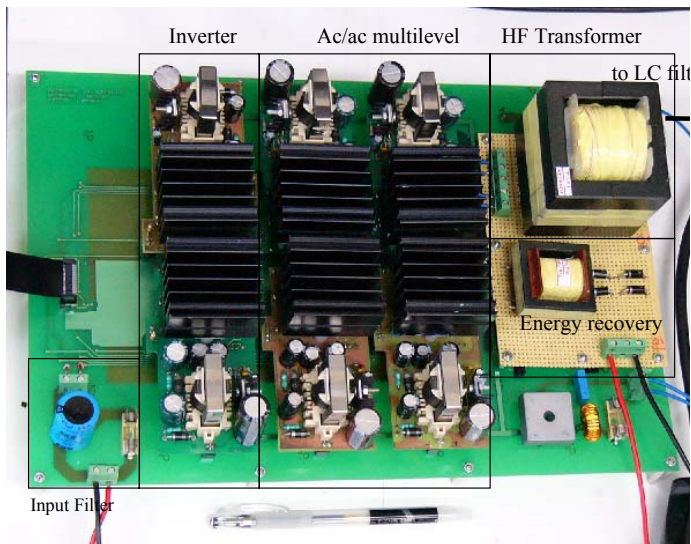


Fig. 10 Dc/ac Converter prototype.

The experimental conditions are:  $V_i=24\text{Vdc}$ ,  $V_o=110\text{Vrms}$ ,  $P_{omax}=160\text{ W}$ ,  $f=50\text{Hz}$ ,  $f_s=20\text{kHz}$ ,  $tr= 8.33$ ,  $L=2\text{mH}$  and  $C=10\mu\text{F}$ .

Fig. 11 includes the output voltage and inductor current during the start-up of the converter. The results show that the response is very fast.

Figs. 12 and 13 show the evolution of the output voltage and the inductor current during a load transient. The response is very fast and there is no overshoot.

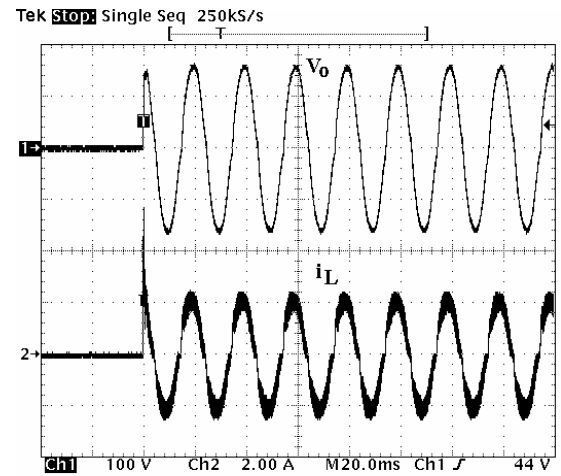


Fig. 11 Experimental results:  $V_o$  and  $i_L$  in closed-loop. Start-up of the converter.

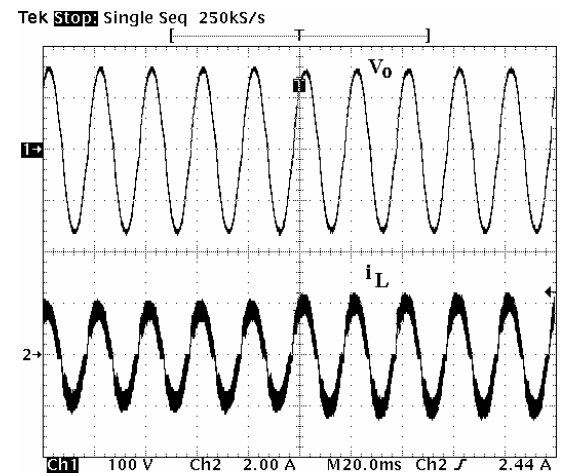


Fig. 12 Experimental results:  $V_o$  and  $i_L$  in closed-loop. Power step from 85% to 100% of rated power.

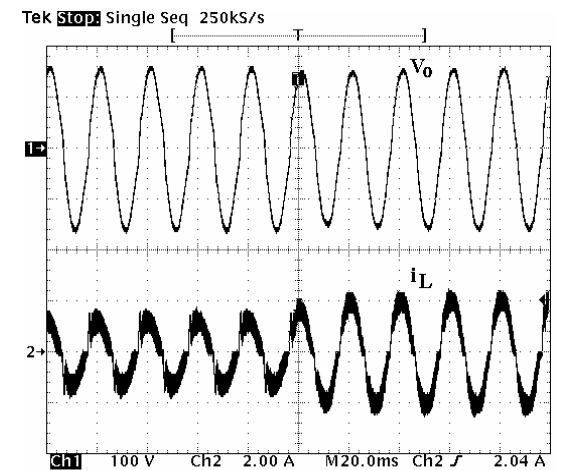


Fig. 13 Experimental results:  $V_o$  and  $i_L$  in closed-loop. Power step from 50% to 100% of rated power.

## VII. CONCLUSIONS

This paper presents a comprehensive study of a dc/ac inverter with high-frequency isolation and bidirectional power flow capability. The benefits of this topology are: reduced number of conversion stages, no dc-link required, and bidirectional operation capability. The LQR+I control approach for a dc/ac topology with high-frequency isolation is presented. The LQR+I is an effective way to implement a multivariable control. Simulations of the proposed control method have proven its good performance in terms of response time and overshoot. The experimental results correlate perfectly with the model and its performance with small overshoot and very low steady-state error. These results show the benefits of the LQR+I control technique.

## REFERENCES

- [1]. Harada, K.; Sakamoto, H.; Shoyama, M., "Phase-controlled DC-AC converter with high-frequency switching", IEEE Trans. Power Electronics, Vol.: 3, Oct. 1988, pp. 406–411.
- [2]. Aoki, T.; Yotsumoto, K.; Muroyama, S.; Kenmochi, Y., "A new uninterruptible power supply with a bidirectional cycloconverter", Intl. Telecom. Energy Conf., 1990. INTELEC '90, pp. 424–429.
- [3]. Fujimoto, H.; Kuroki, K.; Kagotani, T.; Kidoguchi, H., "Photovoltaic inverter with a novel cycloconverter for interconnection to a utility line", IEEE IAS Annual Meeting, IAS '95., pp. 2461–2467.
- [4]. Sree, H.; Mohan, N., "Voltage sag mitigation using a high-frequency-link cycloconverter-based DVR," IEEE Industrial Electronics Soc. Conf., 2000. IECON'00, pp. 344–349.
- [5]. Lima, F.; Anunciada, V. "Novel active uninterruptible power supply using high frequency ZVS cycloconverters," IEEE Intl. Symp. Industrial Electronics, 2000. ISIE'00, pp. 112–117.
- [6]. Alepuz, S.; Salaet, J.; Gilabert, A. and Bordonau J. "Control of three-level VSIs with a LQR-based gain-scheduling technique applied to DC-link neutral voltage and power regulation", IECON2002, pp. 914–912.
- [7]. Salaet, J.; Alepuz, S.; Gilabert, A.; Bordonau, J.; Peracaula, J. "D-Q modeling and control of a single-phase three-level boost

rectifier with power factor correction and neutral-point voltage balancing", PESC2002, pp.514–519.

- [8]. Beristáin J. A. "Inversores bidireccionales con aislamiento en alta frecuencia para aplicaciones de energías renovables", Ph.D. Dissertation, Technical University of Catalonia, June 2005.

**Beristáin J. José A.** was born in Córdoba, Mexico on August 15, 1973. He received the Master's degree in Electronic Engineering from National Center for Technological Development and Research (cenidet) Cuernavaca México, in 1998 and Ph. D. degree in Electronic Engineering from Technical University of Catalonia (UPC), Barcelona Spain, in 2005.

He is currently a Professor of Electrical and Electronics Engineering at Technologic Institute of Sonora, Cd. Obregón, Sonora, México. His teaching and research interest are in application of power electronics in renewable energy systems.

**Bordonau F. Josep** was born in Sitges, Spain. He received the M.Sc. and Ph. D. degrees in electrical engineering (with honors) from Technical University of Catalonia, Barcelona, Spain, in 1984 and 1990, respectively.

He has been Lecture and Assistant Professor in the Electronics Engineering Department, Technical University of Catalonia, where he has been an Associate Professor since 1991. He has been active in research projects with international companies and institutions, and was principal researcher of more than 25 research projects. He has authored more than 50 journal and conference papers and is the holder of a European patent. His fields of interest are ac power converters applied to renewable energy systems and energy management systems.

**Busquets M. Sergio** was born in Barcelona, Spain, on February 14, 1974. He graduated in electrical engineering from the Technical University of Catalonia, Spain, in 1999, received the M.S. degree in electrical engineering from the Virginia Polytechnic Institute and State University, Blacksburg, VA, in 2001, and the Ph.D. degree in electrical engineering from the Technical University of Catalonia in 2006. From 2001 to 2002 he was with Crown International, Inc. He is currently an Assistant Professor in the Electronics Engineering Department of the Technical University of Catalonia. His research interests include multilevel conversion, power converter modeling, control and automated design, power factor correction, and EMI suppression techniques.

**Rocabert S. Joan** was born in Barcelona, Spain. He graduated in electrical engineering from the Technical University of Catalonia, Spain, in 2002. His research interests include multilevel conversion, power converter modeling, control and automated design, power factor correction, and EMI suppression techniques.

**Murillo V. Ismael** was born in Pueblo Yaqui, Mexico on June 16, 1961. He received the Master's degree in Mechanic Engineering from National Autonomous University of México (UNAM), in 2004.

He is currently a Professor of Electrical and Electronics Engineering at Technologic Institute of Sonora, Cd. Obregón, Sonora, México. His teaching and research interest are in application of Electric Machinery in Utilization of energy.



# Inversor trifásico SPWM para el control de velocidad de un motor de inducción implementado en el microcontrolador PIC18F2431

Aganza T. Alejandro, Pérez R. Javier y Beristain J. José Antonio

**Resumen**— En el presente artículo se presenta el desarrollo de un inversor trifásico empleado en el control de velocidad de un motor de inducción trifásico de jaula de ardilla de 1/3 HP, utilizando el método de control Volts - Hertz constante. La técnica de modulación empleada en la conmutación del inversor es la modulación por ancho de pulso senoidal (SPWM); dicha técnica fue implementada en el microcontrolador PIC18F2431. El énfasis de este artículo es la descripción y manejo de los recursos de control de motores y manejo de potencia del microcontrolador. Finalmente se muestran pruebas realizadas al sistema para observar su respuesta.

**Palabras clave**— Inversor, IGBT, microcontrolador PIC.

## I. INTRODUCCIÓN

Un inversor o convertidor DC/AC [1] como su nombre lo indica es un circuito donde se introduce un voltaje de corriente directa a la entrada y se obtiene un voltaje de corriente alterna de magnitud y frecuencia variables. El esquema básico del inversor alimentado por fuente de voltaje, el cual fue el utilizado en el desarrollo de este trabajo, se muestra en la figura 1.

El funcionamiento consiste en la rectificación de las señales de voltaje trifásicas para obtener un voltaje de corriente directa, posteriormente filtrado para disminuir el rizado; con dicho voltaje de CD se alimenta la etapa de potencia, la cual en este caso es un puente inversor empleando transistores IGBT's, los cuales son controlados mediante señales de disparo generadas por el microcontrolador PIC de la etapa digital con base a una referencia provista por el usuario a través de un potenciómetro conectado a la entrada del convertidor analógico a digital (ADC) del microcontrolador; las señales generadas por el PIC se adecuan para introducirlas al puente inversor cuya salida se aplica a la carga.

## II. IMPLEMENTACIÓN DEL HARDWARE DEL INVERSOR

En la figura 1 se pueden observar las etapas que conforman el sistema del inversor, las cuales se describen a continuación:

Manuscrito recibido el 4 de Julio de 2006. Este trabajo fue respaldado por el departamento de Ing. Eléctrica y Electrónica del Instituto Tecnológico de Sonora.

Aganza T. Alejandro hasta la fecha se ha de desempeñado como consultor y diseñador de sistemas electrónicos de manera independiente en Ciudad Obregón, Sonora, México. Tel Cel: (6449) 98-61-68; e-mail: [alejandro.aganza@gmail.com](mailto:alejandro.aganza@gmail.com).

Pérez R. Javier hasta la fecha se ha de desempeñado como Profesor de Tiempo Completo del Instituto Tecnológico de Sonora en el Departamento de Ingeniería Eléctrica y Electrónica Instituto Tecnológico de Sonora; Ave. Antonio Caso S/N Col. Villa ITSON; Ciudad Obregón, Sonora, México; C.P. 85138; C.P. 85130; Tel: (644) 4109000, ext. 101; Fax: (644) 4109001. (e-mail [jperez@itson.mx](mailto:jperez@itson.mx))

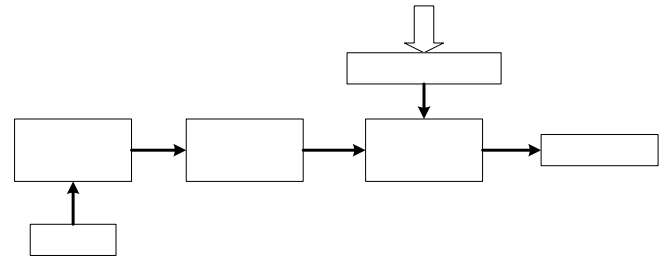


Fig.1 Diagrama a bloques del inversor alimentado por voltaje

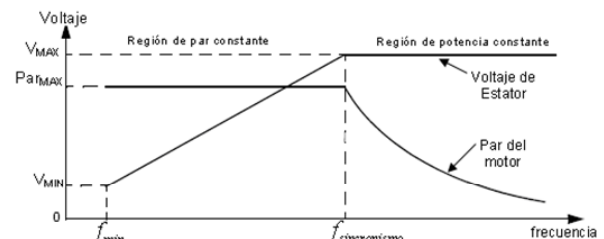


Fig. 2 Curva del par inducido en el motor de inducción trifásico para una relación  $V/f$  constante de entrada

### A. Carga

La carga utilizada fue un motor de inducción trifásico [2] de jaula de ardilla de 1/3 de HP con un voltaje nominal de trabajo de 220Vrms y una corriente nominal de 3.3 amperes, marca GENERAL ELECTRIC, con una velocidad máxima de giro de 1720 RPM a una frecuencia de voltaje de trabajo de 60 Hz.

Para llevar a cabo la variación de velocidad de este tipo de motor existen diferentes métodos descritos en [2], el método utilizado en el desarrollo de este trabajo es el de Voltaje-Frecuencia constante; el cual consiste en mantener una relación constante entre la magnitud del voltaje del estator del motor y la frecuencia eléctrica del mismo, con la finalidad de conservar el par constante a través del rango de operación del motor. La curva del par inducido en el motor para una relación Voltaje-frecuencia ( $V/f$ ) constante en un motor de inducción se muestra en la figura 2.

### B. Etapa de potencia

La etapa de potencia en el desarrollo del sistema se encuentra constituida por un puente inversor trifásico, topología de dos niveles [3], [4] y [5], implementado mediante transistores

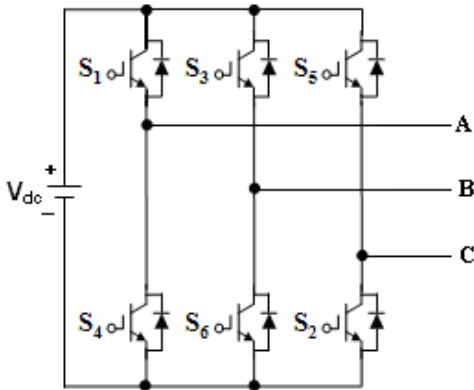


Fig. 3 Puente inversor trifásico de dos niveles.

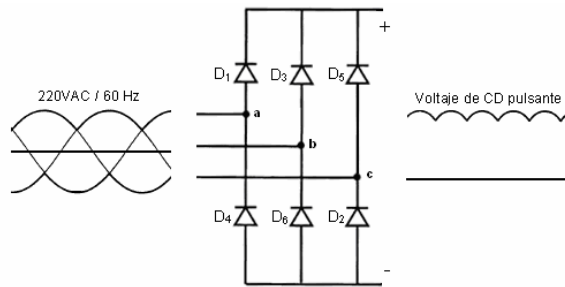


Fig. 4 Rectificador trifásico no controlado.

bipolares de compuerta aislada (IGBT's). La figura 3 muestra el diagrama esquemático del puente inversor utilizado.

Los transistores utilizados son el modelo FGH50N6S2D de Fairchild Semiconductor, los cuales cuentan internamente con diodos de marcha libre en antiparalelo, y tienen la capacidad de soportar un voltaje de trabajo de hasta 600 Volts y 60 Amperes @ 110 °C.

### C. Rectificador trifásico

La conversión de corriente alterna de la red de suministro eléctrico a un voltaje de corriente directa para la alimentación del puente inversor, se realizó mediante un rectificador trifásico no controlado de diodos, el cual se muestra en la figura 4.

La salida del rectificador proporciona un voltaje de corriente directa pulsante, por lo que se debe filtrar para reducir el rizado. Para filtrar la salida del rectificador se utilizó un filtro capacitivo. Sin embargo, debido a que la carga es inductiva, en este caso un motor de inducción, se maneja potencia reactiva en el sistema, por lo que el sistema del inversor debe ser capaz de manejar esta potencia. El elemento encargado de esta tarea es el filtro capacitivo, que también reduce el rizado del voltaje de CD; el método utilizado para el cálculo del filtro se describe en los 4 pasos siguientes:

1. Se realiza el análisis de potencia en la carga; para obtener el análisis de calidad de potencia se utilizó el medidor FLUKE modelo 43B [7]; de las mediciones realizadas se obtuvieron los siguientes datos de trabajo:

- Voltaje: 220 Vrms.
- Corriente: 2.5 Arms.
- Factor de potencia (FP): 0.71.

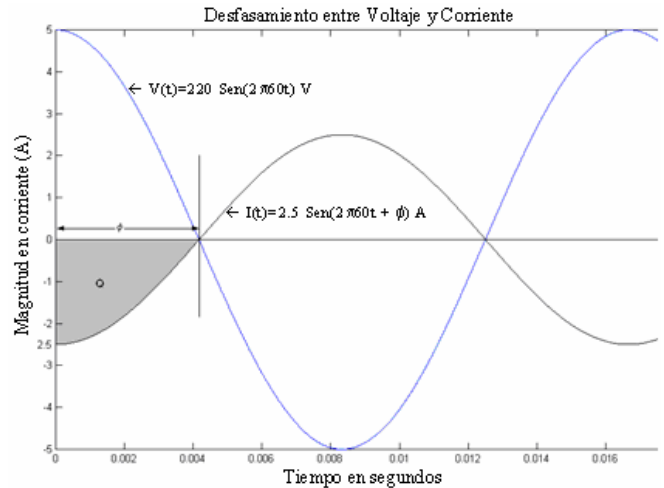


Fig. 5 Representación gráfica del voltaje y corriente en la carga.

- Potencia reactiva en la carga: 600 VAR.

2. Con los datos obtenidos se realiza una representación gráfica de las señales de voltaje y corriente para el máximo desfase entre ambos con el respectivo factor de potencia.

En la figura 5 se puede observar que el área sombreada es la carga debida a la potencia reactiva del sistema; por lo que esta carga es la máxima que debe ser capaz de manejar el filtro capacitivo.

3. Calcular la capacitancia del filtro, partiendo la fórmula 1:

$$C = \frac{Q}{V} \quad (1)$$

Para obtener la carga se realiza una aproximación del área bajo la curva de la corriente a un triángulo rectángulo, por lo que el área se obtiene de la fórmula:

$$A = \frac{bxh}{2} = \frac{(I_{trabajo})(tiempo)}{2} \quad (2)$$

La corriente se obtuvo del análisis realizado con el analizador de potencia, el desfase en tiempo se obtiene del factor de potencia en la carga, partiendo de:

$$FP = \cos\phi \quad (3)$$

Por lo que el desfase en tiempo, se obtiene a partir del desfase en grados, tomando en cuenta que un ciclo completo de 60 Hertz tiene un periodo de 16.667 ms.

$$\cos\phi = 0.71$$

$$\phi = \cos^{-1}(0.71) = 44.76^\circ$$

$$360^\circ \rightarrow 16.667mSeg$$

$$44.76^\circ \rightarrow x$$

$$x = 2.0725mSeg$$

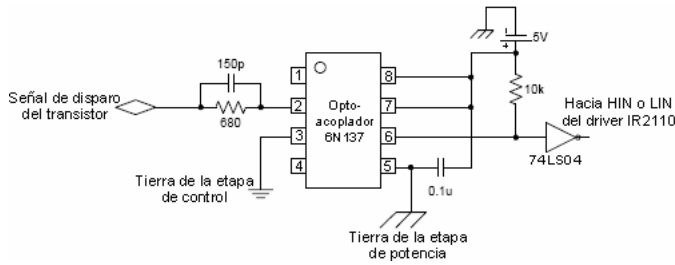


Fig. 6 Diagrama de conexión del optoacoplador 6N137.

Por lo que el área (carga) y las capacitancias obtenidas de (1) y (2) son:

$$A = \frac{bxh}{2} = \frac{(2.5 \text{ Amp})(2.0725 \text{ mSeg})}{2}$$

$$Q = 2.59062 \times 10^{-3} \text{ Coulombs}$$

$$C = \frac{Q}{V} = \frac{2.59062 \times 10^{-3} \text{ Coulombs}}{220 \text{ V}} = 11.7756 \mu\text{F}$$

4. Una vez obtenida la capacitancia se procede a obtener el voltaje de trabajo del capacitor, tomando en cuenta que la potencia reactiva está definida por:

$$\text{Potencia Reactiva} = V \cdot I \cdot \text{Sen}(\phi) \quad (4)$$

Tomando en cuenta que el valor de la potencia reactiva se obtuvo del análisis con el medidor FLUKE, se obtiene el voltaje de trabajo mínimo del capacitor, partiendo de (4).

$$600 \text{ VAR} = (V) \cdot (2.5) \cdot \text{Sen}(44.76^\circ)$$

$$V = \frac{600 \text{ VAR}}{(2.5) \cdot \text{Sen}(44.76^\circ)} = 340.842 \text{ Volts}$$

Cabe mencionar que los valores obtenidos son los valores mínimos; sin embargo debido a las aproximaciones realizadas se recomienda utilizar el valor comercial inmediato superior disponible. Para el desarrollo de este proyecto se proporcionó un margen de trabajo superior de aproximadamente 9 veces el valor mínimo de capacitancia, por lo que el valor del capacitor empleado fue de 99 μF a 440 Volts.

#### D. Aislamiento y adecuación

Una vez que se tiene la etapa de potencia y alimentación de la misma, es necesario aislar las señales de control provenientes de la etapa digital, para proporcionar una protección contra cortocircuito; así como una mayor inmunidad al ruido entre la etapa digital y la etapa de potencia. De la misma manera es necesario adecuar las señales de control para cumplir los requerimientos de funcionamiento de los transistores del puente inversor, garantizando así su correcto control de conducción.

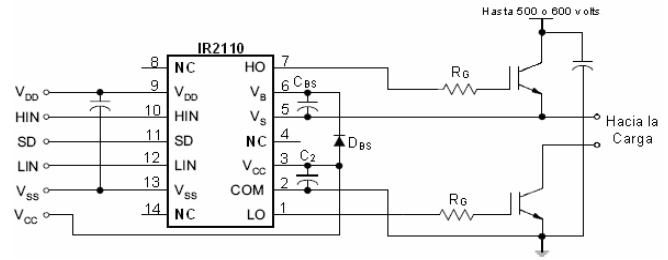


Fig. 7 Diagrama de conexión del driver IR2110.

Para obtener el aislamiento eléctrico entre la etapa digital y de potencia se utilizaron los optoacopladores modelo 6N137 de Fairchild Semiconductor, el diagrama esquemático para una señal de control se muestra en la figura 6.

Una vez aislados las señales de control se procede a adecuar las mismas, para esto se utilizaron los drivers para MOSFET e IGBT modelo IR2110 de International Rectifier. Dicho dispositivo es un driver doble, capaz de manejar dos transistores con una misma fuente de alimentación, para el caso de este trabajo se utilizó una fuente de 15 Volts, como se muestra en la figura 7. Esto mediante la utilización de un circuito de carga de capacitor o *bootstrap*, constituido por el diodo ( $D_{BS}$ ) y el capacitor ( $C_{BS}$ ) mostrados en la figura 7.

Dado que la función del driver es mantener las señales de control a un voltaje constante para el correcto funcionamiento y plena conducción de los transistores, el circuito de *bootstrap* debe ser calculado ya que depende de la frecuencia de trabajo y carga de la compuerta del IGBT.

El cálculo de capacitor de *bootstrap* del driver utilizado en el desarrollo de este proyecto, se basa en las especificaciones de diseño propuestas por el fabricante en [8], de donde se toma la siguiente ecuación:

$$C \geq \frac{2 \left[ 2Q_g + \frac{I_{qbs(max)}}{f} + Q_{ls} + \frac{I_{Cbs(leak)}}{f} \right]}{V_{CC} - V_f - V_{LS} - V_{min}} \quad (5)$$

Donde:

$Q_g$  = Carga de la compuerta del IGBT. [Coulombs]

$I_{qbs(max)}$  = Corriente de reposo en el lado superior del driver. [Amperes]

$Q_{ls}$  = Nivel de carga requerida por ciclo. [Coulombs]

$I_{Cbs(leak)}$  = Corriente de fuga del capacitor de *bootstrap*. [Amperes]

$f$  = Frecuencia de operación. [Hertz]

$V_{CC}$  = Voltaje de alimentación del driver. [Volts]

$V_f$  = Voltaje de caída a través del diodo. [Volts]

$V_{LS}$  = Voltaje de caída a través del transistor inferior. [Volts]

$V_{min}$  = Voltaje mínimo entre las terminales  $V_B$  y  $V_S$ .

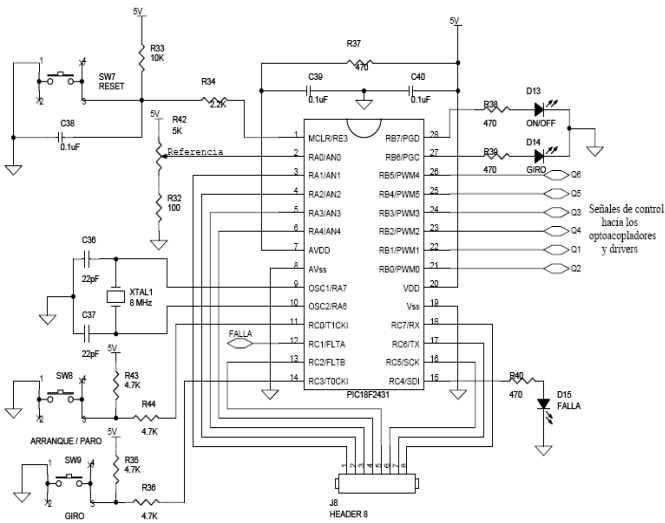


Fig. 8 Diagrama esquemático del sistema mínimo para el microcontrolador.

INICIA_ADC:		
CLRF	ADRESL	; LIMPIA LOS REGISTROS DE RESULTADO
CLRF	ADRESH	; DEL ADC
MOVLW	b'00010000'	; REFERENCIA DE VOLTAJE SELECCIONADA
MOVWF	ADCON1	; COMO LA ALIMENTACIÓN DEL MICROCONTROLADOR ; BUFFER FIFO HABILITADO
MOVLW	b'00110110'	; RESULTADO JUSTIFICADO A LA IZQUIERDA
MOVWF	ADCON2	; TIEMPO DE ADQUISICIÓN = 12 TAD ; BASE DE TIEMPO= Fosc/64
MOVLW	b'11000000'	; FUENTES DE DISPARO EXTERNAS
MOVWF	ADCON3	; DESHABILITADAS
MOVLW	b'00000000'	; SELECCIÓN GRUPO A
MOVWF	ADCHS	; CANAL AN0
MOVLW	b'11111111'	; PUERTO A CONFIGURADOS COMO ENTRADAS ANALÓGICAS
MOVWF	ANSEL0	
MOVLW	b'00111111'	; PUERTO A CONFIGURADO COMO ENTRADAS
MOVWF	TRISA	
MOVLW	b'00100011'	; CONFIGURACIÓN EN MODO DE CONVERSIÓN
MOVWF	ADCON0	; CONTINUA SOBRE EL CANAL A (AN0) ; SE ENCIENDE EL CONVERTIDOR

Fig. 10 Código de configuración del convertidor ADC.

```

include <p18f2431.inc> ; DEFINICIÓN DE CABECERA DEL µC

CONFIG OSC = HS ; OSCILADOR DE ALTA VELOCIDAD
CONFIG FCMEN = OFF ; MONITOR DE RELOJ HABILITADO
CONFIG IESO = OFF ; SWITCH INTERNO/EXTERNO
; DESHABILITADO
CONFIG PWRTE = ON ; POWER UP TIMER HABILITADO
CONFIG BOREN = ON ; RESET POR BAJO VOLTAJE
HABILITADO
CONFIG BORV = 20 ; VOLTAJE DE RESET = 2.0V
CONFIG WDTEN = OFF ; WATCHDOG DESHABILITADO
CONFIG HPOL = HIGH ; TRANSISTORES SUPERIORES
APAGADOS
; AL INICIO
CONFIG LPOL = HIGH ; TRANSISTORES INFERIORES
APAGADOS AL
; INICIO
CONFIG PWMPIN = ON ; SALIDAS PWM DESHABILITADO AL
RESET
CONFIG MCLRE = ON ; HABILITACIÓN DEL PIN MCLRE
COMO
; RESET MAESTRO
CONFIG LVP = OFF ; PROGRAMACIÓN EN BAJO VOLTAJE
(ICSP)
; DESHABILITADO

```

Fig. 9 Código de configuración y cabecera del microcontrolador.

Tomando en cuenta que la frecuencia de conmutación utilizada en el desarrollo de este trabajo es 10 kHz, y sustituyendo los valores provistos por los fabricantes del driver y del IGBT, para cada una de las variables especificadas en [8] para la ecuación 5, se obtiene el valor de capacitancia del capacitor de *bootstrap*.

$$C \geq \frac{2 \left[ 2(70nC) + \frac{125\mu A}{10kHz} + 5nC + \frac{4\mu A}{10kHz} \right]}{15V - 0.7V - 1V - 13V} = 315.8nF$$

Por lo que el capacitor debe tener un valor mayor a 315.8nF, prácticamente en [8] se recomienda que el valor del capacitor de *bootstrap* sea 15 veces el valor obtenido, por lo que el valor óptimo sería 4.77µF. En la realización de este proyecto el capacitor empleado es de 10 µF/50V. En lo que respecta al diodo de *bootstrap*, según lo establecido en la referencia [8], éste debe

tener un tiempo de recuperación inversa ( $t_{rr}$ ) no mayor a 100ns, soportar el voltaje de alimentación del inversor (voltaje del bus de CD) y la corriente de carga del circuito, la cual depende de la carga de compuerta del transistor y la frecuencia de conmutación. El diodo utilizado en este desarrollo es el modelo NTE574, el cual soporta hasta 400 Volts entre terminales, tiene un  $t_{rr}$  de 35ns y soporta una corriente de 1 Ampere; con lo que se cubren todos los requerimientos para el correcto funcionamiento del driver, para el desarrollo de este proyecto.

### E. Etapa digital

Finalmente la última etapa del hardware del sistema es la etapa digital, la cual se implementó mediante el microcontrolador PIC18F2431 de Microchip Technology. El sistema mínimo del microcontrolador se muestra en la figura 8; en dicha figura se observa que la etapa digital se encuentra constituida de manera relativamente sencilla, ya que solamente se precisa de un circuito oscilador como base de tiempo del microcontrolador y botones de funciones que pueden ser configurados mediante software por el usuario.

## III. IMPLEMENTACIÓN DE LA TÉCNICA DE MODULACIÓN SPWM

La técnica empleada para la generación de las señales de control del puente inversor trifásico es la técnica SPWM [3], [4] y [5], la cual se implementó en el microcontrolador PIC18F2431, haciendo uso de los recursos de control de motores propios del mismo [9], entre los que se pueden mencionar:

- 6 salidas PWM con 3 generadores de ciclo útil, cuyos pines pueden ser agrupados en pares.
- Ciclo útil de PWM con una resolución de 14 bits.
- Cambios de ciclo útil en plena ejecución de código.
- Modos de salida alineados a centro o a flanco.
- Control de tiempo muerto programable entre pares de salidas PWM.
- Convertidor ADC de 10 bits, alineado a izquierda o derecha.

La configuración del microcontrolador para llevar a cabo la tarea de control de motores se muestra en la figura 9, donde se muestra los aspectos a configurar antes de la ejecución del código principal.

Posteriormente se configura el convertidor analógico-digital, mediante el cual se proporciona la referencia para la frecuencia de salida de la componente fundamental del voltaje del inversor. El convertidor se configuró en modo de conversión continua y con el resultado alineado a la izquierda, figura 10.

Posteriormente se configura el módulo PWM para obtener la frecuencia de conmutación de salida de 10 kHz., un tiempo muerto de 500 ns entre pares de salidas complementadas y salidas alineadas por flanco. Dicho establecimiento de la frecuencia de conmutación de salida del módulo PWM, se realiza cargando un valor al registro PTPERH:PTPERL, en base a la ecuación 6.

$$PTPERH : PTPERL = \frac{F_{OSC}}{(F_{PWM} \square PTMR_{PS})} - 1 \quad (6)$$

De la misma manera el valor de tiempo muerto entre pares complementarios PWM se establece mediante la modificación del registro DTCON, en base a la ecuación 7.

$$Tiempo Muerto = \frac{Valor\ numérico\ de\ DTCON}{(F_{OSC} / Preescalador)} \quad (7)$$

La configuración de la totalidad del módulo PWM se completa mediante el establecimiento de la base del tiempo y de la respuesta a señales de falla o generación de señales debido a eventos configurables por el usuario. La totalidad del código de configuración del módulo PWM se muestra en la figura 11.

Un punto importante para el manejo del módulo PWM es el valor de ciclo útil que toma cada una de las salidas, mediante los registros de 14 bits denominados PDCxH:PDCxL mostrados en la figura 12; el número de bits que contribuyen al ciclo útil varía dependiendo de la frecuencia de salida de la señal PWM y de la frecuencia del cristal de oscilador. Tomando en cuenta los puntos anteriores la resolución disponible de los 14 bits mencionados, se encuentra dada por la ecuación 8.

$$Resolución\ (bits) = \frac{\log(\frac{F_{OSC}}{F_{PWM}})}{\log(2)} \quad (8)$$

Sin embargo, cabe mencionar que solo 12 bits realmente contienen el valor del ciclo útil, estos son los 6 bits menos

```

INICIA_PCPWM:
MOVW  b'00000000'      ; PTCON0 CONFIGURADO PARA QUE:
                        ; POSTESCALAMIENTO 1:1
MOVWF PTCON0           ; FRECUENCIA DE REFERENCIA
Fosc/4 Y               ; OPERACION ALINEADA POR
FLANCO                 ;
MOVW  0xF3             ; PTPERL Y PTPERH CARGADOS
PARA                   ;
10KHz                  ;
MOVWF PTPERL          ; PTPERH:PTPERL = ((Fosc/4)/(PWMfreq
x                      ; PTMRps))-1
MOVW  0x01             ; PTPERH:PTPERL = ((20MHz/4)/(10KHz
x                      ; )-1) = 499d = 1F3h
MOVWF PTPERH          ;
MOVW  b'01000000'      ; PWMCON0 CONFIGURADO DE TAL
                        ; MANERA QUE:
MOVWF PWMCON0         ; a) PWM0, PWM1, PWM2, PWM3,
                        ; PWM5 SON SALIDAS.
                        ; b) TODOS LOS PARES PWM ESTAN
EN                     ;
MOVW  b'00000001'      ; PWMCON1 CONFIGURADO DE TAL
                        ; MANERA QUE:
MOVWF PWMCON1         ; a) POSTESCALADO DE DISPARO DE
                        ; EVENTOS ESPECIALES 1:1
                        ; b) EVENTO ESPECIAL DE DISPARO
                        ; OCURRE EN CONTEO ASCENDENTE
                        ; c) ACTUALIZACION DE CICLO ÚTIL
Y                      ;
TIEMPO                 ; PERIODO HABILITADOS
                        ; d) IGNORAR SALIDAS
MOVW  b'00000101'      ; TIEMPO MUERTO DE 500 ns
MOVWF DTCON           ; TIEMPO MUERTO = VALOR DE
TIEMPO                ; MUERTO / (Fosc/2)
    
```

Fig. 11 Código de configuración del módulo PWM.

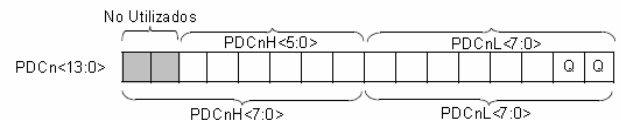


Fig. 12 Registros de ciclo útil PDCxH y PDCxL.

significativos de los registros PDCxH y los 6 bits más significativos de los PDCxL, los cuales se muestran en la figura 12. Así pues si se tuviese una resolución de 8 bits para una frecuencia de salida deseada y una frecuencia de reloj conocida, se utilizarían los 6 bits más significativos del registro PDCxL y los dos bits menos significativos del registro PDCxH, cabe mencionar que los dos bits menos significativos de PDCxL marcados como "Q" sincronizan los disparos de todos los canales con respecto a un reloj interno del módulo PWM, donde el valor de estos campos Q dependen del preescalador del módulo de PWM, el cual es establecido por el usuario pero no contribuye al valor de ciclo útil a la salida.

Para el desarrollo de este trabajo la resolución en bits fue de 9 bits, debido al reloj de 20 MHz y una frecuencia PWM de 10 KHz.

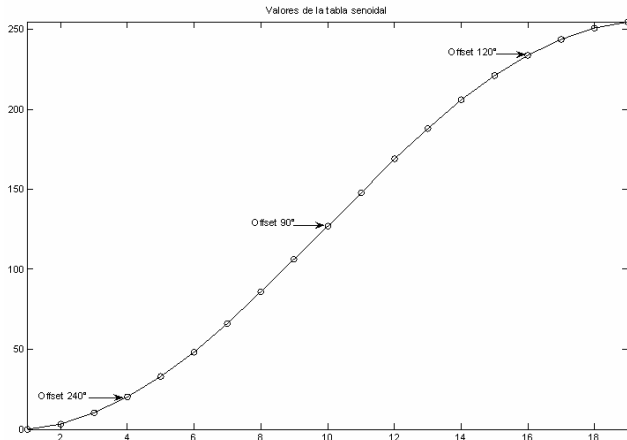


Fig. 13 Punteros en la tabla de valores senoidales para la generación del voltaje de salida del inversor.

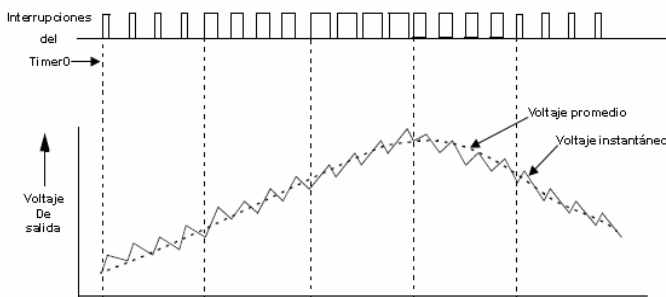


Fig. 14 Voltaje instantáneo y promedio generado mediante las interrupciones del timer0 y los valores de la tabla senoidal.

La implementación de la técnica SPWM se realiza haciendo uso de una tabla de valores que cambian senoidalmente, la cual se carga en la memoria del microcontrolador y se accede a ella cíclicamente; multiplicando este valor de la tabla por la frecuencia de salida se obtiene el valor de ciclo útil correspondiente a cada valor de la tabla, resultando en una variación senoidal del ciclo útil a la salida. Dado que el inversor es trifásico, se utilizan tres punteros desfasados 120 grados uno con respecto del otro.

Los valores de la tabla senoidal y los punteros en la misma se muestran en la figura 13. Dichos valores van desde 0 a 256, debido a que la variable utilizada es de 8 bits; para la realización de este proyecto se utilizaron 19 valores en la tabla senoidal, con lo que se obtiene una resolución de 10 grados por cada valor en la tabla senoidal y un total de 36 puntos para sintetizar un ciclo completo de la señal de salida.

La frecuencia de salida se proporciona mediante el uso del timer0 del microcontrolador, el cual interrumpe al terminar el ciclo útil y cambia al siguiente valor, para obtener el nuevo ciclo útil, el proceso de la generación del voltaje a la salida según las interrupciones del timer0 se muestra en la figura 14. Cada interrupción del timer0 corresponde a un valor de la tabla senoidal.

En la figura 14 se observa que para cada valor de ciclo útil se genera un voltaje instantáneo, y la suma de este voltaje a través del tiempo da el voltaje promedio de salida; el cual como se puede observar es del tipo senoidal. El semiciclo de voltaje de salida mostrado en la figura 14 corresponde al voltaje en un

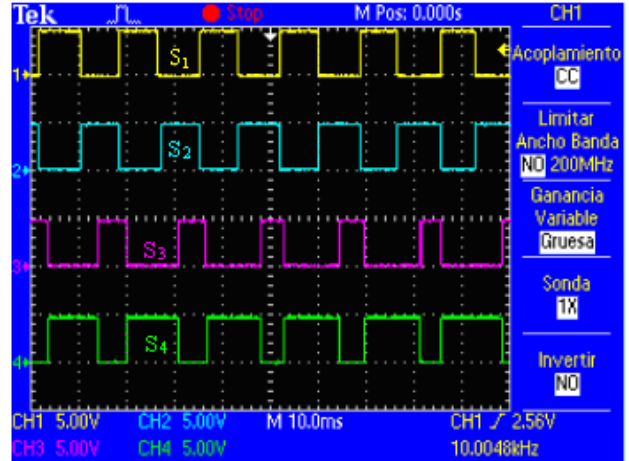


Fig. 15 Disparos en los transistores de dos ramas contiguas del inversor.

IGBT de una rama del puente inversor. La frecuencia de salida se encuentra dada por el temporizador timer0 del microcontrolador de la siguiente manera:

$$\text{Timer0} = \frac{F_{osc}}{2(NPT - 1) \times PT0 \times \text{Frec}} \quad (9)$$

Donde:

$F_{osc}$  = Frecuencia interna de trabajo del microcontrolador.

NPT = Número de puntos en la tabla senoidal.

PT0 = preescalador del Temporizador 0 (timer0).

Frec = Es el valor digitalizado proveniente del potenciómetro conectado al convertidor ADC.

Una variable que se desprende de la modulación SPWM es el índice de modulación en amplitud ( $m_a$ ), el cual repercute en el nivel de voltaje a la salida del inversor [5], dicha variable puede modificarse de manera relativamente sencilla en la implementación del código; esto se logra escalando los valores de la tabla senoidal cargada en memoria.

#### IV. PRUEBAS Y RESULTADOS

Después de la implementación de la modulación SPWM y del sistema en su totalidad; se muestran las pruebas realizadas al sistema, así como los resultados arrojados por las mismas. Se realizaron tres diferentes tipos de pruebas:

- Pruebas con carga resistiva conectada en estrella sin filtro a la salida del inversor.
- Pruebas con carga resistiva conectada en estrella con filtro a la salida del inversor.
- Pruebas con el motor de inducción trifásico sin filtro a la salida del inversor.

Las respuestas obtenidas se observaron utilizando un osciloscopio digital Tektronix modelo TDS2024 [10].

Cabe mencionar que antes de realizar las pruebas listadas, se verificaron las señales de control generadas por el

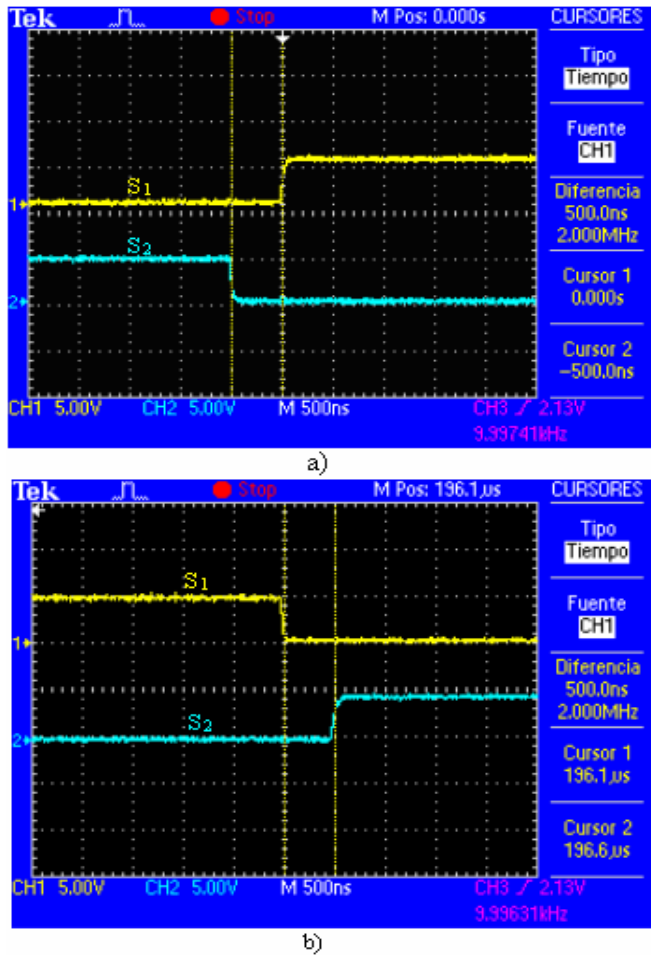


Fig. 16 Tiempo muerto entre señales de transistores de una misma rama: a) en el flanco de subida b) en el flanco de bajada (continuación).

microcontrolador para cada uno de los transistores; así como el tiempo muerto entre las señales de control de una misma rama del inversor. De esta manera se verificaron que las señales de control entre ramas del inversor estuvieran desfasadas 120 grados una con respecto de las otras; además, que las señales de control entre transistores de una misma rama se encontraran complementadas y existiera entre éstas el tiempo muerto establecido.

La figura 15 muestra las señales de control generadas correspondientes a los transistores de dos ramas contiguas del inversor; el par de señales superior corresponden a los transistores  $S_1$  y  $S_2$  y el par inferior a  $S_3$  y  $S_4$ , del puente inversor de dos niveles mostrado anteriormente en la figura 3.

En las figuras 16 a) y b), se observa el tiempo muerto entre los transistores  $S_1$  y  $S_2$  de una rama del inversor; para el flanco de subida y de bajada respectivamente. Para el caso de las dos figuras se observa que el tiempo muerto es de 500 ns en ambos flancos.

*Pruebas con carga resistiva en estrella sin filtro de salida.*

Las pruebas con carga resistiva conectada en estrella se realizaron utilizando el diagrama de conexiones mostrado en la figura 17 a), bajo las siguientes condiciones de prueba:

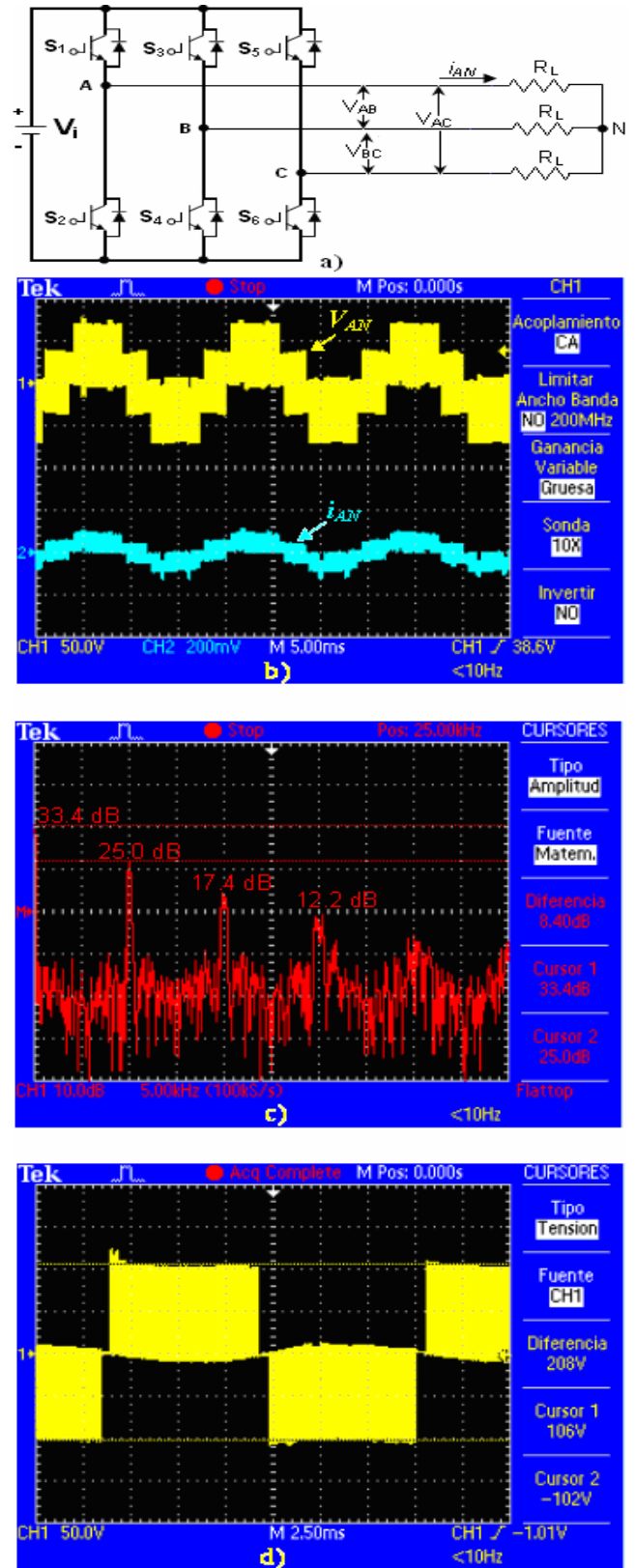


Fig. 17 a) Diagrama esquemático del inversor con carga resistiva en estrella sin filtro de salida. b) Voltaje de fase ( $V_{AN}$ ) y corriente de fase ( $I_{AN}$ ). c) Distribución armónica del voltaje de fase ( $V_{AN}$ ). d) Voltaje de línea ( $V_{AB}$ ).

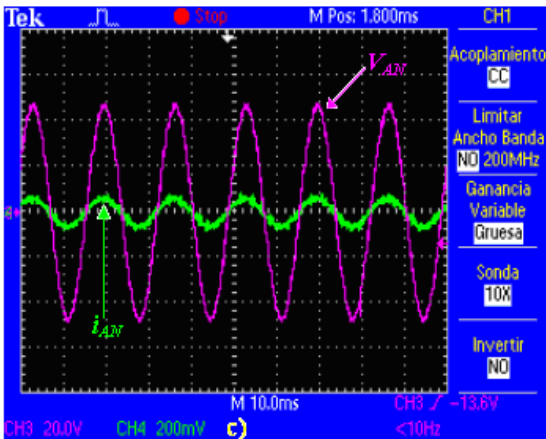
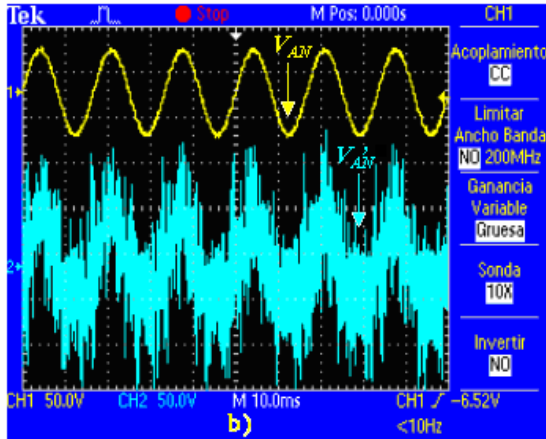
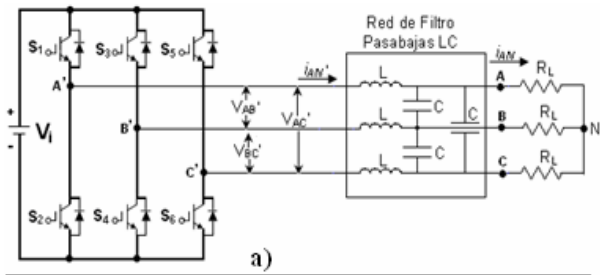


Fig. 18 a) Diagrama esquemático del inversor con carga resistiva en estrella con filtro de salida. b) Voltaje de fase antes del filtro ( $V_{AN}$ ) y después del filtro ( $V_{AN'}$ ). c) Voltaje de fase y corriente de fase ( $i_{AN}$ ).

- Voltaje de entrada ( $V_i$ ) = 100 Volts.
- Frecuencia de conmutación = 10 kHz.
- Tiempo muerto = 500 nS.
- Carga resistiva  $R_L = 75\Omega$ .
- Índice de modulación en amplitud  $m_a = 1$ .

En la figura 17 b) se observan las formas de onda del voltaje de fase ( $V_{AN}$ ) y la corriente de fase ( $i_{AN}$ ).

En la figura 17 c) se observa la distribución armónica del voltaje de fase y la magnitud en decibelios para la frecuencia de conmutación y para cada armónico múltiplo de la frecuencia de conmutación.

Finalmente en la figura 17 d) se observa la forma de onda del voltaje de línea ( $V_{AB}$ ).

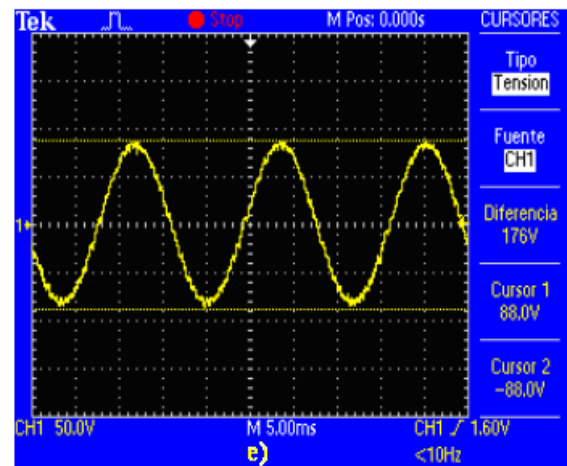
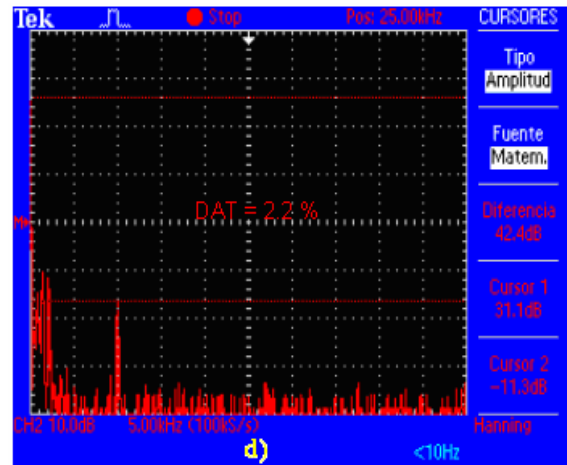


Fig. 18 d) Distribución armónica del voltaje de fase ( $V_{AN}$ ) después del filtrado. e) Voltaje de línea ( $V_{AB}$ ) después del filtrado. (Continuación).

#### Pruebas con carga resistiva en estrella con filtro de salida

Las pruebas realizadas utilizando un filtro a la salida del inversor se realizaron utilizando un filtro pasabajas LC de segundo orden diseñado para tener una frecuencia de corte de 2 kHz. El diagrama de conexiones del filtro de salida con la carga se observa en la figura 18 a). Las pruebas se realizaron bajo las siguientes condiciones de prueba:

- Voltaje de entrada ( $V_i$ ) = 100 Volts.
- Frecuencia de conmutación = 10 kHz.
- Tiempo muerto = 500 nS.
- Carga resistiva  $R_L = 75\Omega$ .
- Índice de modulación en amplitud  $m_a = 1$ .
- Inductancia del filtro de salida = 1.5288mH.
- Capacitancia del filtro de salida = 10 $\mu$ F.

La figura 18 b) muestra la forma de onda del voltaje antes ( $V_{AN}$ ) y después del filtro de salida ( $V_{AN}$ ). La figura 18 c) muestra las formas de onda del voltaje de fase ( $V_{AN}$ ) y la corriente de fase ( $i_{AN}$ ), donde se puede observar que ambas están en fase.



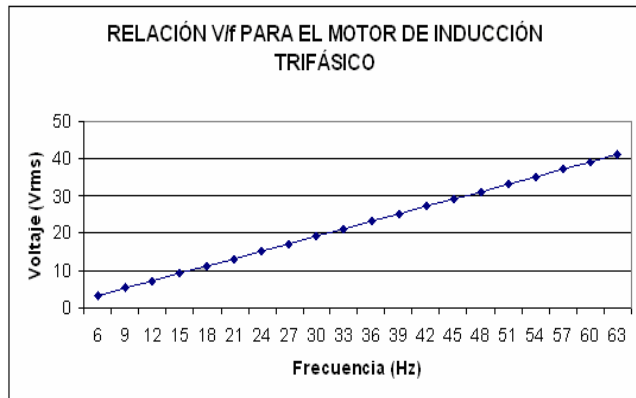


Fig. 19 Relación  $V/f$  a la salida del inversor con el motor de inducción como carga.

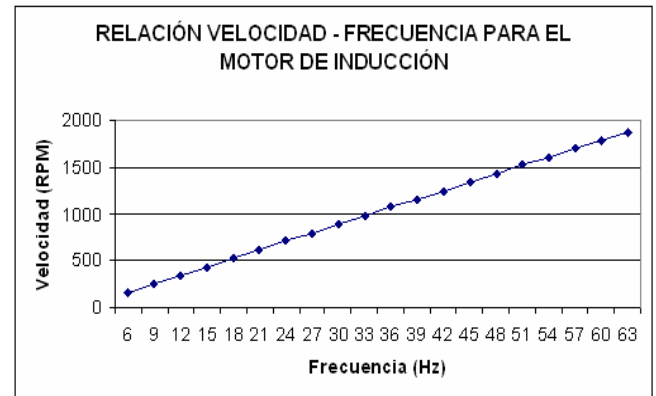


Fig. 20 Relación Velocidad/frecuencia para el motor de inducción trifásico, utilizado como carga a la salida.

La distribución armónica del voltaje de fase ( $V_{AN}$ ) se observa en la figura 18 d), donde la distorsión armónica medida con el analizador de potencia FLUKE 43B es de 2.2 %. La figura 18 e) muestra la forma de onda del voltaje de línea ( $V_{AB}$ ).

Un aspecto importante que cabe mencionar es el tipo de capacitor utilizado en el filtro de salida, los cuales deben ser capacitores no polarizados, se recomienda que sean de un material dieléctrico que proporcione una dinámica rápida, con la finalidad de obtener un mejor filtrado de las componentes de alta frecuencia correspondientes a la frecuencia de conmutación de los transistores del inversor.

#### Pruebas con el motor de inducción trifásico sin filtro a la salida del inversor

Las pruebas realizadas con el motor de inducción trifásico consistieron en la obtención de la característica  $V/f$  de salida y la relación voltaje-velocidad. Las condiciones de prueba utilizando al motor de inducción como carga, sin filtro de salida, son:

- Voltaje de entrada ( $V_i$ ) = 120 Volts.
- Frecuencia de conmutación = 10 kHz.
- Tiempo muerto = 1  $\mu$ S.
- Índice de modulación en amplitud  $m_a=1$ .
- Potencia del motor = 1/3 HP.
- Conexión de los devanados del motor en estrella (Y).
- Voltaje máximo de trabajo del motor = 220V/440V
- Corriente máxima de trabajo del motor = 3.3 Amps.

Realizando mediciones del valor de voltaje a la salida y la frecuencia de dicho voltaje para cada fase se obtuvo la relación mostrada en la gráfica de la figura 19, donde se observa que la relación  $V/f$  es constante en casi todo el rango de operación del motor de inducción.

Realizando mediciones de la velocidad de giro del motor y la frecuencia del voltaje de salida para cada velocidad se obtuvo la relación mostrada en la gráfica de la figura 20; en esta se observa que la relación Velocidad-frecuencia es constante en casi todo el rango de operación del motor de inducción, al igual que la relación  $V/f$ .

## V. CONCLUSIONES

Mediante la realización de este trabajo presenta una alternativa para el diseño de un inversor de dos niveles, principalmente enfocado a la comunidad estudiantil de ingeniería; además, paralelamente, se cumplió con el objetivo de llevar a cabo la implementación del mismo y variar la velocidad del motor de inducción manteniendo el par del motor constante en el rango de operación del mismo. Igualmente se llevó a cabo la implementación y desarrollo de la técnica de modulación SPWM en el microcontrolador PIC18F2431, asimismo se abordaron las implicaciones de esta técnica en lo referente a su aplicación a este tipo de trabajos, específicamente en lo referente al microcontrolador utilizado.

Recurriendo a este trabajo como base, es posible aumentar la complejidad de la topología del inversor, voltajes y corrientes de trabajo; así como la técnica de modulación empleada. El sistema queda listo para la aplicación de técnicas y algoritmos de control.

## RECONOCIMIENTOS

Se agradece al Ing. José Luis Flores Nolasco y a Leonardo Samayoa Veliz, por su colaboración en el desarrollo de este trabajo.

## REFERENCIAS

- [1] Muhammad H. Rashid, *Electrónica de potencia. Circuitos, Dispositivos y Aplicaciones*, Editorial Pearson Educación, Segunda edición, 1995.
- [1] Chapman Stephen, *Máquinas Eléctricas*, Editorial McGraw-Hill, Tercera edición, 2000.
- [2] Mohan Ned, Undeland Tore M., Robbins William P., *Power electronics. Converters, Applications and Design*, Editorial John Wiley & Sons, INC., Segunda edición, 1995.
- [3] Bose Bimal K., *Modern Power Electronics and AC drives*, Editorial Prentice Hall, 2001.
- [4] Skvarenina Timothy L., *The Power Electronics Handbook*, Editorial CRC Press, 2001.
- [5] Arizona Microchip Technology, Application Note: 900: Controlling 3-Phase AC Induction Motors Using the PIC18F4431, [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&noDel=1824&appnote=en020394](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&noDel=1824&appnote=en020394), consultada en Julio de 2006.
- [6] Fluke Electronics, Fluke 43B. Analizador eléctrico avanzado. Manual de uso.
- [7] International Rectifier, Design Tip DT98-2a: Bootstrap Component Selection for Control IC's, <http://www.irf.com/technical-info/designtip/dt98-2.pdf>, consultada en Julio de 2006.

- [8] Arizona Microchip Technology; PIC18F2331/2431/4331/4431 Datasheet, <http://ww1.microchip.com/downloads/en/DeviceDoc/39616b.pdf>, consultada en Julio de 2006.
- [9] Tektronix, Osciloscopio de almacenamiento digital. Series TDS1000 y TDS2000. Manual del usuario.



**Aganza T. Alejandro** nació el 28 de Marzo de 1983 en Ciudad Obregón, Sonora, México. Obtuvo el grado de Ingeniero en Electrónica con acentuación en Instrumentación y Control en el Instituto Tecnológico de Sonora en Agosto de 2006. Actualmente se desempeña como consultor y diseñador de sistemas electrónicos. Sus áreas de interés son: Análisis, diseño, modelado e implementación de sistemas electrónicos de potencia y teoría de control aplicada a los mismos.

**Pérez R. Javier**, nació el 2 de Febrero de 1973. Obtuvo el título de Lic. en electrónica en la Universidad Autónoma de Puebla en 1999, el grado de Maestro en Ciencias en Ing. Electrónica en el Centro Nacional de Investigación y Desarrollo Tecnológico, en Cuernavaca Morelos, en el 2000.

Del 2000 a la fecha labora como profesor de tiempo completo en el Instituto Tecnológico de Sonora, en Ciudad Obregón, Sonora, México. Sus áreas de interés son: control automático, procesamiento de imágenes y electrónica de potencia.



**Beristain J. José Antonio**, obtuvo el grado de Maestro en Ciencias en el centro nacional de investigación y desarrollo tecnológico, CENIDET, y el grado de Doctor por la Universidad Politécnica de Cataluña, actualmente labora en el Instituto Tecnológico de Sonora como Maestro-Investigador de tiempo completo.

# Implementación de un controlador adaptativo para un sistema de primer orden con ganancia y polo variantes, basado en DSP

Pérez R. Javier, Pelayo S. Roger Pelayo y Manzanarez A. Francisco

**Resumen**— El presente artículo expone la implementación de un controlador adaptativo, basado en un modelo de referencia, aplicado a un proceso de primer orden con polo y ganancia variantes. La implementación se llevó a cabo con un procesador digital de señales (DSP) TMS320C31 de Texas Instruments, ensamblado en una tarjeta TMS320C3x DSP Starter Kit (DSK). En este caso se controla la velocidad de un motor de corriente continua de 115V y 3.2A nominales. También, se muestra a detalle el desarrollo y análisis matemático, basado en la regla del MIT, así como la programación que se llevó a cabo en el DSP. Finalmente, a través de los resultados, se hace una evaluación práctica del controlador mostrando las bondades y limitaciones de la implementación.

**Palabras clave**—Control adaptativo, DSP, modelo de referencia, regla del MIT.

## I. INTRODUCCIÓN

Un controlador adaptativo, es un controlador con parámetros ajustables y un mecanismo para ajustarlos, lo cual permite su aplicación en sistemas variantes en el tiempo [1]. En este caso el comportamiento de la velocidad de un motor de CD, se aproxima a un sistema de primer orden y al incluir un sensor de velocidad no lineal podemos observar un comportamiento como el de la figura 1. De aquí que si nosotros cambiamos el punto de operación, el modelo de nuestro sistema de primer orden cambia tanto su ganancia como su polo y entonces con esto es posible simular un sistema variante en el tiempo. Aquí es importante hacer notar que no se realizará un análisis matemático de la no linealidad del sistema, únicamente se utilizará dicha no linealidad de manera práctica para mostrar el rango de funcionamiento del controlador adaptativo.

Por otro lado es importante mencionar que se han tomado como base los trabajos presentados en [3] y [8] en los cuales ya se han implementado controladores adaptativos solo que en estos trabajos solo se consideraron variaciones en la ganancia, mientras que este caso se consideraran variaciones tanto en la ganancia como en el polo de la planta de primer orden.

Manuscrito recibido el 4 de Julio de 2006 Este trabajo fue respaldado por el departamento de Ing. Eléctrica y Electrónica del Instituto Tecnológico de Sonora

Pérez R. Javier hasta la fecha se ha desempeñado como Profesor de Tiempo Completo del Instituto Tecnológico de Sonora en el Departamento de Ingeniería Eléctrica y Electrónica Instituto Tecnológico de Sonora; Ave. Antonio Caso S/N Col. Villa ITSON; Ciudad Obregón, Sonora, México; C.P. 85138; C.P. 85130; Tel: (644) 4109000, ext. 101; Fax: (644) 4109001.(e-mail [jperez@itson.mx](mailto:jperez@itson.mx))

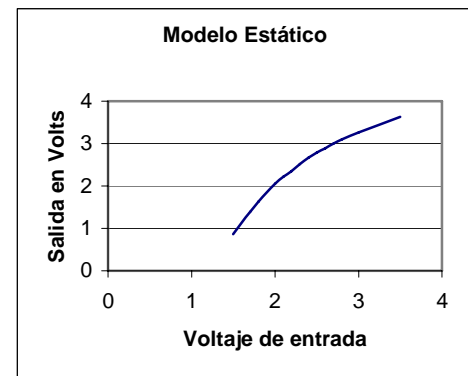


Fig. 1 Modelo estático de la velocidad del motor de CD incluyendo un sensor no lineal con salida de voltaje[8].

### Planta de primer orden

Como planta de primer orden, se empleó un motor de corriente continua, marca Baldor, con número de serie 475 de Baldor Electric CO, con voltaje de hasta 115V y 3.2A nominales, con una velocidad máxima de 1750 rpm. En [8] se puede ver a detalle como se implementó la parte del actuador con un circuito integrado que genera el ancho de pulso que utiliza un mosfet para dar el voltaje requerido al motor; ahí mismo se puede ver a detalle como se implementó la parte de sensado de velocidad con un fototransistor y un convertidor de frecuencia a voltaje.

### Modelo Estático

El modelo estático nos permite saber que salida debemos esperar de la planta, una vez que ésta se encuentra en estado estable, dependiendo de la entrada que se le aplique. La figura 1 muestra la gráfica resultante del modelo estático, la cual muestra un comportamiento no lineal.

### Modelo Dinámico

Para la obtención del modelo dinámico, se aplicó un escalón de 2 volts de entrada a la planta, graficando la salida con el osciloscopio [2] se obtuvo la respuesta mostrada en la figura 2.

Con base a la respuesta obtenida, el modelo de la planta queda definido por la ecuación 1, [5].

$$G(s) = \frac{(1.01)(1.4)}{1.4s + 1} \quad (1)$$

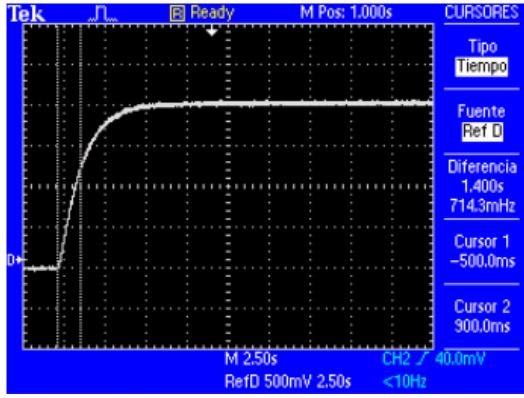


Fig. 2 Respuesta al escalón de la planta.

### Modelo de Referencia

El mecanismo para el ajuste de parámetros, de este tipo de controlador, se basa en un modelo de referencia mediante el cual se especifica el funcionamiento deseado. En este caso el modelo representa el comportamiento ideal de un proceso de primer orden, al cual se desea aproximar el comportamiento del sistema una vez que éste se exponga a diversas perturbaciones.

## II. DISEÑO

El diseño del controlador está basado en la regla del MIT [1] y se escoge, por ser una de las más típicamente empleadas, el error medio cuadrático como función de costo, ecuación 2.

$$J(\theta) = \left(\frac{1}{2}\right)e^2 \quad (2)$$

En base a la ecuación 2, es posible realizar el ajuste de parámetros de una manera tal que se minimice  $J$ , para lo cual se cambian los parámetros en dirección a su gradiente negativo tal como se aprecia en la ecuación 3, la cual es conocida como la regla del MIT.

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta} \quad (3)$$

### Aplicación de la regla del MIT para un sistema de primer orden con variación en la ganancia y el polo

Sea el sistema de primer orden representado por la ecuación 4, o descrito por la función de transferencia en el dominio de Laplace de la ecuación 5.

$$\frac{dy}{dt} = -ay + bu \quad (4)$$

$$\frac{y}{u} = \frac{b}{s+a} \quad (5)$$

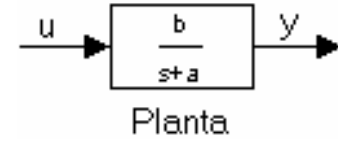


Fig. 3 Función de transferencia de la planta.

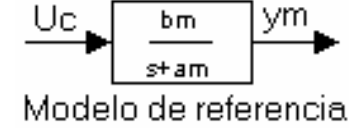


Fig. 4 Función de transferencia del Modelo de referencia

Donde  $u$  es la variable de control,  $y$  es la salida del sistema,  $b$  la ganancia y  $a$  el polo del sistema, se requiere que el sistema de la figura 3 tenga el comportamiento del modelo de la ecuación 6.

$$\frac{dy_m}{dt} = -a_my_m + b_mu_c \quad (6)$$

$$\frac{y_m}{u_c} = \frac{b_m}{s+a_m} \quad (7)$$

Como se desea que el comportamiento de la planta real, se aproxime al comportamiento del modelo, si  $y = y_m$ , entonces:

$$0 = \frac{dy}{dt} - \frac{dy_m}{dt} = -ay + bu - (-a_my_m + b_mu_c)$$

y por lo tanto

$$0 = bu - b_mu_c + (a_m - a)y$$

$$u = \frac{b_m}{b}u_c - \frac{a_m - a}{b}y \quad (8)$$

y como  $a$  y  $b$ , son los dos parámetros que varían, podemos escoger  $\theta_1 = b_m/b$  y  $\theta_2 = (a_m - a)/b$ , quedando la ecuación 8 de la siguiente forma:

$$u = \theta_1 u_c - \theta_2 y \quad (9)$$

Para poder aplicar la regla del MIT, hay que dejar al error en función de los parámetros  $\theta$

$$e = y - y_m \quad (10)$$

con base en las ecuaciones 4 y 9, la salida  $y$  queda como:

$$py = -ay + bu = -ay + b(\theta_1 u_c - \theta_2 y)$$

$$y = \frac{b\theta_1}{p + a + b\theta_2} u_c \quad (11)$$

donde  $p=d/dt$ , es el operador diferencial. La ecuación 11 es la misma para  $y_m$ , por lo tanto las relaciones de entrada-salida del sistema y el modelo son las mismas, lo cual se conoce como un seguimiento perfecto del modelo. Las derivadas sensitivas son obtenidas a partir de las derivadas parciales con respecto a los parámetros del controlador,  $\theta_1$  y  $\theta_2$ .

$$\frac{\partial e}{\partial \theta_1} = \frac{b}{p + a + b\theta_2} u_c \quad (12)$$

$$\frac{\partial e}{\partial \theta_2} = -\frac{b^2\theta_1}{(p + a + b\theta_2)^2} u_c$$

Y utilizando (11)

$$\frac{\partial e}{\partial \theta_2} = -\frac{b}{p + a + b\theta_2} y \quad (13)$$

Estas ecuaciones, no pueden usarse directamente, debido a que los parámetros  $a$  y  $b$  son desconocidos ya que son variantes en el tiempo, por lo tanto se requiere hacer alguna aproximación. Una aproximación posible se puede obtener a partir de que  $p + a + b\theta_2 = p + a_m$  cuando los parámetros permiten un perfecto seguimiento del modelo de referencia, de tal forma que se puede aproximar diciendo que  $p + a + b\theta_2 \approx p + a_m$ , lo cual es razonable cuando los parámetros se acercan a sus valores correctos. Con esta aproximación se aplica la regla del MIT y por comodidad repetimos la ecuación 3:

$$\frac{d\theta}{dt} = -\gamma \frac{\partial J}{\partial \theta} = -\gamma e \frac{\partial e}{\partial \theta}$$

obteniendo así las siguientes ecuaciones:

$$\frac{d\theta_1}{dt} = -\gamma \frac{a_m}{p + a_m} u_c e \quad (14)$$

$$\frac{d\theta_2}{dt} = -\gamma \frac{a_m}{p + a_m} y e \quad (15)$$

donde se propone que  $\gamma = \gamma'(b/a_m)$ .

Al aplicar la transformada de Laplace a las ecuaciones 14 y 15, se obtienen las ecuaciones 16 y 17, que se representan en los esquemas de la figura 5.

$$\theta_1 = \frac{-\gamma}{s} \left( \frac{a_m}{s + a_m} \right) u_c e \quad (16)$$

$$\theta_2 = \frac{\gamma}{s} \left( \frac{a_m}{s + a_m} \right) y e \quad (17)$$

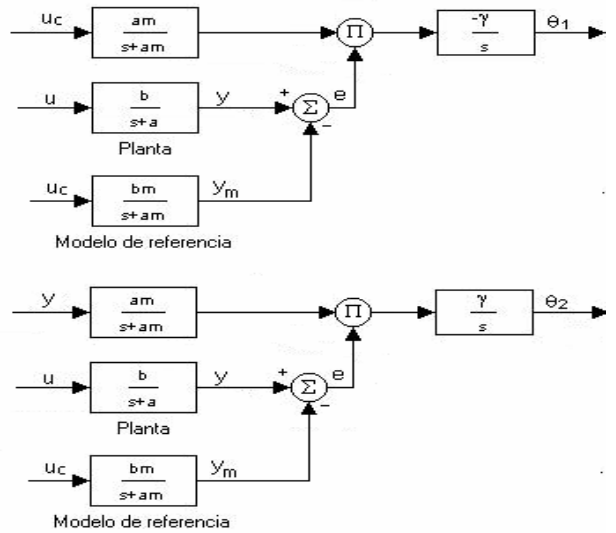


Fig. 5 Esquemas de los parámetros  $\theta_1$  y  $\theta_2$ .

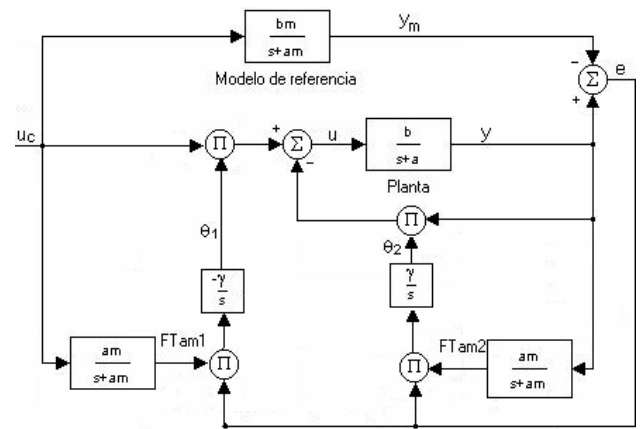


Fig. 6 Esquema completo del diseño de un controlador basado en el modelo de referencia para un sistema de primer orden.

Los esquemas de la figura 5 junto con las ecuaciones 9 y 10, y las figuras 3 y 4, forman el diagrama a bloques de la figura 6.

#### Digitalización del Modelo de referencia

Considerando que nuestro modelo de referencia de 1er orden, se encuentra definido por la siguiente función de transferencia en el dominio de Laplace [5].

$$G_m(s) = \frac{1.01}{s + \frac{1}{1.4}} \quad (18)$$

Se aplicó la transformada Z, para un tiempo de  $T=100 \mu\text{seg.}$ , quedando la función del modelo de la siguiente forma.

$$G_m(z) = \frac{y_m(z)}{U_c(z)} = k \frac{1.01z}{z + \frac{1}{1.4}}$$

Para realizar esta transformación se utilizó el método de igualación polo-cero modificado [4]. El valor de  $k$  se determina haciendo que  $G_m(s)=G_m(z)$ , cuando  $s$  tiende a cero y  $z$  tiende a uno, mismo valor que se muestra a continuación.

$$k = (1 - e^{-1.4T}) \quad (1.4)$$

Por lo tanto, la función de transferencia del modelo en el dominio de  $Z$ , viene quedando de esta manera.

$$\frac{Y_m(z)}{U_c(z)} = \frac{0.0001}{1 - 0.9999z^{-1}} \quad (19)$$

$$Y_m(z) = 0.0001U_c(z) + Y_m(z)0.9999z^{-1}$$

Finalmente al pasar al dominio del tiempo, se obtiene la siguiente ecuación en diferencias 20.

$$Y_m(t) = 0.0001U_c(t) + 0.9999Y_m(t-1) \quad (20)$$

#### Digitalización del control

Tal como se puede ver en la figura 6, se tienen FTam1 y FTam2, quienes están definidas por la misma función de transferencia bajo el dominio de  $S$ .

$$FTam_{1,2}(s) = \frac{a_m}{s + a_m}$$

La única diferencia entre estas dos funciones, son sus respectivas entradas, ya que para FTam1, la entrada por la que multiplica viene siendo la  $u_c$ , mientras que para FTam2, viene siendo la  $y$ , de tal forma que al pasarlas al dominio de  $Z$ , aplicando el mismo método que se empleó para el modelo de referencia y tomando en cuenta que el polo  $a_m=1/1.4$ , se obtuvieron las respectivas ecuaciones en diferencias 21 y 22.

$$FTam1(t) = 0.0007143 U_c(t) + 0.9999 FTam1(t-1) \quad (21)$$

$$FTam2(t) = 0.0007143 Y(t) + 0.9999 FTam2(t-1) \quad (22)$$

Para el caso de los parámetros del controlador,  $\theta_1$  y  $\theta_2$ , las cuales, basándose en lo que se observa en la figura 6, están definidas por las ecuaciones 23 y 24, mismas que al transformarse de forma directa al dominio de  $Z$ , se obtienen también de éstas, sus respectivas ecuaciones en diferencias.

$$\theta_1(s) = \frac{-\gamma FTam1(s)e}{s} \quad (23)$$

$$\theta_2(s) = \frac{\gamma FTam2(s)e}{s} \quad (24)$$

<pre>ent .float 10000 const1 .float 0.0001 const2 .float 0.9999 sal_ant .float 0 sal_act .int 0</pre> <p style="text-align: center;">b)</p>	<pre>----- ldp @const1 ldf @const1, r5 ldf @const2, r2 mpyf @ent, r5 mpyf @sal_ant, r2 addf r5, r2 stf r2, @sal_ant fix r2, r3 sti r3, @sal_act ----- a)</pre>
---	--

Fig. 7 a) Codificación del modelo de referencia en el DSP; b) Inicialización de variables.

$$\theta_1(z) = \frac{-\gamma FTam1(z)e}{1 - z^{-1}}$$

$$\theta_1(t) = -\gamma FTam1(t)e + \theta_1(t-1) \quad (25)$$

$$\theta_2(z) = \frac{-\gamma FTam2(z)e}{1 - z^{-1}}$$

$$\theta_2(t) = -\gamma FTam2(t)e + \theta_2(t-1) \quad (26)$$

Por último, la señal de control ( $u$ ) ya se puede obtener como se encuentra definida en la ecuación (9).

Hasta aquí ya se tiene toda la parte previa a la implementación y programación del control adaptativo para el DSP, el cual se lleva a cabo con ayuda de las ecuaciones 20, 21, 22, 25, 26 y 9, junto con el esquema de la figura 6.

### III. IMPLEMENTACIÓN

El controlador se implementó en un Procesador Digital de Señales (DSP) de punto flotante, Modelo TMS320C31 de Texas Instruments, que se encuentra ensamblado en una tarjeta TMS320C3x DSP Starter Kit (DSK), la cual cuenta con un ADC y un DAC con salida y entrada RS232 respectivamente[6][7]. La única entrada analógica del ADC se utilizó para capturar la salida de la planta ( $y$ ), mientras que la única salida analógica del DAC, se empleó para la transmisión de la señal de control ( $u$ ).

En primer lugar se validó el modelo de referencia presentado en la ecuación 1, el cual se implementó en el DSP empleando su correspondiente ecuación en diferencias, ecuación 20. La codificación en ensamblador requerida para este modelo, se muestra en la figura 7.

La figura 8 presenta la respuesta al escalón tanto del motor como del modelo de referencia implementado en el DSP con lo cual se corrobora que el modelo en ecuación en diferencias es correcto.

Tomando en cuenta que la programación del control adaptativo expuesto en este artículo, está basado en el esquema de la figura 6 y en las ecuaciones 20, 21, 22, 25, 26 y 9; se expone, a excepción de la programación del modelo que ya se mostró, toda la codificación necesaria que se realizó en el DSP para la ejecución de estas ecuaciones, se encuentran listadas en la figura 9.

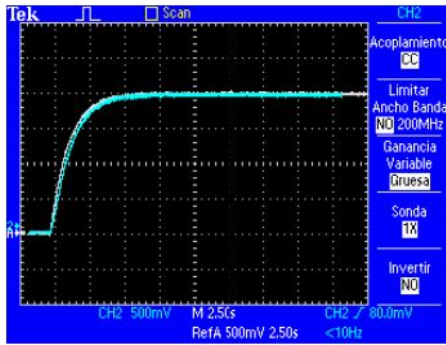


Fig. 8 Respuestas del motor CD (curva azul) y del modelo implementado en el DSP (curva blanca), al mismo escalón.

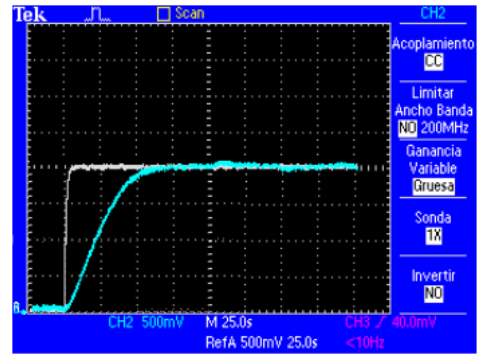


Fig. 10 Respuesta del sistema con un valor de gamma de 0.3e-14.

<pre> Segmento que se encarga de la captura de la salida de la planta y de su conversión a flotante ;----- ldi @S0_rdata,R3 float r3,r2 mpyf 0.0000154958,r2 fix r2,r3 sti r3,@sal_real ldi @sal_real,r7 float r7,r1 ;----- Segmento que se encarga del cálculo del error, de la ecuación 22, y de <math>\theta_2</math>. ;----- subi @sal_act,r7 float r7,r6 stf r6,@error ldf @cam1,r5 ldf @cam2,r2 mpyf r1,r5 mpyf @amant,r2 addf r5,r2 stf r2,@amant mpyf @error,r2 mpyf @gama,r2 addf @teta2ant,r2 stf r2,@teta2ant a)                 </pre>	<pre> Segmento que se encarga del calculo de la ecuación 21 y de <math>\theta_1</math>. ;----- ldf @cam1,r5 ldf @cam2,r2 mpyf @ent,r5 mpyf @bamant,r2 addf r5,r2 stf r2,@bamant mpyf @error,r2 mpyf @gama,r2 negf r2,r2 addf @teta1ant,r2 stf r2,@teta1ant ;----- Segmento que se encarga del cálculo de la ecuación 9 (u). ;----- mpyf @ent,r2 mpyf @teta2ant,r1 subf r1,r2 fix r2,r3 sti r3,@u b)                 </pre>
--	--

Fig. 9 a) Codificación para obtener el error y  $\theta_2$ ; b) codificación para obtener  $\theta_1$  y u.

La idea principal de las figuras 9 y 10 es mostrar la sencillez de la implementación del controlador en el DSP.

#### IV. RESULTADOS

Después de haber realizado la programación del controlador en el DSP, en este apartado se muestran los resultados de la implementación. Se hicieron tres grupos de pruebas: a) pruebas para encontrar la gama adecuada; b) pruebas variando la ganancia del modelo de referencia y c) pruebas variando el polo del modelo de referencia.

##### Pruebas con diferentes valores de gamma

Se realizaron diversas pruebas para encontrar un valor de gama para el cual el controlador pudiera seguir al modelo de referencia dado en la ecuación (1).

Las figuras 10, 11 y 12 muestran la repuesta del sistema en lazo cerrado para diferentes valores de gamma si la gama es pequeña la respuesta es lenta y si la gamma es grande el sistema se vuelve oscilatorio. En este caso el valor de gamma se estableció de manera heurística de a cuerdo a la experiencia que

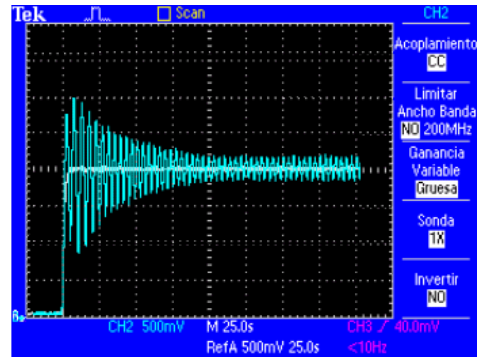


Fig. 11 Respuesta del sistema a una gama de 30e-14.

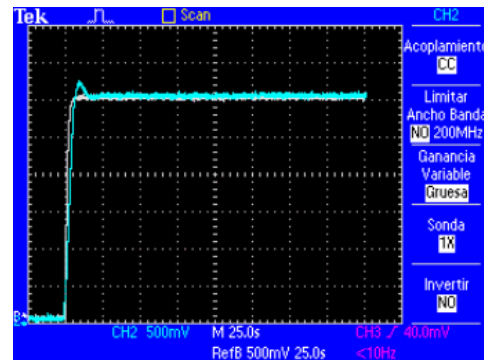


Fig. 12 Respuesta del sistema a una gama de 3e-14.

ya se tenía en [3] y el valor con el que mejor desempeño tuvo fue de 3e-14. En [1] se presenta una forma de obtener una primera aproximación de gamma, la cual depende de  $U_c$ ; sin embargo, no está considerada en los alcances de este trabajo.

##### Pruebas cambiando la ganancia

Las pruebas anteriores sólo sirvieron para encontrar la gamma adecuada pero no se hizo ninguna variación ni en la ganancia ni en el polo que es para lo que realmente fue diseñado este tipo de control.

En este caso se realizaron diferentes variaciones a la ganancia del modelo  $b_m$  y el control respondió manteniendo un sobreimpulso menor al 20% para variaciones en el rango de 0.3b a 4b. La figura 13 muestra una de las pruebas con una variación

de 1.5b. Cabe hacer notar que para variaciones fuera del rango mencionado se tienen sobreimpulsos muy grandes sin embargo siempre alcanza el estado estable.

### Pruebas cambiando el polo

Las terceras y últimas pruebas, se realizaron variando al polo del modelo de referencia, obteniendo las respuestas, que se muestran a continuación.

En las figuras 14 y 15 se muestra, como al variar la  $\tau$ , a diferentes tiempos de 1.4 seg, se sigue dando un perfecto seguimiento del modelo, sólo que la única diferencia es que al usar una  $\tau$  mayor (ver figura 14), se presenta un sobreimpulso menor al que se genera empleando la  $\tau$  con valor de 1.4 seg. (ver figura 12), mientras que al utilizar una  $\tau$  menor (ver figura 15), se da un sobreimpulso mayor. La magnitud del sobreimpulso aumenta al mismo tiempo que aumenta el valor del polo.

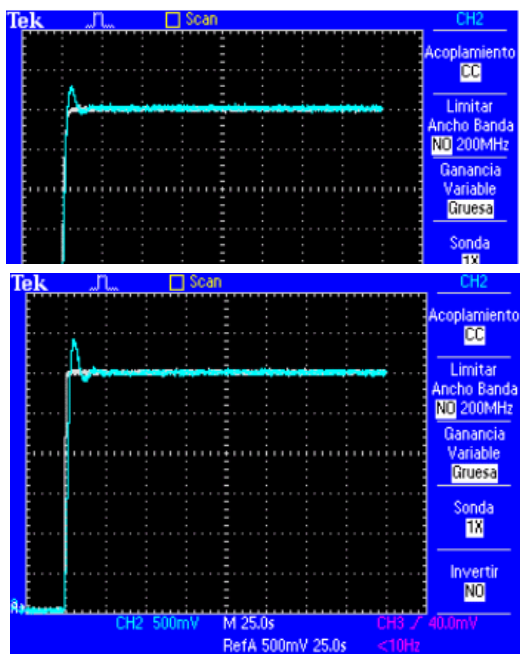


Fig. 15 Respuestas del modelo y de la planta a una  $\tau$  de 0.4 seg.

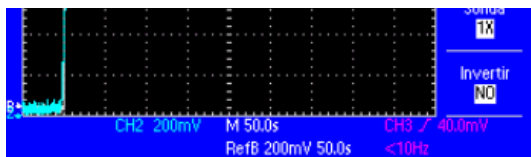


Fig. 14 Respuestas del modelo y de la planta a una  $\tau$  de 2.4 seg.

Se hicieron pruebas con perturbaciones variando la carga y también trabajando a diferentes puntos de operación y en todos los casos el controlador siempre ha seguido al modelo de referencia. Haciendo cambios bruscos en la referencia el controlador tiene problemas en el estado transitorio ya que el sobreimpulso llega a ser muy grande, sin embargo siempre llega a estabilizarse en el nivel de referencia deseado.

## V. CONCLUSIONES

La digitalización de un control adaptativo basado en un modelo de referencia de primer orden, puede llevarse a cabo en cualquier microprocesador, mas sin embargo, el grado de dificultad para su realización varía dependiendo del microprocesador que se emplee, ya sea por las limitantes que éste conlleve o por la complejidad de su propio lenguaje de programación, pero independiente de esto, a pesar de que el control adaptativo es considerado como un esquema de control avanzado, en este caso se ha mostrado que la implementación del controlador en el DSP es relativamente sencilla y sobre todo si se utiliza uno con operaciones en punto flotante.

Se trabajo con una planta de primer orden para la cual se obtuvo su modelo. Para poder probar el controlador no se hicieron cambios en la planta, si no más bien se cambiaron los modelos de referencia programados el DSP. En este caso, se ha mostrado de manera práctica como la planta ha seguido a modelos distintos cambiando la ganancia y el polo con lo cual se cumple de manera satisfactoria el objetivo de este proyecto.

Se tuvieron problemas para encontrar la gamma adecuada; sin embargo, se llegó, por experimentación, a que una primera aproximación del valor de la gamma pudiera ser el inverso de la referencia ( $U_c$ ) al cuadrado y a partir de ahí ajustar hasta obtener el valor adecuado.

Para cambios grandes de la referencia se tiene un sobreimpulso grande, esto se pudiera resolver (se está trabando actualmente en ello) haciendo un ajuste dinámico de la gamma dependiendo del valor de la referencia.

## REFERENCIAS

- [1] Aström, Karl J.; Wittenmark, Björn; Adaptative Control; Second Edition. Addison-Wesley publishing Co.; 1995
- [2] TEKTRONIX, osciloscopio de almacenamiento digital de las series TDS1000 y TDS2000, manual de usuario.
- [3] Pérez R. Javier, Control adaptativo implementado en un micro-PLC, Impulso, Revista de electrónica, eléctrica y sistemas computacionales, volumen 1, Diciembre del 2005, pag 48
- [4] Franklin Gene F., Powell J. David; Control de sistemas dinámicos con retroalimentación; Addison-Wesley Iberoamericana; E.U.A.; 1991.
- [5] Ogata, Katsuhiko; Ingeniería de Control Moderna; Tercera edición. Prentice Hall; 1997.
- [6] Texas Instruments, TMS320C3x DSP Starter Kit User's Guide, 1996.
- [7] Texas Instruments, TMS320C3x User's Guide, Julio de 1997.
- [8] Pérez R. Javier, Bracamontes M. Marco Antonio, Felix G. Juan C. Prototipo de un Control Adaptable de Ganancia Variable aplicado a un motor de CD de un HP, CIECE 2006

**Pérez R. Javier**, nació el 2 de Febrero de 1973. Obtuvo el título de Lic. en electrónica en la Universidad Autónoma de Puebla en 1999, el grado de Maestro en Ciencias en Ing. Electrónica en el Centro Nacional de Investigación y Desarrollo Tecnológico, en Cuernavaca Morelos, en el 2000. Del 2000 a la fecha labora como profesor de tiempo completo en el Instituto Tecnológico de Sonora, en Ciudad Obregón, Sonora, México. Sus áreas de interés son: control automático, procesamiento de imágenes y electrónica de potencia.



**Pelayo S. Abelardo Roger**, nació el 5 de Febrero de 1980. Obtuvo el título de Ing. en Electrónica en el Instituto Tecnológico de Sonora en el 2006. Su área de desempeño es en control automático e instrumentación electrónica.





**Manzanarez A. Francisco**, nació el 28 de Junio de 1981. Obtuvo el título de Ing. en Electrónica en el Instituto Tecnológico de Sonora en el 2006. Su área de desempeño es en control automático e instrumentación electrónica.

# Diseño de un sistema para análisis de las vibraciones mecánicas como método de mantenimiento predictivo en máquinas rotativas

Bojórquez D. Gilberto, Rivera M. José, Chacón M. Mario, Carillo R. Mariano.

**Resumen**— En este artículo se presenta el desarrollo de un sistema para monitoreo y detección de fallas en motores de rotación, por medio del análisis de las secuencias sonoras que éstos generan, utilizando algoritmos como la FFT. Los algoritmos se implementaron en una librería de acceso dinámico (DLL), y bajo la interfaz gráfica de un software desarrollado en el lenguaje de programación de propósito general *Delphi*. En los resultados que aquí se presentan demuestran el desarrollo de un sistema económico para el análisis de vibraciones, por los sensores que utiliza y las bases necesarias para el desarrollo a futuro de otros sistemas.

**Palabras clave**—Detección de fallas en motores de rotación, DLL, FFT, vibraciones.

## I. INTRODUCCIÓN

El empleo del análisis de las vibraciones mecánicas como método de mantenimiento predictivo en máquinas rotativas es una técnica que lleva siendo empleada con éxito desde hace muchos años en la industria.

Se han desarrollado una gran variedad de aplicación para este propósito, que consisten en sensores de vibraciones (acelerómetros) los cuales son colocados en varios puntos de la o las máquinas que se deseen monitorear, estos sensores o transductores se encargan de transformar las vibraciones en señales eléctricas. Luego, estas señales eléctricas, a través de la tarjeta de adquisición de datos son ingresadas a una computadora donde se realizan diferentes tipos de procesamientos para obtener toda la información que se requiere para el análisis y monitoreo de las vibraciones de las máquinas [1].

Sin embargo los sensores que se usan en este tipo de sistemas son muy costosos y poco prácticos, éstos podrían ser sustituidos por sensores como micrófonos para detectar las ondas sonoras y con base en esto determinar los niveles de vibración.

### A. Las Vibraciones

El estudio de las vibraciones mecánicas es una rama de la mecánica, y por lo tanto de la ciencia que estudia los movimientos oscilatorios de los cuerpos, sistemas y de las fuerzas asociadas. El interés de las vibraciones mecánicas llega al mantenimiento industrial de la mano del mantenimiento preventivo, con el interés de alerta que significa un elemento

vibrante en una máquina, y la necesaria prevención de las fallas que traen las vibraciones a medio plazo [2].

El interés principal para el mantenimiento deberá ser la identificación de las amplitudes predominantes de las vibraciones detectadas en el elemento o máquina, la determinación de las causas de la vibración, y la corrección del problema que ellas representan [5].

Las consecuencias de las vibraciones mecánicas son el aumento de los esfuerzos y las tensiones, pérdidas de energía, desgaste de materiales, y las más temidas: daños por fatiga de los materiales, además de ruidos molestos en el ambiente laboral, etc [2]

Algunas de las razones más habituales por las que una máquina o elemento de la misma pueden llegar a vibrar podrían ser las siguientes:

- vibración debida a la falta de alineamiento (maquinaria rotativa)
- vibración debida a la excentricidad (maquinaria rotativa).
- vibración debida a la falla de rodamientos y cojinetes.
- vibración debida a problemas de engranajes y correas de transmisión (holguras, falta de lubricación, roces, etc). [3]

Para el análisis de este tipo de vibraciones es esencial el modelado matemático, que se define de la siguiente manera.

La ecuación general de las vibraciones es:

$$m\frac{\partial^2 y}{\partial t^2} + b\frac{\partial y}{\partial t} + ky = p(t) \quad (1)$$

Donde “y” es la magnitud que sufre variaciones periódicas temporales,  $P(t)$  la variable de reforzamiento o fenómeno incidente de la vibración;  $a$ ,  $b$ ,  $y$   $k$  son las constantes características del sistema.

Utilizando transformada de Laplace, tenemos que:

$$as^2 y(s) + bsy(s) + ky(s) = p(s) \quad (2)$$

$$(as^2 + bs + k)y(s) = p(s)$$

Manuscrito recibido el 16 de Agosto de 2006.

Rivera M. José es profesor investigador en la división de estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua. (e-mail: jrivera@itchihuahua.edu.mx)

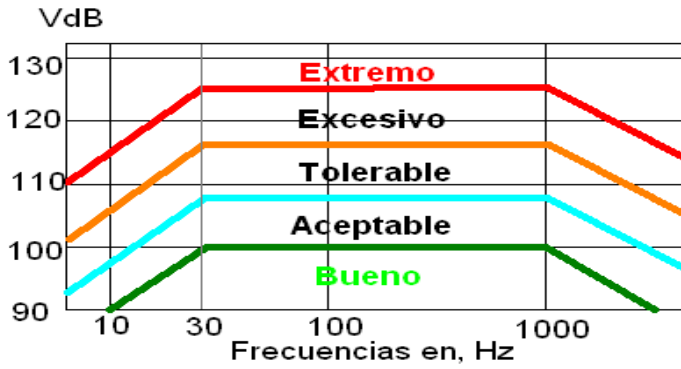


Fig. 1 Gráfica del grado de vibración en máquinas.

$$\frac{y(s)}{P(s)} = G(s) = \frac{1}{s^2 + \frac{b}{a}s + \frac{k}{a}} \quad (3)$$

Observamos que la ecuación 3 nos define la función de transferencia general de nuestro sistema de vibración, lo cual facilita el modelado y elaboración de simulaciones [4]. El estado de una máquina se determina mejor por una serie de mediciones de vibración hecho en un largo tiempo. Normas absolutas se pueden usar como guía si no hay datos históricos [2].

En la figura 1 se muestra la relación de las normas absolutas que se pueden aplicar a un gran número de máquinas rotativas con una confianza razonable.

## II. DESARROLLO

Debido a que las vibraciones mecánicas generan señales sonoras de la misma frecuencia [5], se usará un sistema de captura de audio para analizar las componentes de frecuencia que se encuentran en las ondas sonoras, y de esta forma determinar las vibraciones mecánicas.

El sistema está formado por un software, el cual se encarga de fungir como manejador de la tarjeta de sonido de una PC, para poder capturar en forma digital la señal de audio proveniente del micrófono, tal como se muestra en la figura 2.

El software implementa un algoritmo de análisis, que básicamente se encarga de comparar una señal adquirida con una señal patrón que se encuentra en una base de datos de la computadora y de esta forma poder determinar disturbios que se hayan generado en las máquinas.

El sensor que se utiliza para capturar las ondas sonoras es un micrófono piezoeléctrico estéreo, con una frecuencia de 50Hz ~ 22KHz y una sensibilidad de  $-58 \pm 3$ dB. El cual se encarga de transformar las señales de audio en señales eléctricas, las cuales posteriormente serán capturadas por la tarjeta de sonido.

La tarjeta de adquisición de datos que se usó es la Realtek AC'97 Audio, que es una tarjeta convencional de audio, que nos permite muestrear señales de hasta 22,050 Hz, suficiente para el propósito de este proyecto.

### B. Acondicionamiento de la señal

Sin embargo cuando se obtiene la señal de audio, contiene información que no es importante para el análisis, así que, se usa

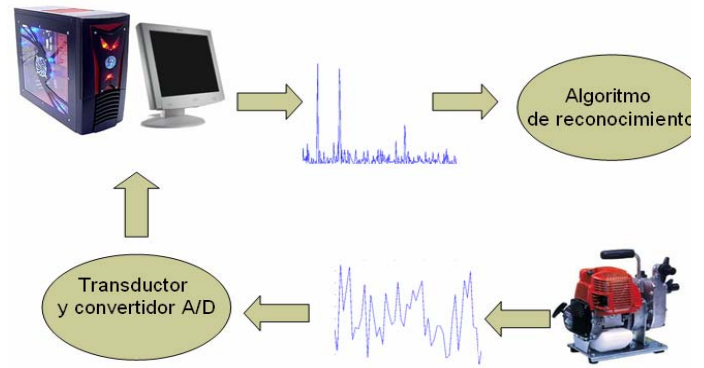


Fig. 2 Descripción del sistema de adquisición de datos

un sistema acondicionador de señal para generar una secuencia que contenga solamente aquella información que requerimos para nuestro algoritmo.

Debido a que las vibraciones que se pueden presentar en máquinas rotativas, no superan los 4kHz [6], se usa un filtro digital para eliminar frecuencias superiores a este valor. Para este propósito se diseñó un filtro Butterworth IIR, el diseño del filtro se llevó a cabo mediante la función “butter” de Matlab. El filtro es de orden 10, este valor se obtuvo de varias pruebas realizadas, cuando el orden disminuía, se presentaba una atenuación de la señal en la salida del filtro, y al aumentar el orden del filtro, su salida no mostraba cambios significativos, pero si aumentaba el tiempo de computo del algoritmo.

$$H(z) = \frac{B(z)}{A(z)} = \frac{0.05 + 0.56z^{-1} + 2.5z^{-2} + 6.7z^{-3}}{1 + 4.4z^{-1} + 9.9z^{-2} + 13.8z^{-3} \dots} \dots \dots \dots \frac{11.8z^{-4} + 14.2z^{-5} + 11.8z^{-6} + 6.7z^{-7}}{13.2z^{-4} + 9.0z^{-5} + 4.4z^{-6} + 1.51z^{-7} \dots} \dots \dots \dots \frac{2.5z^{-8} + 0.5z^{-9} + 0.05z^{-10}}{0.35z^{-8} + 0.04z^{-9} + 0.0031z^{-10}} \quad (4)$$

En la ecuación 4, se muestra la función matemática que define al filtro pasa bajas, usado para eliminar aquellas componentes de frecuencia superiores a los 4kHz, y en la figura 3 se muestra su comportamiento en la frecuencia.

El filtro podría haber sido implementado directamente en el programa de aplicación, mediante el lenguaje de programación de alto nivel como lo es Delphi, donde fue desarrollada la interfase grafica, sin embargo se optó por ser implementado mediante una dll escrita en Matlab, esto para propósito didáctico, y así ejemplificar el uso de funciones de Matlab en lenguajes convencionales, y lograr con ello disminuir el tiempo de desarrollo del software.

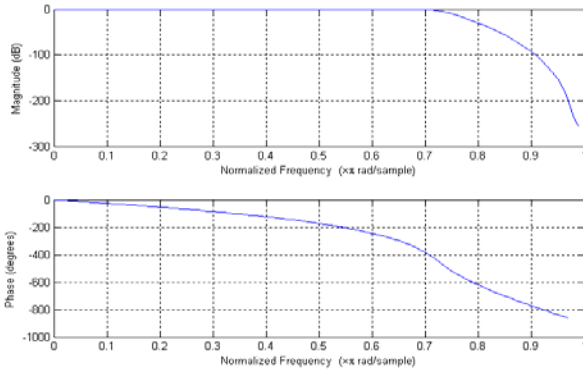


Fig. 3 Análisis en la frecuencia del filtro para acondicionador de señal

La función “datos”, que a continuación se lista, adquiere la señal que ha sido muestreada con una frecuencia de 11025 Hz y con un tiempo de adquisición de 1s, posteriormente esa secuencia es procesada por el filtro pasa bajas de 4kHz.

#### Función de adquisición y filtrado de datos

```
function y = datos
    AI = analoginput('winsound');
    chan = addchannel(AI,1);
    duration = 1; %x second acquisition
    set(AI,'SampleRate',11025)
    ActualRate = get(AI,'SampleRate');
    set(AI,'SamplesPerTrigger',duration*ActualRate)
    set(AI,'TriggerType','Manual')
    blocksize = get(AI,'SamplesPerTrigger');
    Fs = ActualRate;
    start(AI)
    trigger(AI)
    data = getdata(AI);
    delete(AI)
    clear AI
    fc = 4000
    fs = 11025
    Wn = (fc/(fs/2));
    [B,A]=butter(10,Wn)
    y = filter(B,A,data)
```

Como vemos en la figura 3 tenemos una frecuencia angular normalizada de corte de  $W_n=0.72\pi$  lo cual corresponde a  $f_c=4000\text{Hz}$ , esto por la relación de la ecuación 5.

$$F_c = \frac{W_n f_s}{2} \quad (5)$$

#### B. Diseño del algoritmo del sistema

Cuando la señal es obtenida después de ser filtrada, hay una etapa de procesamiento, la cual se encarga de descomponer la secuencia de datos  $x_a(n)$  en una secuencia de frecuencias  $x_a(k)$ , una vez que la señal puede ser analizada en el dominio de la frecuencia, se toman 2 muestras en diferentes tiempos y se determina que componentes de frecuencia no están contenidas en ambas muestras, de esa forma se eliminan los ruidos esporádicos, cuando existen ruidos constantes que provienen de otros motores en el mismo rango de frecuencias de las señales que nos interesan, se descartan por su bajo nivel en dB, ya que señales de potencia menor o igual a 100dB no son críticas, esto por la relación de la figura 1. En la figura 4, se muestra el

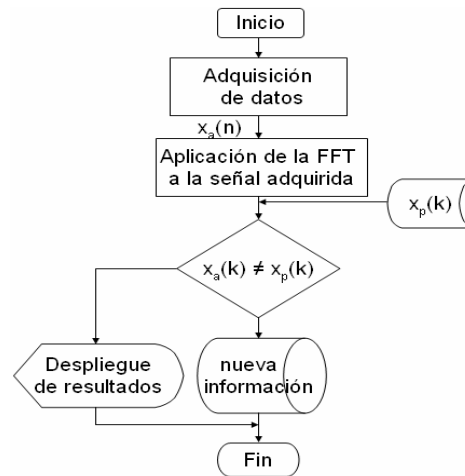


Fig. 4 Diagrama de Flujo del algoritmo del sistema.

diagrama de flujo del algoritmo.  $x_p(k)$  representa una señal patrón almacenada en una base de datos, la cual está en función de índices  $k$ , ya que se encuentra en el dominio de la frecuencia, esta secuencia es comparada con la señal  $x_a(k)$ , que representa la información adquirida en tiempo real.

Si existen componentes de frecuencia en  $x_a(k)$  que no se encuentran presentes en  $x_p(k)$ , podemos determinar que el sistema que está generando a  $x_a(n)$  está presentando anomalías que podrían ser fallas mecánicas, en este caso se generaría información útil para poder predecir en análisis futuros de que tipo de falla se trata y como podría ser solucionada, de esta forma los datos procesados en este sistema pueden ser aplicados para predecir el mantenimiento de las máquinas. La secuencia  $x_a(k)$ , es obtenida después de la implementación del algoritmo de la FFT, este algoritmo fue implementado en una función para Matlab. El listado de la función FFT se muestra a continuación. Las funciones DATOS y FFT se encapsularon en una librería de acceso dinámico (DLL), para ser exportados al lenguaje de programación Delphi, ya que en este lenguaje se desarrolló la interfaz gráfica del usuario.

a

#### Función en Matlab para la implementación del algoritmo de la FFT

```
function y=FFT(data)
    xfft = abs(fft(data));
    index = find(xfft == 0);
    xfft(index) = 1e-17;
    mag = 20*log10(xfft);
    blocksize = 4000*2
    mag = mag(1:floor(blocksize/2));
    f = (0:length(mag)-1);
    f = f(:);
    y = mag
```

En Delphi se desarrolló el algoritmo del sistema auxiliándonos de las funciones DATOS y FFT, el cual consiste en comparar una secuencia de datos de un patrón  $x_p(z)$  que define el comportamiento normal de la máquina de rotación que se está analizando con la secuencia que se está adquiriendo  $x_a(z)$ .

En la figura 5 se muestra la ventana de *comtool* donde se generó la DLL.

La DLL se constituye de una Clase (DSP), que contiene dos objetos (DATOS y FFT), esto se muestra en la Figura 6.

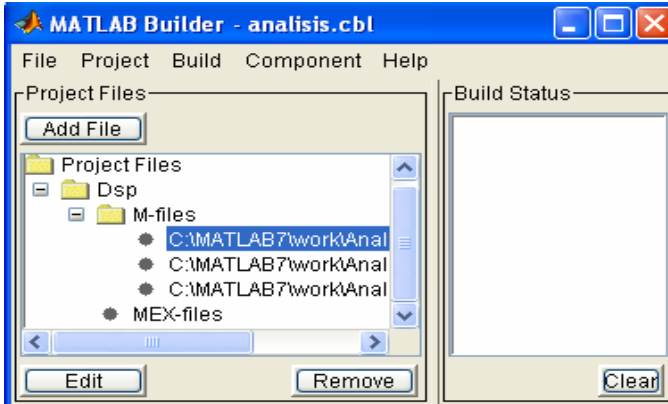


Fig. 5 Ventana de comtool de Matlab para generación de DLL a partir de funciones.

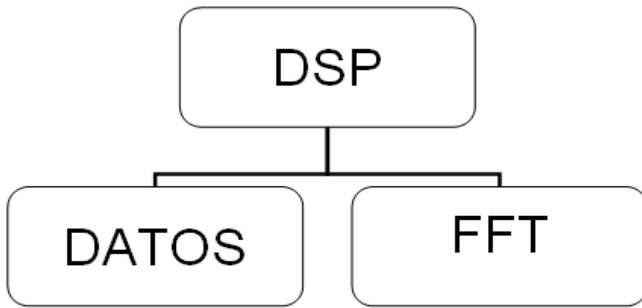


Fig. 6 Composición de la DLL

C. Diseño de la interfaz gráfica.

La secuencia  $x_p(z)$  se obtiene a partir de una muestra que puede ser obtenida desde la ventana de la Figura 7, En esta ventana podemos configurar el tiempo de adquisición de la señal, por defecto está configurada para 1s. Sin embargo, si aumentamos el tiempo de adquisición, la señal se convierte en el promedio del número de secuencias proporcional al tiempo en segundos de adquisición, con ello también podemos disminuir errores de ruido.

También nos permite cargar la secuencia adquirida  $x_p(z)$  como referencia para la comparación con la señal a muestrear  $x_a(z)$ . Dándonos la posibilidad de almacenar esta secuencia para comparaciones futuras y así determinar en nivel de cambio de la señal de la máquina con respecto al tiempo y poder predecir posibles fallas antes de que éstas se presenten. En la Figura 8 se muestra la IDE del software, donde podemos ver la comparación de las señales  $x_p(z)$  y  $x_a(z)$ .

III. PRUEBAS Y RESULTADOS

Se realizaron dos pruebas para análisis y obtención de resultados. Los cuales se muestran a continuación.

La primera prueba consistió en probar el comportamiento del sistema para diferentes componentes de frecuencia. Al sistema se

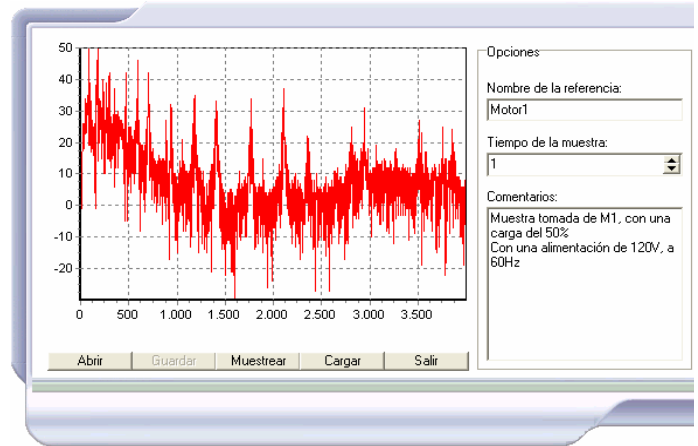


Fig. 7 Ventana de adquisición de la  $x_p(z)$ .

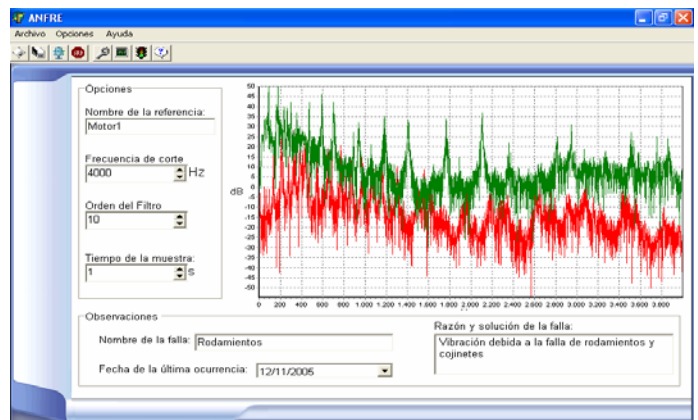


Fig. 8 IDE del Software.

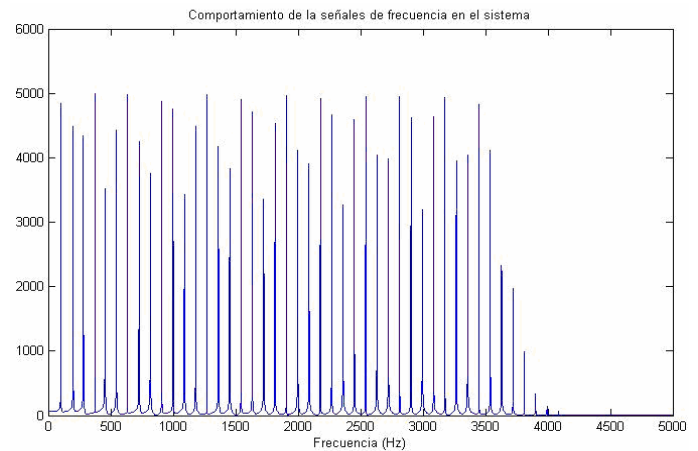


Fig. 9 Comportamiento del sistema ante una señal de audio de componentes de frecuencia entre 100Hz y 5000Hz.

le alimentó con una señal de audio, con componentes de frecuencia desde 100Hz hasta 5000Hz, con intervalos de 100Hz. Como podemos ver en la figura 9, el sistema presenta una atenuación de componentes de frecuencias para señales superiores a los 4 kHz, como lo establece el filtro pasa-bajas que describe la ecuación 4.

Una segunda prueba realizada al sistema fue para el algoritmo que se implementó en este software, se hizo la captura

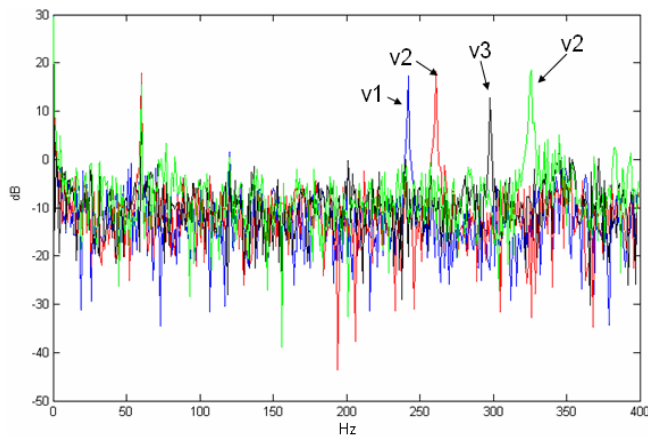


Fig. 10 Empalme de las 4 muestras.

de cuatro señales de una misma fuente, en este caso un motor de AC pero con diferentes velocidades de rotación, y nos dio como resultado cuatro secuencia de datos donde su variación más significativa estaba en el bloque que comprende los 0 Hz hasta los 400 Hz.

En la figura 10, se muestran las 4 secuencias empalmadas, las velocidades se denotan por v1, v2, v3, v4, podemos ver que para cada velocidad del motor existe una componente de frecuencia diferente, esto en proporción a la frecuencia de vibración mecánica del motor.

## V. CONCLUSIONES

En base a los resultados obtenidos, podemos asegurar que el sistema de adquisición de datos mediante percepción de ondas sonoras resulta un muy buen método, y el algoritmo utilizado nos presenta una buena alternativa para determinación de frecuencias y presenta una muy buena alternativa para este tipo de análisis y un bajo costo en cuanto a los sensores, los algoritmos utilizados en este sistema, pueden ser el punto de partida para el desarrollo de un segundo software escrito para sistemas operativos tales como Palm SO y Windows Mobile [6], para de esta forma crear un sistema portátil haciendo mas versátil y práctico el proceso de prevención de fallas de motores de rotación.

Cabe señalar que este proyecto que se presentó en este artículo es meramente de carácter didáctico, ya que muestra una de las muchas aplicaciones que podrían desarrollarse con una computadora, y la teoría de análisis de señales como filtros digitales y la transformada discreta de Fourier, a su vez este sistema podría ser usado para determinar posibles fallas en motores como un análisis previo, sin embargo no definitivo, ya que en este sistema aun faltarían varias etapas de desarrollo como su calibración, caracterización y validación.

## RECONOCIMIENTOS

Agradezco al CONACYT por el apoyo recibido a través de la beca otorgada para mis estudios.

## REFERENCIAS

- [1] Sistema de Evaluación y Diagnóstico de Máquinas Rotativas mediante Análisis de Vibraciones *National Instruments* <http://www.ni.com>.

- [2] Diagnóstico de Vibraciones en Máquinas Rotativas. <http://www.dliengineering.com/section.asp?nID=22>
- [3] Diferentes Tipos de Vibraciones Mecánicas <http://www.monografias.com>
- [4] Modelos matemáticos de vibraciones <http://www.oscarbarajas.com/vibration.htm>
- [5] Glenn D. White, Introducción a la vibración en máquinas. Predict-DLI 1997
- [6] JasonTranter, Information and power in your hands: data collectors meet mobile computers -, Application note Predict-DLI, 1998
- [7] Thurston Brooks & Brett Gidge Next generation vibration sensors show their dexterity - (Wilcoxon Research), Machine Plant and Systems monitor, Mayo 1998.



Gilberto Bojórquez Delgado recibió el título de ingeniero Electrónico en Sistemas Digitales en el año 2004 por el Instituto Tecnológico de los Mochis (ITLM), Actualmente cursa la Maestría en Ciencias en Ingeniería Electrónica en el ITCh. Es miembro activo de la IEEE y de las sociedades de Instrumentation and Measurement, Communications, Power Engineering, Signal Processing y Computational Intelligence Society, actual integrante de la rama estudiantil IEEE chihuahua. Áreas de interés: Instrumentación y control, comunicaciones, sistemas digitales, procesamiento digital de señales, redes neuronales, lógica difusa y sistemas distribuidos.



José Rivera Mejía en 1987 obtuvo el título de Ingeniero en Electrónica con especialidad en instrumentación en el Instituto Tecnológico de Cd. Guzmán Jal. Obtuvo el grado Maestro en Ciencias en 1983 en Ingeniería Electrónica en el Instituto Tecnológico de Chihuahua. De 1985 a 1995 laboró en la empresa Cable Productos de Chihuahua y desde 1995 a la fecha es profesor investigador en la división de estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua. Ha publicado internacionalmente varios artículos científicos y técnicos. Sus áreas de investigación son: Instrumentación y control, Metrología, Estudios de confiabilidad, procesamiento digital de señales, y desarrollo de aplicaciones con lógica difusa y redes neuronales.



Mario Ignacio Chacón Murguía, Recibió el título de Ingeniero en Electrónica en 1982 del ITCh en 1985 recibió el grado de Maestro en ciencias en Ingeniería Electrónica del ITCh. En 1998 Obtuvo el grado de Ph. D. en ingeniería Eléctrica en la Universidad Estatal de Nuevo México U.S.A. Es profesor investigador en la división de estudios de Posgrado e Investigación del ITCh. El Dr. Chacón ha publicado internacionalmente varios artículos científicos y técnicos. Es miembro *Senior* de la IEEE y miembro de la sociedad de reconocimiento de patrones. Sus áreas de investigación son: Visión por computadora, procesamiento digital de señales, reconocimiento de patrones, sistemas digitales, y aplicaciones con lógica difusa y redes neuronales.



Mariano Carrillo Romero recibió el grado de Ingeniero Electromecánico en la especialidad de instrumentación por el Instituto Tecnológico de Delicias en el año 2003. Actualmente cursa el cuarto semestre de Maestría en Ciencias en Ingeniería Electrónica en el Instituto Tecnológico de Chihuahua, y su área de investigación es Instrumentación y Control. Es miembro activo de la IEEE y de las sociedades de Instrumentation and Measurement, Communications, Power Engineering, Signal Processing y Computational Intelligence Society, actual integrante de la rama estudiantil chihuahua. Sus áreas de interés son sensores inteligentes y redes neuronales.

# Aplicación y adaptación del método end to end para mejorar el uso de los recursos y el desempeño de redes de cómputo

Marchena M. Misael, Domitsu K. Manuel y Solís G. Ricardo T.

**Resumen.**—¿Qué sucede cuando se presentan los problemas de desempeño en la red? máxime cuando ya hemos agotado todos los recursos disponibles para aislar el fallo y hemos fracasado, sin duda alguna es una situación que se nos ha presentado, sobre todo en las redes complejas de hoy en día que no se limitan a la infraestructura local o propia. A través de esta investigación se analizó dicha problemática adaptando el método “end to end” para ser aplicado en el equipo del usuario final de red, método propuesto por algunas de las compañías de monitoreo y análisis de configuración, fallos y desempeño en redes actuales. La principal aportación de la investigación radica en la adaptación del método end to end, con el fin de utilizarlo para mejorar el uso de los recursos de red y dar así solución a problemas comunes que afectan el desempeño de la red.

**Palabras clave.**—método end to end, redes de computo

## I. INTRODUCCIÓN

Hoy en día las redes de cómputo experimentan un crecimiento sin precedentes en el ámbito mundial, la aparición del Internet y las necesidades de desarrollo de sistemas de información, bases de datos más eficientes y el vertiginoso crecimiento del poder de cómputo en las PC, han llevado a las redes de área local, al límite de sus capacidades en cuanto a su rendimiento.

Esto demanda el trabajo constante por parte del administrador de red y su equipo de trabajo, ya que si la red no proporciona los servicios de manera oportuna y eficiente no justifica su razón de ser.

En este contexto el presente proyecto de investigación se plantea como objetivo principal, aplicar el método end to end adaptado para la medición de parámetros de desempeño de red en el equipo del usuario final, con el fin de detectar y resolver diversos tipos de fallos que limitan la eficiencia en los equipos y de esta forma contribuir en la mejora del desempeño de la red, en

su función como herramienta de soporte a las principales actividades de la comunidad universitaria, así mismo suplir, a través de este proceso de una útil, novedosa y eficaz herramienta para la administración de fallos, configuración y desempeño de redes.

## II. DESARROLLO

Hoy en día las redes de cómputo ayudan a los integrantes de una organización a lograr un flujo de información más oportuno y menos costoso, aunque se conoce que el óptimo funcionamiento de las redes depende directamente de una buena administración de los recursos disponibles y particularmente del monitoreo de su desempeño, de no ser así se incurre en periodos prolongados de ineficiencia en los cuales el trabajo de las organizaciones se ve claramente afectado, ésta es una tendencia que se agudiza a medida que las redes de cómputo se han tomado más extensas y por lo tanto más complejas (Robles, 2000) [1].

El proceso de administrar redes de cómputo es conceptualizado de manera diferente para diversas organizaciones, ya que para algunos implica, a un individuo solitario que supervisa la actividad de la red con un analizador de protocolos, para otros puede ser una base de datos distribuida, o un auto sondeo de dispositivos de conectividad y estaciones de trabajo, que generan gráficas estadísticas en tiempo real de los cambios de la topología y el tráfico de la misma.

En general se puede inferir que a todo ese conjunto de herramientas, acciones, aplicaciones, y dispositivos, operados por individuos que proporcionan este servicio se le denomina proceso de administración de redes, el cual podemos clasificar en diversos modelos, entre los más importantes tenemos el modelo TMN (Telecommunications Management Network) de la ITU y el modelo OSI-NM (Network Management). Ambos modelos dividen la administración de una red en áreas funcionales definiendo de esta forma una estructura organizacional con tareas bien definidas, coincidiendo ambos modelos en una clasificación de cinco áreas que son:

Administración de la configuración.

Administración de fallos.

Administración del desempeño.

Administración de la seguridad.

Administración de la contabilidad.

Aunque de todas estas áreas depende directamente el buen funcionamiento de las redes de cómputo actuales, en esta investigación abordamos la administración del desempeño, configuración y fallos, con el fin de indagar los principales

Manuscrito recibido el 4 de Julio de 2006 Este trabajo fue respaldado por el departamento de Ing. Eléctrica y Electrónica del Instituto Tecnológico de Sonora

Solís T. Ricardo hasta la fecha se ha de desempeñado como Coordinador de Carrera del Instituto Tecnológico de Sonora en el Departamento de Ingeniería Eléctrica y Electrónica Instituto Tecnológico de Sonora; Ave. Antonio Caso S/N Col. Villa ITSON; Ciudad Obregón, Sonora, México; C.P. 85138; C.P. 85130; Tel: (644) 4109000, ext. 121; Fax: (644) 4109001.(e-mail [rsolis@itson.mx](mailto:rsolis@itson.mx))

Domitsu k. Manuel hasta la fecha se ha de desempeñado como Profesor de Tiempo Completo del Instituto Tecnológico de Sonora en el Departamento de Ingeniería Eléctrica y Electrónica Instituto Tecnológico de Sonora; Ave. Antonio Caso S/N Col. Villa ITSON; Ciudad Obregón, Sonora, México; C.P. 85138; C.P. 85130; Tel: (644) 4109000, ext. 101; Fax: (644) 4109001.(e-mail [jperez@itson.mx](mailto:jperez@itson.mx))

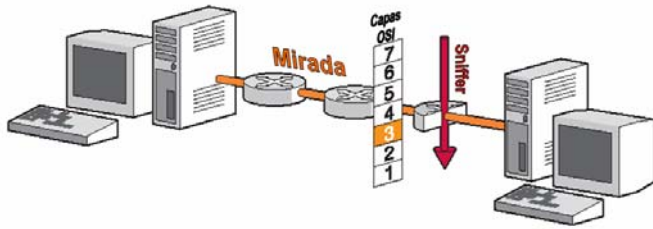


Fig. 1 Comparación del método end to end y métodos tradicionales de monitoreo

parámetros que influyen en el desempeño de las redes actuales, para poder medirlas y ponerlas a disposición del administrador de redes, el cual las analizará y tomará las acciones necesarias para que el funcionamiento de la red se pueda mantener a un nivel aceptable (Ford y Kim, 1998) [2].

Algunos ejemplos de estas variables de desempeño que se plantean con valores mínimos en los requerimientos establecidos en los manuales de operación del sistema de telefonía sobre IP de la red ITSON (Avaya communications, 2001) [3] son:

- El retardo, delay o latencia que tiene que ver con el tiempo requerido para enviar un paquete entre dos puntos de la red.
- La variación del retardo o jitter.
- El throughput o la tasa de transmisión real que depende del ancho de banda y el porcentaje de utilización del mismo entre otros.

Por este motivo los parámetros seleccionados para medir el desempeño de la red en esta investigación son:

- Throughput.
- Latencia.
- Paquetes perdidos.
- MTU máximo.

Estas variables describen de manera general el desempeño de redes de convergencia (voz, datos y video) como lo es la red ITSON.

#### Métodos de monitoreo de redes

En los modelos de monitoreo tradicionales más utilizados actualmente (esto incluye Sniffers, SNMP, entre otros) el proceso de detección de fallas está basado sólo en herramientas que incluyen para su análisis una muestra de tráfico en un punto de la red conocido con el término anglosajón pinpointing, el cual incluye el barrido de las 7 capas del modelo OSI de manera vertical desde la capa de aplicación hasta la capa física como se muestra en la Figura 1 en color rojo, esto implica una gran cantidad de datos si este proceso se realiza en tiempo real, es por eso que regularmente estos métodos realizan capturas discretas, para generar a partir de esto el análisis requerido, sobre todo cuando necesitamos conocer el estado que guarda un enlace de comunicación de alta velocidad (Apparent Networks, 2003)[5].

Una tendencia en las técnicas de monitoreo modernas es el utilizar la método end to end para el análisis del desempeño de todo tipo de redes, el cual utilizamos en el desarrollo de a investigación adaptándolo a través del uso de software de

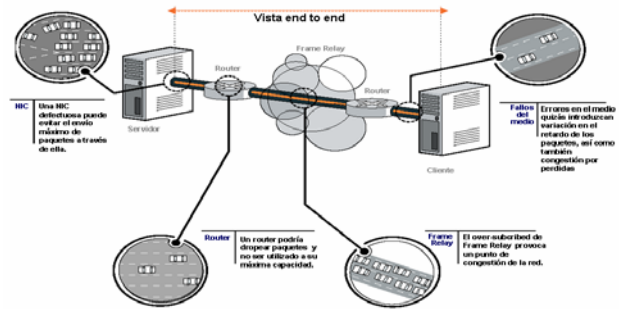


Fig. 2 Elementos que afectan la vista horizontal de la ruta end to end.

monitoreo y configuración de parámetros de red como:

NetDoppler de la compañía WildPackets, IPNetTuner de la compañía Sustentable Softworks, 3Com 3C90X NIC Diagnostic Program Test de la compañía 3Com, y TCP Optimizer de la compañía Speed Guide Inc.

El método end to end utiliza una visión holística del proceso de monitoreo, ya que realiza las mediciones con la red en operación, observando la red de la misma forma en que la aplicación mira la infraestructura de red instalada, esto es, de forma horizontal registrando el flujo de datos de todas las operaciones que suceden entre el programa del nodo que envía los datos por medio de una interfaz de programación (API) y el otro programa del nodo que recibe los datos por medio de una API como se puede observar en la Figura 1 en color naranja.

Algo que se puede destacar es que el método permite encontrar y diagnosticar problemas de desempeño de manera rápida y remota, sin necesidad de dispositivos o agentes remotos, sin impactar significativamente el desempeño de la red, aunque ésta no sea una red propia (como lo es el caso de las redes públicas como Internet). Esto se debe a que los algoritmos de análisis del método están basados en mediciones de capa 3 como se muestra en la Figura 1 en color naranja.

Medir en la capa IP, permite al método el acceso completo a la ruta de red end to end. Esta vista end to end está definida por la máquina del usuario final en un extremo de la red, la cual es conectada de manera continua por la ruta de capa 3 que incluye muchos dispositivos físicos y el mismo medio. Por lo cual podemos decir que es la misma red que la aplicación utiliza en su proceso de transmisión de paquetes, la cual comienza por la API del sistema operativo conectado a la red, continua hacia abajo a través del controlador de la NIC; viaja por el medio físico, pasando a través de los hubs, conmutadores, y compuertas (cada uno con sus NIC, controladores conectores y cables); para finalmente llegar al host destino, donde camino arriba, pasa a través de la NIC, el controlador y sistema operativo de la aplicación destino.

Cada elemento de esta ruta puede ser una limitante potencial que influya en el pobre desempeño, por ejemplo, un controlador de NIC obsoleto o con bugs puede causar la pérdida de paquetes, un cable de red hecho con mala calidad puede provocar que sea más vulnerable al ruido electromagnético causando retardos y paquetes perdidos, como se muestra en la Figura 2.



Algunos de los efectos negativos más comunes encontrados en la red con esta metodología de monitoreo son:

- Desajuste dúplex.
- Conflictos MTU.
- Mala configuración de pila de protocolos TCP/IP.
- Bajo desempeño en NICs.

### Metodología de la investigación

La metodología utilizada en este proyecto se basa en la aplicación y adaptación del método de monitoreo y medición de redes end to end, resaltándose las siguientes fases del proceso:

Proponer al Departamento de Tecnologías y Servicios de Informática (DTSI) del ITSON Aplicar la adaptación del método end to end para detectar diversos tipos de fallas que limitan la eficiencia de los equipos del usuario final en el Centro de Informática y Servicios de Cómputo (CISCO) como una prueba piloto y realizar el análisis preliminar de la red.

Obtener toda la información disponible de características y configuración de parámetros de red definidos en el equipo del usuario final, así como las características generales y la configuración en la que están operando los dispositivos de interconexión de red que estén relacionados con los nodos de usuario final analizados.

Elaborar las pruebas bajo el método a adaptado para determinar el estado actual de desempeño entre pares nodos de red, específicamente en la red CISCO, así como comprobar la confiabilidad del método adaptado en la detección de problemas de red.

Determinar y analizar en base a los datos obtenidos los posibles tipos de fallos que comúnmente provocan que los recursos del equipo del usuario final de la red no sean utilizados al 100% de la capacidad instalada.

Implementar alternativas para corrección de fallos que sean aplicables a las problemáticas diagnosticadas en los pares de nodos analizados.

Elaborar de nuevo las pruebas bajo el adaptación del método propuesto, para determinar el estado final de desempeño entre los mismos pares nodos de la red procurando registrar condiciones similares de tráfico en la red.

Evaluar y sustentar el método end to end adaptado en base a los resultados obtenidos, para determinar si esta herramienta es efectiva para la administración de configuración, fallos y desempeño, que permita detectar y solucionar problemas que impacten en la mejora del desempeño de los equipos del usuario final en el CISCO y por consecuencia sea factible aplicarlo en otras áreas de la red ITSON.

Determinar en base al desarrollo de la metodología de investigación y los resultados obtenidos en la aplicación de la adaptación del método end to end a los equipos del usuario final del CISCO, cuales fueron las principales mejoras en el desempeño de los equipos y el impacto que se obtendría al aplicar la metodología en otras áreas de la red ITSON.

Para las diferentes etapas de adaptación la metodología end to end, se utilizó el software NetDoppler de WildPackets, IPNetTuner de Sustentable Softworks, y el TCP Optimizer de Speed Guide, los cuales realizan mediciones de las parámetros de red seleccionados utilizando para ello la capa 3 del modelo de referencia OSI.

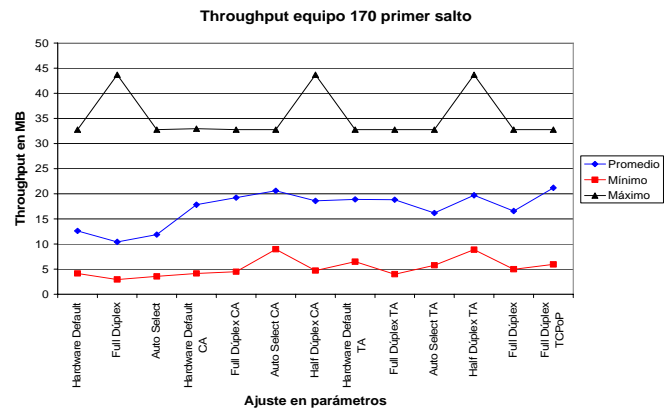


Fig. 3 Gráfica de incremento en el throughput equipo 170 primer salto.

### Resultados observados en la aplicación del método end to end adaptado.

El método se aplicó en una prueba piloto en equipos del Centro de Informática y Servicios de Cómputo del ITSON los cuales presentan las siguientes características:

- Equipo de prueba 1
- Marca Lanix.
- Sistema operativo Windows 98 SE.
- Procesador Intel Pentium III de 450 MHz.
- Memoria RAM de 256 MB.
- NIC 3Com EtherLink 10/100 modelo 3C905C-TX montada en bus PCI.

#### Equipo de prueba 2

- Marca Lanix.
- Sistema operativo Windows XP Profesional.
- Procesador Intel Pentium III de 450 MHz.
- Memoria RAM de 512 MB
- NIC 3Com EtherLink 10/100 modelo 3C905C-TX montada en bus PCI.

Para el caso del primer equipo que opera en la plataforma Windows 98 SE, el primer salto que dan los paquetes del equipo de prueba a la compuerta de salida, el parámetro throughput registró un incremento de 68% en relación a su estado inicial, una vez aplicada la el método end to end adaptado, como se muestra en la gráfica de la figura 3.

La configuración de parámetros utilizada para cada medición recavada en las gráficas son las siguientes:

- Hardware Default: la configuración corresponde a la establecida por omisión y representa el estado inicial de los parámetros a medir con respecto a los parámetros throughput y latencia.
- Full Dúplex: en configuración full dúplex se realizan las mediciones modificando los parámetros de configuración dúplex a modo full, vía las opciones permitidas por el sistema operativo de red utilizado.
- Half Dúplex: en configuración half dúplex se realizan las mediciones modificando los parámetros de configuración dúplex a modo half, vía las opciones permitidas por el sistema operativo de red utilizado.

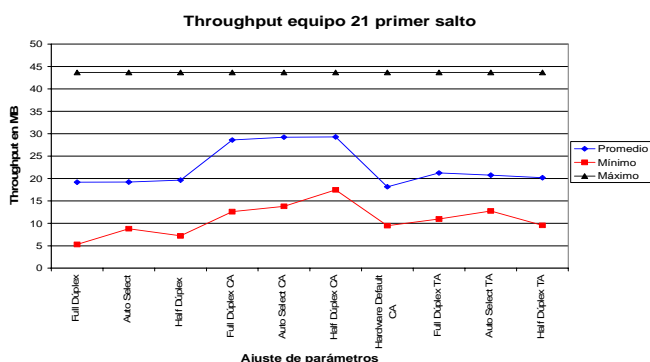


Fig. 4 Gráfica de incremento en el throughput equipo 21 primer salto.

- Auto Select: en la configuración Auto Select se efectúan las mediciones modificando los parámetros de configuración dúplex en modo auto seleccionado, vía las opciones permitidas por el sistema operativo de red utilizado.
- Full Dúplex TCPoP: es la medición que se efectúa una vez modificados los parámetros TCP/IP a través del software TCPOptimizer.

Todas las mediciones de parámetros se repiten para actualización de controladores de NIC, configuración de parámetros de NIC a nivel físico y configuración de parámetros TCP, como se observa en las gráficas. Para el caso del parámetro latencia se observó que los cambios realizados a través de la aplicación de la metodología no afectaron de forma significativa a los valores que comúnmente se tienen el parámetro latencia. En el segundo salto del primer equipo, de manera similar al primer salto se observó un incremento en el parámetro throughput del 40% en relación a su valor de inicio y el parámetro latencia no mostró variación significativa.

Para el caso del segundo equipo en plataforma Windows XP, en el primer salto a la compuerta de salida el throughput mostró un incremento menor que el equipo en plataforma Windows 98 SE, el cual fue de 5% en relación a su estado inicial una vez aplicada la metodología end to end, como se muestra en la gráfica de la figura 4.

Para el caso de la latencia se observó que los cambios realizados a través de la aplicación de la metodología no afectaron de forma significativa a los valores que comúnmente se tienen de latencia.

En el segundo salto al servidor DNS del segundo equipo de manera similar al primer salto se observó un pequeño incremento en el throughput del 10% en relación a su valor de inicio y la latencia no mostró variación significativa.

### III. CONCLUSIONES

Con base en los resultados obtenidos en la presente investigación se llegó a las siguientes conclusiones:

De manera específica y concreta la principal aportación de este trabajo de investigación radica en la adaptación del método end to end, con el fin de utilizarlo para mejorar el uso de los recursos de red y dar así solución a problemas de desempeño de la red.

En los que respecta a la hipótesis de investigación que se planteó, se encontró que la optimización de parámetros de configuración de los recursos de red en el equipo del usuario final, a través de la detección y solución de fallos auxiliado por el método end to end adaptado, impacta de manera significativa en el rendimiento efectivo del equipo del usuario final de la red.

Generándose para los casos analizados en la investigación un mayor impacto a través de:

- En primer término, en la actualización de controladores de red, ya que en todos los casos analizados los equipos mostraron una mejora significativa al realizar los ajustes correspondientes.
- En segundo término, la configuración de parámetros TCP/IP. Es importante mencionar que existe la posibilidad de lograr un mayor impacto a través de la manipulación individual de cada uno de los parámetros de la pila TCP/IP, ya que en esta investigación sólo se utilizó la propuesta de mejora sugerida por el software TCP Optimizer como configuración óptima.
- En tercer término, la configuración dúplex a nivel de hardware y software. Para el caso específico de este ajuste podemos comentar que no se registraron desajustes dúplex en la configuración de la infraestructura de red del Centro de Informática y Servicios de Cómputo, las variaciones registradas fueron provocadas por el mismo ajuste en la configuración de pruebas lo que fue suficiente para mostrar las repercusiones negativas de este tipo de fallas, y no se descarta encontrarla en otras áreas de la red.

El parámetro que registró la variación más significativa fue el throughput, los demás se mantuvieron razonablemente constantes por lo cual, se puede considerar como el parámetro a monitorear en futuras investigaciones en la red del ITSON.

Es importante destacar a manera de resumen los principales beneficios de la aplicación del método end to end en los equipos del usuario final del Centro de Informática y Servicios de Cómputo, los cuales se listan a continuación:

1. Mejora significativa en los sistemas propietarios, como lo es el caso de las aplicaciones manejadas a través de la intranet ITSON, al contar en algunos casos con un throughput final de más del doble del valor inicial.
2. Posibilidad de conocer el rendimiento real de los equipos del usuario final y los equipos de conectividad, que puede influir en la toma de decisiones como:

Mejoras en los procesos de administración de la red sobre todo en la configuración, fallos y desempeño de la misma.

### REFERENCIAS

- [1] Robles, A.. "Análisis de desempeño de una Intranet en una red corporativa", Tesis de maestría. Universidad de los Andes, Departamento de Ingeniería Eléctrica y Electrónica. Año 2000, 8 p.
- [2] Ford Merilee y Kim Lew, H. "Tecnologías de interconectividad de redes", Cisco Press. Año 1998.
- [3] Avaya communications, "ECLIPS Technical Training, Installation Student Guide", Año 2001, pp 7-8.
- [4] Tanenbaum, Andrew S. "Redes de Computadoras", 4ta edición. Prentice Hall. Año 2002.

- [5] Apparent Networks. "The Apparent Network: Concepts & Terminology. Technical White Paper Series", Año de 2005, 8 p.



**Marchena M. Misael** Coordinador de la carrera de Licenciado en Diseño gráfico del Instituto Tecnológico de Sonora. Últimos estudios: Maestría en Ingeniería en Administración de la Tecnología Eléctrica. Área de trabajo: Soporte Técnico de hardware de computadora y redes de 1995 – 1999 en el Instituto Tecnológico de Sonora, Coordinador del Centro de Informática Servicios de Cómputo (CISCO) de 1999 – 2001 en el Instituto Tecnológico de Sonora, y Maestro Auxiliar del Departamento de Ingeniería Eléctrica de 1999 – 2005 en el Instituto Tecnológico de Sonora.



**Domitsu K. Manuel** tiene el grado de Maestro en Ciencias Computacionales en el del Instituto Tecnológico y de Estudios Superiores de Monterrey y el Grado de Maestro en Administración en el Instituto tecnológico de Sonora. De 1976 a 1983 y de 1989 a la fecha se ha desempeñado como Profesor – Investigador de Tiempo Completo en el Instituto tecnológico de Sonora. Ha participado en algunos proyectos de desarrollo de software, entre ellos el software de Recepción de Imágenes del Satélite



Metereológico GOES y un Programa de Procesamiento de Imágenes. Sus principales intereses son el Desarrollo de Software y los Sistemas Distribuidos.

Solís G. Ricardo obtuvo el grado de Maestro en Ciencias en Electrónica y Telecomunicaciones con especialidad en Telecomunicaciones del CICESE en 1998 en la ciudad de Ensenada, BCN, México.

Es maestro de tiempo completo de la carrera de Ingeniero en Electrónica del Instituto Tecnológico de Sonora desde 1988, fue coordinador de la misma del 2000 al 2005 y actualmente es maestro investigador y responsable del programa. Sus líneas de interés son las telecomunicaciones y redes de computadoras.

El M.C. Solís es miembro de la IEEE, pertenece al cuerpo académico de Tecnología Eléctrica en el Instituto Tecnológico de Sonora y es Evaluador del Consejo de Acreditación de la Enseñanza de la Ingeniería (CACEI).

# Biblioteca de simulación por software multiplataforma para el microcontrolador M68HC11

Olea A. César y Romero A. Eduardo

**Resumen** — En el estudio de sistemas electrónicos es inevitable el tema de programación en lenguaje de bajo nivel para el desarrollo de sistemas basados en microprocesador y microcontrolador. Está claro que es crucial el manejo y empleo de herramientas que faciliten el proceso de enseñanza, tales como los simuladores. Pero con la desventaja de que casi siempre hay que desembolsar un costo por su uso o por la limitante de poder usarse en un sistema operativo específico. De tal forma, sería preferible crear una biblioteca de simulación multiplataforma que tome en cuenta los beneficios de un sistema unificado de clases que permitan el desarrollo de programas capaces de integrar la simulación de un microcontrolador de una manera transparente para el programador, aún cuando se le restan las desventajas que esto presenta como la sobrecarga que aplica el intérprete del lenguaje multiplataforma a la computadora anfitriona.

La presente investigación propone una solución basada en la ejecución de código manejado por una máquina virtual. En teoría con el uso de una máquina virtual los sistemas de simulación pueden ser reutilizados en diferentes plataformas computacionales sin necesidad de recompilar el código, siempre y cuando exista una implementación de la máquina virtual para el sistema operativo y arquitectura sobre la cual esta corriendo la computadora anfitriona.

**Palabras clave** — Microcontrolador, simulador, máquina virtual, biblioteca de software.

## I. INTRODUCCIÓN

En la electrónica actual, es indudable el impacto que representan los microcontroladores (MCU). Estos circuitos integrados a gran escala pueden ser herramientas útiles para la ingeniería siempre y cuando se encuentren en las manos de un ingeniero capaz de programarlos para un rendimiento óptimo de sus características. El desarrollo de aplicaciones con estos dispositivos ha dado origen a lo que se le conoce como “sistemas embedded”.

Manuscrito recibido el 1 de Diciembre de 2006. Este trabajo fue respaldado por el departamento de Ing. Eléctrica y Electrónica del Instituto Tecnológico de Sonora.

César Olea Aguilar es Ingeniero en Electrónica, con acentuación en Instrumentación y Control, egresado del Instituto Tecnológico de Sonora. Trabajó en la fábrica de software del ITSON (Novutek) y en la empresa CIMSA en Nogales Sonora. Actualmente cursa estudios de posgrado en el área de computación en el CICESE en Ensenada, B.C.N.

Eduardo Romero A. Autor hasta la fecha se ha desempeñado como Profesor de Tiempo Completo del Instituto Tecnológico de Sonora en el Departamento de Ingeniería Eléctrica y Electrónica Instituto Tecnológico de Sonora; Ave. Antonio Caso S/N Col. Villa ITSON; Ciudad Obregón, Sonora, México; C.P. 85138; C.P. 85130; Tel: (644) 4109000, ext. 1200; Fax: (644) 4109001 (e-mail [eromero@itson.mx](mailto:eromero@itson.mx)).

La familia del M68HC11xx está formada por microcontroladores avanzados de 8 bits desarrollados con tecnología HCMOS. Cuentan con capacidades periféricas sofisticadas que se encuentran construidas en el mismo chip (*on-chip*). Operan con una velocidad nominal de 2 MHz y su diseño estático logra mantener el consumo de potencia bajo [1].

Los dispositivos de la familia HC11 se utilizan ampliamente en el ambiente educacional debido a su facilidad de programación y sus excelentes capacidades periféricas como convertidores analógico – digitales, puertos de entrada – salida de uso general, interfaz de comunicación serie y temporizadores. Todo lo anterior los hace un excelente candidato para usarlo como material de enseñanza en un curso avanzado de sistemas digitales.

También estos dispositivos son regularmente empleados en aplicaciones alimentadas por baterías o en la industria automotriz debido a sus modos de conservación de energía controlados por software [2].

En la actualidad, existen múltiples simuladores de la familia de microcontroladores HC11 y esto se debe a que fue muy popular en la época de los 90's y aún se utiliza por estudiantes, amateurs y profesionales de la electrónica para proyectos principalmente de robótica o instrumentación y control [3, 4].

La forma más común de simular el HC11 es interpretando el código máquina del programa (en formato S19), para después cargarlo a la memoria del simulador y comenzar el ciclo de búsqueda y decodificación modificando los registros internos y memoria como sea necesario [5].

El incentivo principal que motiva la creación de la biblioteca de simulación multiplataforma para el HC11 es tener disponible una plataforma para la creación de distintas herramientas que faciliten el aprendizaje y el desarrollo para el microcontrolador. Actualmente existen varios simuladores (algunos mostrados en la tabla I), pero ninguno que integre características que lo hagan especialmente atractivo al ambiente educativo y amateur.

Se eligió el lenguaje de programación Java para lograr máxima portabilidad en la biblioteca de programación con poco esfuerzo en este punto específico.

Java, desarrollado principalmente por James Gosling en Sun Microsystems, utiliza el paradigma de programación orientada a objetos. Desde el inicio fue concebido con cuatro principales puntos en mente: orientación a objetos, independencia de la plataforma, con bibliotecas de apoyo y facilidades en el lenguaje para redes y diseñado para ejecutar código de fuentes externas de manera segura.

TABLA I COMPARACIÓN DE DISTINTOS SIMULADORES PARA EL HC11

Nombre	Plataforma	Costo (dlls)	Aún activo
AVSim11	MS-DOS	Abandonado	No
THRSim11	Windows	\$50	Si
Wookie	Windows	Gratis	No
6811wsim	Windows	Abandonado	No
EAF HC11 Simulator	Windows	\$39.95 Estudiante	No

Gracias a los procesadores modernos y a las nuevas técnicas de simulación como la recompilación dinámica, el impacto en la velocidad de ejecución de simuladores en código manejado puede ser bastante eficiente manteniendo la multiplataforma.

### II. ALCANCE DEL PROYECTO

La biblioteca de simulación se ha concentrado en las características básicas del HC11 como lo son el CPU y el mapa de memoria. La biblioteca no intenta replicar de manera exacta el comportamiento del HC11, aunque construye una infraestructura que puede servir para futuras ampliaciones en la programación de un simulador.

La simulación utiliza la técnica de interpretación. Esta técnica produce código lento pero es más fácil de mantener e implementar. La biblioteca de simulación es completamente independiente de cualquier interfaz gráfica, lo que quiere decir que con la biblioteca de simulación se pueden crear aplicaciones de escritorio con interfaz gráfica, aplicaciones de consola o aplicaciones Web.

Ninguno de los métodos de entrada/salida del microcontrolador fue simulado. Sin embargo, los registros que controlan a los puertos así como aquellos que reciben o externan el dato, si pueden ser simulados. Esto se debe a que en el HC11 se usa la E/S mapeada a memoria, con lo que, estos registros a final de cuentas son similares a los registros de propósito general.

La biblioteca de simulación soporta el juego de instrucciones completo del HC11, sin embargo algunas instrucciones no realizan ninguna operación. Al momento de cargar un programa a la memoria del dispositivo simulado, el cargador (*loader*) acepta cualquier instrucción válida en formato S19, sin embargo, las instrucciones pertinentes a sistemas no simulados no realizan ninguna acción en especial, tan solo dejan pasar los ciclos de reloj que se consumirían al ejecutar la instrucción

### III. DESARROLLO DE LA BIBLIOTECA DE SIMULACIÓN

Al inicio del proceso de programación de la biblioteca fueron definidas dos clases que representan las dos estructuras de almacenamiento presentes en el HC11: *Jbyte* y *Jword* para representar bytes y palabras respectivamente. En la figura 1 se muestra la relación entre las clases.

Un *Jbyte* se encuentra definido como un arreglo de ocho posiciones de tipo booleano, en donde cada casilla del arreglo puede tener valor verdadero (1) o falso (0) como se muestra en la figura 2.

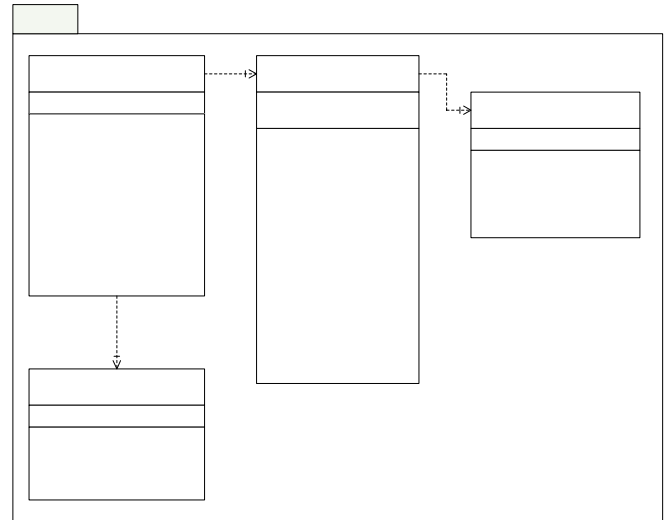


Fig. 1 Clases y la relación que existe entre ellas

0	1	2	3	4	5	6	7
False	False	False	False	False	False	False	False

Fig. 2. almacenamientojbyte a 0.

La clase *Jword* esta formada internamente por dos *Jbyte*, representando la parte más significativa y menos significativa. Cada una de las clases se encarga de proporcionar métodos para escribir y leer un dato a la memoria que representan, así como también proporcionan una interfaz para realizar otro tipo de acciones como complementos, incrementos, conversiones entre representaciones numéricas, etc.

Para construir el mapa de memoria se creó una clase llamada *JmemoryRegister* que representa a una localidad de memoria. Su estructura se muestra en la figura 3.

De la clase *JmemoryRegister* heredan dos clases: RAM y ROM. Estas clases representan un byte del tipo designado por el nombre e implementan sus propios métodos de lectura y escritura, es decir, sobrescriben los métodos heredados de su clase padre.

Al momento de construir el mapa de memoria, se agruparon segmentos iguales de memoria y son estos grupos (representados por la clase *JmemoryObject*) los que se mapean. La clase *JmemoryObject* contiene un vector que guarda cualquier tipo de objetos hijo de la clase *JmemoryObject* (que hereden de *JmemoryObject*).

La clase *JmemoryObject* ya se asemeja más a un mapa de memoria completo. Para leer o escribir a estos grupos de memoria es necesario proporcionarle la dirección y el dato a escribir si es el caso. Los objetos *JmemoryObject* son concientes de los límites en los que se encuentran, para lo cual utilizan la dirección física en la que se encuentran mapeados y el tamaño del vector de memoria que contienen.

Cuando un *JmemoryObject* lee un dato exitosamente, este regresa como resultado un *JmemoryRegister* que contiene el dato leído. Este *JmemoryRegister* puede ser promovido a su clase original para no perder la implementación de sus métodos de lectura y escritura específicos.

Con el propósito de mantener un mapa de memoria coherente, se desarrollo la clase *JmemoryMap*. Esta clase

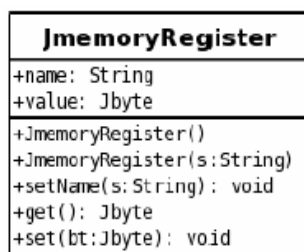


Fig. 3 Diagrama de clase para *JmemoryRegister*.

encapsula al mapa de memoria del HC11 y se encuentra formada de varios objetos *JmemoryObject* los cuales, a su vez, contienen los bytes que forman el mapa de memoria.

Gracias a que las clases que forman el mapa de memoria manejan una abstracción de lo que es un byte mapeado en memoria (*JmemoryRegister*) se pueden crear distintos dispositivos que se incluyan en el mapa de memoria, manteniendo una interfaz común de lectura y escritura de datos, es decir, sin importar el tipo de memoria o dispositivo en el mapa, todos se leen y escriben llamando al mismo método y proporcionando los mismos datos y siempre se obtienen objetos del mismo tipo como resultado, pudiendo el usuario promoverlos a su tipo de clase original.

La clase HC11 encapsula todo el comportamiento del microcontrolador, este contiene un mapa de memoria y se encarga de controlar el ciclo de búsqueda y decodificación (*fetch/decode*) que a su vez se encarga de ejecutar por medio de reflexión a los métodos de la clase *Opcodes* la cual contiene todas las instrucciones que el HC11 puede ejecutar.

#### IV. USO DE LA BIBLIOTECA DE SIMULACIÓN

Para crear un microcontrolador HC11 simulado utilizando la biblioteca dentro de nuestro programa, basta con instanciar un objeto de tipo HC11.

```
HC11 mi cro = new HC11();
```

En este momento se tiene un objeto llamado micro el cual es un HC11 capaz de cargar a su memoria programas en formato S19, desensamblarlos y ejecutarlos paso a paso, con la opción de modificar el flujo del programa en cualquier momento. Cuando se instancia un HC11 nuevo, automáticamente se agregan tres bloques de memoria a su mapa: La RAM interna en la dirección 0x00, los registros especiales en la dirección 0x1000 y un vector de memoria RAM que comienza en 0xFF00.

Es posible agregar o eliminar bloques de memoria al mapa del microcontrolador. Continuando con el ejemplo del HC11 declarado anteriormente, si se quiere eliminar el bloque de memoria presente en 0xFF00.

```
mi cro. removeJmemoryObject(2);
```

El argumento del método *removeJmemoryObject* es debido a la posición del bloque de memoria en el microcontrolador. Como se mencionó anteriormente, el primer bloque (índice 0) lo ocupa la RAM interna, el segundo los registros y el tercero lo ocupa el bloque que se desea eliminar.

Si por el contrario, se desea agregar un bloque de memoria en la dirección 0x100 de tamaño 0x100 y del tipo RAM.

```
Jword di r l n i c i o = new Jword(0x100);
```

```
Jword tam = new Jword(0x100);
RAM tipo = new RAM();
JmemoryObject mi Obj ect = new
JmemoryObject(di r l n i c i o, tam, tipo);
```

Primero se crea una palabra con la dirección de inicio, otra palabra con el tamaño del bloque y para finalizar un objeto que herede de *JmemoryRegister* (en este caso RAM) para especificar el tipo del bloque. Con esos datos se construye el nuevo objeto de memoria, sin embargo, aún hace falta mapearlo en el HC11.

```
mi cro. addJmemoryObject(mi Obj ect);
```

Con la sentencia anterior, el objeto de memoria es agregado al mapa del microcontrolador. Antes de agregarlo la clase *JmemoryMap* verifica que el objeto que se quiere agregar no viola ninguna de las reglas establecidas (no traslapes, no mayor a 0xFFFF) y si cumple, el objeto es agregado en la primera posición disponible.

Una vez que se tiene el objeto mapeado, puede ser leído y escrito de una manera que es común para todo lo mapeado en el microcontrolador.

```
mi cro. memoryMap. set(new Jword(0x100), new
Jbyte(0xAA));
mi cro. memoryMap. get(new Jword(0x100));
```

La primera sentencia escribe en la dirección 0x100 el dato 0xAA. La segunda lee de la dirección 0x100 el dato contenido que por consecuencia será 0xAA.

La biblioteca de simulación contiene también un cargador de programa para el formato S19 así como un desensamblador. Para cargar un programa a la memoria del microcontrolador primero es necesario contar con los bloques de memoria necesarios para acomodar el programa. La clase responsable de cargar el programa se denomina *S19Loader* y cuenta con tan solo dos métodos: *loadS19* y *checksum*. El primero se encarga de cargar el programa mientras que el segundo tan solo hace una verificación del programa pero no lo carga a la memoria.

Para cargar un programa S19 primero se instancia un objeto de la clase *S19Loader*, para después llamar a su método *loadS19* e intentar cargar el programa leído a la memoria del microcontrolador.

```
String ruta = "C:\\progs\\hc11\\test1.s19";
S19Loader s19 = new S19Loader(ruta, mi cro);
s19. loadS19();
```

La ruta es dependiente del sistema operativo y por lo general se obtiene del usuario. En el momento que se llama *loadS19* sobre el objeto de tipo *S19Loader* se inicia la decodificación y carga del programa en formato S19.

Una vez que se tiene el programa en la memoria del microcontrolador, se debe modificar el contador de programa (PC) para que apunte a la primera dirección del programa cargado.

Suponiendo que el programa inicia en la dirección 0x100:

```
mi cro. setPC(0x100);
```

De la misma manera se puede alterar cualquiera de los registros internos del HC11, la tabla 2 lista los registros y los métodos utilizados para leer y escribir a ellos.

TABLA II.  
NOMBRE DE LOS REGISTROS INTERNOS.

Registro	Nombre de los métodos
PC	getPC, setPC
Acumulador A	getA, setA
Acumulador B	getB, setB
Acumulador D	getD, setD
Registro índice X	getX, setX
Registro índice Y	getY, setY
Apuntador de stack	getSP, setSP

Vale la pena señalar que estos registros internos no se acceden de la misma manera que los registros de memoria ya que estos no se encuentran mapeados en memoria, sino que se encuentran dentro de la CPU del HC11 [6].

La clase HC11 provee también métodos para controlar el estado del microcontrolador, como lo son las instrucciones de “reset” y “step”. La primera pone a los registros especiales del HC11 en su estado inicial y la segunda se utiliza para ejecutar paso a paso el programa cargado en memoria.

```
mi cro. step();
```

En este momento el microcontrolador ejecuta el método *fetch* para traerse la instrucción de la dirección apuntada por PC y acto seguido ejecuta el método *decode*, encargado de ejecutar por medio de reflexión la instrucción correspondiente al código de operación traído de la memoria del microcontrolador.

Para descompilar el programa y mostrar el listado a partir de lo que se encuentra en la memoria del microcontrolador, la biblioteca utiliza una base de datos que necesita ser instalada en la computadora anfitriona. Al descompilador se le indica el dato a descompilar y la dirección en la cual se encuentra. La dirección se utiliza para mostrar los desplazamientos o la dirección de los datos que se van a ocupar como argumento de la instrucción.

Al recibir la instrucción en hexadecimal, la biblioteca ejecuta una sentencia SQL sobre la base de datos que contiene las instrucciones. La sentencia ejecutada es la siguiente:

```
SELECT * FROM opcodes WHERE OPID = + 'opcode en hex'.
```

Lo que se traduce a: “selecciona todos aquellos renglones de la base de datos en los cuales el campo OPID coincida con el opcode enviado”. No pueden existir códigos de operación repetidos por lo que no puede darse el caso en el que se obtengan más de un renglón. Como resultado de la consulta, el método regresa una cadena con la instrucción decompilada.

Todas las funciones de la biblioteca de simulación funcionan de la misma manera y están diseñadas con la facilidad de uso en mente. De cualquier manera, es necesario consultar la documentación de las clases e inclusive ver el código fuente para entender mejor el funcionamiento de cada una de las clases y conocer las posibilidades de uso.

## V. CONCLUSIONES

La biblioteca de simulación producto de esta investigación no es un producto terminado todavía y aún le hacen falta piezas clave para elevar su utilidad en cualquier ambiente en el que se pretenda utilizar. Algunas de las piezas clave faltantes son:

Creación de los registros mapeados a memoria.

Implementación correcta de inicialización de memoria ROM con un valor por defecto.

Implementación de los diferentes modos del microcontrolador (expandido, “bootstrap”, “special test”).

Interrupciones.

Es importante estar familiarizado con el lenguaje de programación Java y con los conceptos de la programación orientada a objetos, para poder comprender la relación que existe entre las diferentes clases y hacer uso correcto de la biblioteca.

Con el desarrollo y utilización de esta biblioteca de simulación se puede comprobar que el sobrepeso que ejerce el código manejado por una máquina virtual es prácticamente nulo para la simulación de dispositivos con velocidades de reloj relativamente bajas, como lo es el HC11. Es por lo mismo que una aproximación de interpretación para simular al HC11 es más que adecuada, siendo la recompilación dinámica demasiado complicada sin grandes beneficios para este caso.

Con esta biblioteca se pueden desarrollar herramientas que faciliten el aprendizaje y enseñanza del microcontrolador HC11 como lo son: aplicaciones del lado del servidor (web), un simulador gráfico específicamente diseñado para las necesidades de la institución, una herramienta de búsqueda de códigos de operaciones con ejemplos “vivos” en la que se pueda comprobar el funcionamiento de una operación al mismo tiempo que se tiene la definición y explicación de la instrucción, y cualquier otra herramienta útil que se imagine.

## REFERENCIAS

- [1] Motorola Inc., *M68HC11, The Referente Manual*, USA, 1991.
- [2] Motorola Inc., *M68HC11A8, Data Technical Manual*, USA, 1991.
- [3] Driscoll, F., *Data Acquisition and Process Control with the M68HC11 Microcontroller*, USA, Macmillan Publishing Company, 1994.
- [4] Huang, Han-Way, *MC68HC11 an Introduction: Software and Hardware Interfacing*, USA, Thomson Learning, 1996.
- [5] Motorola Inc., *Freeware 8-bit Cross Assemblers User's Manual*, USA, Kevin Anderson Field Applications Engineer.
- [6] Spasov, Peter, *The Microcontroller Technology*, USA, Prentice Hall.



**César Olea Aguilar** es Ingeniero en Electrónica, con acentuación en Instrumentación y Control, egresado del Instituto Tecnológico de Sonora.

Sus áreas de interés son: Programación en C/C++, Java, ensamblador para el x86, ensamblador para el microcontrolador HC11, y los sistemas digitales en general.

Trabajó en la fábrica de software del ITSON (Novutek) y para la empresa CIMS A en Nogales Sonora. Actualmente cursa estudios de posgrado en el área de computación en el CICESE, B:C:N.

Es activista del movimiento de software libre y actualmente se encuentra desarrollando el simulador Jiss para el HC11 utilizando la biblioteca de simulación que creó como proyecto de titulación. Promueve los sistemas operativos Open Source y su uso tanto en empresas particulares como en instituciones públicas.

**Eduardo Romero Aguirre** obtuvo el grado de Ingeniero en Electrónica opción Instrumentación en el Instituto Tecnológico de Orizaba en 1995 y el grado de Maestro en Ciencias en Ingeniería Electrónica en el área de Sistemas Digitales, en el Centro Nacional de Investigación y Desarrollo Tecnológico – CENIDET de Cuernavaca, Morelos, México en 1999. Ha realizado diversos proyectos relacionados con sistemas de adquisición de datos. Actualmente labora como profesor/investigador de tiempo completo en el Instituto Tecnológico de Sonora.



# **RIEE&C**

**Revista de Ingeniería Eléctrica, Electrónica y Computación**

## **INFORMACIÓN PARA LOS AUTORES**

RIEE&C, Revista de Ingeniería Eléctrica, Electrónica y Computación se publica semestralmente. Se aceptan artículos originales en aplicaciones de la ciencia, desarrollo de nueva tecnología o soluciones eficientes de ingeniería, siempre y cuando no hayan sido publicados o estén bajo consideración para publicarse en alguna otra revista.

Todos los documentos deberán enviarse al editor en jefe de RIEE&C, Javier Pérez Ramírez, por e-mail a: [rieandc@itson.mx](mailto:rieandc@itson.mx)

Los artículos se revisarán por especialistas en el área y dictaminarán si el artículo es apropiado para su publicación. No se deberán enviar manuscritos directamente a los editores asociados.

Los manuscritos enviados por los autores deberán seguir el formato que muestra la guía para el autor, la cual se encuentra en la página de internet <http://www.itson.mx/rieandc>.



# RIEE&C

## Contenido

### **Single phase DC/AC bi-directional converter with high-frequency isolation**

Beristáin J. José A., Bordonau F. Josep, Busquets M. Sergi, Rocabert S. Joan and Murillo V. Ismael

1

### **Inversor trifásico SPWM para el control de velocidad de un motor de inducción implementado en el microcontrolador PIC18F2431**

Aganza T. Alejandro, Pérez R. Javier y Beristáin J. José Antonio

7

### **Implementación de un controlador adaptativo para un sistema de primer orden con ganancia y polo variantes, basado en DSP**

Pérez R. Javier, Pelayo S. Roger Pelayo y Manzanarez A. Francisco

17

### **Diseño de un sistema para análisis de las vibraciones mecánicas como método de mantenimiento predictivo en máquinas rotativas**

Bojórquez D. Gilberto, Rivera M. José, Chacón M. Mario y Carillo R. Mariano

24

### **Aplicación y adaptación del método end to end para mejorar el uso de los recursos y el desempeño de redes de cómputo**

Marchena M. Misael, Domitsu K. Manuel y Solís G. Ricardo T.

29

### **Biblioteca de simulación por software multiplataforma para el microcontrolador M68HC11**

Olea A. César y Romero A. Eduardo

34

